



UNIVERSITY OF PATRAS
DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING

DIVISION OF SYSTEMS AND AUTOMATIC CONTROL
LABORATORY OF AUTOMATION AND ROBOTICS

Development of a Robotic System for Automating Synthetic Chemistry Workflows using Visual Feedback

DIPLOMA THESIS

VASILEIOS KARATRANTOS

SUPERVISOR: CHARALAMPOS MPECHLIOULIS

PATRAS – OCTOBER 2026

University of Patras, Department of Electrical and Computer Engineering.

Karatrantos Vasileios

© 2026 – All rights reserved

The whole work is an original work, produced by Karatrantos Vasileios, and does not violate the rights of third parties in any way. If the work contains material which has not been produced by him, this is clearly visible and is explicitly mentioned in the text of the work as a product of a third party, noting in a similarly clear way his/her identification data, while at the same time confirming that in the case of using original graphic representations, images, graphs, etc., he has obtained the unrestricted permission of the copyright holder for the inclusion and subsequent publication of this material.

CERTIFICATION

It is certified that the Diploma Thesis titled

Development of a Robotic System for Automating Synthetic Chemistry Workflows using Visual Feedback

of the Department of Electrical and Computer Engineering student

KARATRANTOS VASILEIOS

Registration Number: 1083998

was presented publicly at the Department of Electrical and Computer
Engineering on

06/10/2026

and was examined by the following examining committee:

Charalampos Mpechlioulis, Professor, Electrical and Computer Engineering
Department (supervisor)

Konstantinos Chatzilygeroudis, Assistant Professor, Electrical and Computer
Engineering Department (committee member)

Konstantinos Moustakas, Professor, Electrical and Computer Engineering
Department (committee member)

The Supervisor

The Director of the Division

Charalampos Mpechlioulis
Professor

Michail Markakis
Associate Professor



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ
ΕΡΓΑΣΤΗΡΙΟ ΑΥΤΟΜΑΤΙΣΜΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

**Ανάπτυξη ρομποτικού συστήματος για την
αυτοματοποίηση διαδικασιών συνθετικής χημείας με
χρήση οπτικής ανατροφοδότησης**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΚΑΡΑΤΡΑΝΤΟΣ ΒΑΣΙΛΕΙΟΣ

ΕΠΙΒΛΕΠΩΝ: ΧΑΡΑΛΑΜΠΟΣ ΜΠΕΧΛΙΟΥΛΗΣ

ΠΑΤΡΑ – ΟΚΤΩΒΡΙΟΣ 2026

Πανεπιστήμιο Πατρών, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών.

Βασίλειος Καρατράντος

© 2026 – Με την επιφύλαξη παντός δικαιώματος

Το σύνολο της εργασίας αποτελεί πρωτότυπο έργο, παραχθέν από τον Καρατράντο Βασίλειο, και δεν παραβιάζει δικαιώματα τρίτων καθ' οιονδήποτε τρόπο. Αν η εργασία περιέχει υλικό, το οποίο δεν έχει παραχθεί από τον ίδιο, αυτό είναι ευδιάκριτο και αναφέρεται ρητώς εντός του κειμένου της εργασίας ως προϊόν εργασίας τρίτου, σημειώνοντας με παρομοίως σαφή τρόπο τα στοιχεία ταυτοποίησής του, ενώ παράλληλα βεβαιώνει πως στην περίπτωση χρήσης αυτούσιων γραφικών αναπαραστάσεων, εικόνων, γραφημάτων κ.λπ., έχει λάβει τη χωρίς περιορισμούς άδεια του κατόχου των πνευματικών δικαιωμάτων για την συμπερίληψη και επακόλουθη δημοσίευση του υλικού αυτού.

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η Διπλωματική Εργασία με τίτλο

Ανάπτυξη Ρομποτικού Συστήματος για την Αυτοματοποίηση Διαδικασιών Συνθετικής Χημείας με χρήση Οπτικής Ανατροφοδότησης

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας
Υπολογιστών

ΚΑΡΑΤΡΑΝΤΟΥ ΒΑΣΙΛΕΙΟΥ ΤΟΥ ΘΕΟΔΩΡΟΥ

Αριθμός Μητρώου: 1083998

Παρουσιάστηκε δημόσια στο Τμήμα Ηλεκτρολόγων Μηχανικών και
Τεχνολογίας Υπολογιστών στις

06/10/2026

και εξετάστηκε από την ακόλουθη εξεταστική επιτροπή:

Χαράλαμπος Μπεχλιούλης, Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και
Τεχνολογίας Υπολογιστών (επιβλέπων)

Κωνσταντίνος Χατζηλυγερούδης, Επίκουρος Καθηγητής, Τμήμα
Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών (μέλος επιτροπής)

Κωνσταντίνος Μουστάκας, Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και
Τεχνολογίας Υπολογιστών (μέλος επιτροπής)

Ο Επιβλέπων

Ο Διευθυντής του Τομέα

Χαράλαμπος Μπεχλιούλης
Καθηγητής

Μιχαήλ Μαρκάκης
Αναπληρωτής Καθηγητής

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στο Εργαστήριο Αυτοματισμού και Ρομποτικής του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών της Πολυτεχνικής Σχολής του Πανεπιστημίου Πατρών.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα Καθηγητή, κ. Χαράλαμπο Μπεχλιούλη, για τη βοήθεια να επιλέξω ένα θέμα που με ενδιέφερε, με θεωρητικό αλλά και τεχνικό σκέλος, για την καθοδήγησή του σε κάθε βήμα της εργασίας και για την ελευθερία που μου έδωσε στον τρόπο ανάπτυξης του θέματος. Επίσης, θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα Λάμπρο Πρίντζιο για τη βοήθειά του με τον εξοπλισμό του εργαστηρίου και την πειραματική διάταξη.

Επιπλέον, θα ήθελα να ευχαριστήσω την οικογένειά μου, που στάθηκε δίπλα μου σε όλη τη διάρκεια των σπουδών μου, με στήριξε στις δύσκολες στιγμές και μου έδωσε τη δυνατότητα να αφοσιωθώ σε αυτό που ήθελα να κάνω.

Τέλος, ένα μεγάλο ευχαριστώ στους φίλους μου, με τους οποίους τα τελευταία έξι χρόνια μοιραστήκαμε τις δυσκολίες και περάσαμε όμορφες στιγμές που δεν πρόκειται να ξεχάσω. Χωρίς αυτούς, η φοιτητική ζωή δεν θα ήταν το ίδιο ευχάριστη.

ABSTRACT

Development of a Robotic System for Automating Synthetic Chemistry Workflows using Visual Feedback

STUDENT NAME, SURNAME:
VASILEIOS KARATRANTOS

SUPERVISOR NAME, SURNAME:
CHARALAMPOS MPECHLIOULIS

This thesis develops a robotic system for automating liquid-handling procedures in a chemistry laboratory, using a seven-degree-of-freedom manipulator that locates the objects it handles by fiducial markers, observed by a camera before each operation, rather than replaying coordinates fixed in advance.

The task imposes a requirement that governs the whole design: the arm carries an open test tube containing liquid, and the tube must remain vertical throughout its motion and not merely at the endpoints. In motion-planning terms this is a path constraint on the orientation of the end-effector, and it confines the admissible configurations to a surface of measure zero within the space a sampling-based planner explores. General-purpose planners address such constraints by rejecting samples or by projecting them numerically, and both remedies are expensive and satisfy the constraint only approximately.

The contribution of this work is to remove the constraint from the search rather than to enforce it during the search. The manipulator has a spherical-shoulder, revolute-elbow, spherical-wrist structure, which is established here from its kinematic model, and which admits a closed-form inverse solution. Exploiting that structure, an explicit analytic chart of the constraint surface is constructed, so that the planner searches a five-dimensional parameter domain each point of which maps in closed form to a complete seven-joint configuration. Within the nominal kinematic model of the arm, no configuration violating the constraint can be produced, because none exists in the space being searched.

Three planning architectures are built and compared: constrained search in the full joint space, a conservative four-dimensional reduction reported as a measured negative result, and the analytic chart. On the physical arm the chart planned each of the transits it handled in 7.7 - 80.5 ms with a maximum planned tool tilt of 0.00035° , against 457 to 7055 ms and tilts of up to 33.0° for the constrained baseline on the transits the chart declined. On a common benchmark of the same transits, repeated ten times per method, the chart planned about twenty times faster than the baseline. In a proof-of-concept evaluation, fourteen of sixteen complete transfers succeeded.

The system is assembled from a low-cost educational manipulator, a commodity webcam, printed fixtures and open-source software, and is operated through a graphical interface requiring no knowledge of the algorithms beneath it.

Keywords: *robotic manipulator, kinematic redundancy, constrained motion planning, constraint manifold, closed-form inverse kinematics, S-R-S structure, ArUco markers, ROS 2, laboratory automation.*

ΕΚΤΕΤΑΜΕΝΗ ΕΛΛΗΝΙΚΗ ΠΕΡΙΛΗΨΗ

Ανάπτυξη ρομποτικού συστήματος για την αυτοματοποίηση διαδικασιών συνθετικής χημείας με χρήση οπτικής ανατροφοδότησης

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ: ΒΑΣΙΛΕΙΟΣ ΚΑΡΑΤΡΑΝΤΟΣ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΕΠΙΒΛΕΠΟΝΤΟΣ: ΧΑΡΑΛΑΜΠΟΣ ΜΠΕΧΛΙΟΥΛΗΣ

Εισαγωγή και Κίνητρο

Η σύγχρονη έρευνα στη χημεία, τη βιολογία και τη φαρμακευτική απαιτεί την εκτέλεση μεγάλου αριθμού διαδικασιών με ταχύτητα, ακρίβεια και επαναληψιμότητα. Παρά την πρόοδο της εργαστηριακής αυτοματοποίησης, πολλές τυποποιημένες διαδικασίες εξακολουθούν να εκτελούνται χειρωνακτικά, γεγονός που εκθέτει τους ερευνητές σε επικίνδυνες ουσίες και καθιστά τα αποτελέσματα εξαρτημένα από την τεχνική και την κρίση του χειριστή.

Τα εμπορικά διαθέσιμα συστήματα χειρισμού υγρών είναι δαπανηρά και βασίζονται σε κλειστές αρχιτεκτονικές που επιτρέπουν ελάχιστη τροποποίηση. Επιπλέον, λειτουργούν χωρίς αντίληψη του περιβάλλοντος: εκτελούν κινήσεις προς συντεταγμένες καθορισμένες εκ των προτέρων και δεν διαθέτουν μέσο για να προσδιορίσουν πού βρίσκονται πραγματικά τα αντικείμενα που χειρίζονται. Οποιαδήποτε μεταβολή στη διάταξη του πάγκου εργασίας οδηγεί σε αποτυχία της διαδικασίας ή σε σύγκρουση.

Η άρση αυτού του περιορισμού εισάγει ωστόσο μία νέα απαίτηση. Ένα σύστημα που εντοπίζει τους στόχους του κατά τον χρόνο εκτέλεσης δεν μπορεί να στηριχθεί σε προκαταγεγραμμένες τροχιές αλλά οφείλει να τις παράγει το ίδιο, για θέσεις που δεν είναι γνωστές, παρά μόνο τη στιγμή της εκτέλεσης. Όταν ο βραχίονας μεταφέρει ανοικτό δοκιμαστικό σωλήνα με υγρό, το πρόβλημα δεν ανάγεται στην επίτευξη των κατάλληλων ακραίων θέσεων, διότι το υγρό επιβάλλει απαίτηση σε ολόκληρο το μήκος της κίνησης.

Διατύπωση του Προβλήματος

Ο άξονας του σωλήνα πρέπει να παραμένει κατακόρυφος καθ' όλη τη διάρκεια της μεταφοράς και όχι μόνο στα άκρα της. Δηλαδή, πρόκειται για **περιορισμό διαδρομής** (path constraint) στον προσανατολισμό του τελικού στοιχείου δράσης, και η διάκριση αυτή μεταβάλλει ριζικά τον χαρακτήρα του προβλήματος.

Το σύνολο των διαμορφώσεων που ικανοποιούν την απαίτηση αποτελεί υποσύνολο συνδιάστασης δύο μέσα στον επταδιάστατο χώρο διαμορφώσεων, επομένως σύνολο μηδενικού μέτρου: μία διαμόρφωση που λαμβάνεται τυχαία από τον χώρο των αρθρώσεων ανήκει σε αυτό με πιθανότητα μηδέν. Οι σχεδιαστές που βασίζονται σε δειγματοληψία λειτουργούν ακριβώς με τυχαία δείγματα και συνεπώς σχεδόν κάθε δείγμα που παράγουν είναι άκυρο.

Υπάρχουν δύο γενικές τεχνικές που αντιμετωπίζουν το ζήτημα, και καθεμία έχει κόστος. Η **χαλάρωση** αντικαθιστά τον περιορισμό με μία ζώνη ανοχής και απορρίπτει τα δείγματα εκτός αυτής. Έτσι, η ικανοποίηση καθίσταται προσεγγιστική και ο ρυθμός αποδοχής μειώνεται τετραγωνικά καθώς η ανοχή στενεύει. Η **προβολή** μετακινεί κάθε δείγμα αριθμητικά επάνω στην επιφάνεια του περιορισμού. Δηλαδή, κάθε δείγμα απαιτεί επαναληπτική επίλυση, και το αποτέλεσμα εξακολουθεί να ορίζεται από ένα αριθμητικό κατώφλι σύγκλισης.

Η παρούσα εργασία ακολουθεί διαφορετική οδό: αφαιρεί τον περιορισμό από την αναζήτηση αντί να τον επιβάλλει κατά τη διάρκειά της.

Κινηματική Ανάλυση

Ο χειριστής που χρησιμοποιείται είναι ο Elephant Robotics myArm 300 Pi, βραχίονας επτά βαθμών ελευθερίας. Η ανάλυση κατά Denavit-Hartenberg δείχνει ότι όλα τα μήκη συνδέσμων μηδενίζονται και ότι οι μετατοπίσεις των ζυγών αρθρώσεων είναι επίσης μηδενικές. Από τις ιδιότητες αυτές προκύπτει ότι οι άξονες των αρθρώσεων 1,2 και 3 τέμνονται σε κοινό σημείο, τον ώμο, και οι άξονες των αρθρώσεων 5,6 και 7 σε δεύτερο κοινό σημείο, το κέντρο του καρπού. Ο βραχίονας είναι επομένως δομής σφαιρικού ώμου – περιστροφικού αγκώνα – σφαιρικού καρπού (**S-R-S**).

Δύο συνέπειες απορρέουν. Πρώτον, η θέση του κέντρου του καρπού καθορίζεται αποκλειστικά από τις τέσσερις πρώτες αρθρώσεις. Δεύτερον, η απόσταση ώμου-καρπού εξαρτάται μόνο από τη γωνία του αγκώνα και προκύπτει

από την εφαρμογή του νόμου των συνημίτονων. Επιπλέον, αμφότερες οι τριάδες αρθρώσεων ανάγονται σε γωνίες Euler τύπου ZYZ, για τις οποίες το αντίστροφο πρόβλημα επιδέχεται ρητή λύση. Ο βραχίονας ικανοποιεί επομένως τη συνθήκη του Piereg και διαθέτει αντίστροφη κινηματική κλειστής μορφής.

Προτεινόμενη Μέθοδος

Η απαίτηση της κατακορυφότητας δεσμεύει δύο από τους τρεις προσανατολιστικούς βαθμούς ελευθερίας του τελικού στοιχείου δράσης και αφήνει έναν ελεύθερο: την περιστροφή περί τον κατακόρυφο άξονα, η οποία δεν μεταβάλλει την κλίση του αντικειμένου που μεταφέρει. Συνεπώς, το σύνολο των επιτρεπτών διαμορφώσεων έχει διάσταση πέντε.

Απαιτείται, λοιπόν, ένας χάρτης, δηλαδή μία ρητή απεικόνιση από πενταδιάστατο πεδίο παραμέτρων επάνω στην πολλαπλότητα του περιορισμού. Οι γενικοί σχεδιαστές κατασκευάζουν τέτοιους χάρτες αριθμητικά, μέσω προβολής, και η δομή S-R-S του βραχίονα παρέχει έναν σε κλειστή μορφή.

Η αναζήτηση διεξάγεται στον χώρο $(q_1, q_2, q_3, q_4, \varphi)$, όπου φ είναι η περιστροφή της αρπάγης περί την κατακόρυφο. Οι τέσσερις πρώτες συντεταγμένες καθορίζουν τη θέση του κέντρου του καρπού και η φ τον προσανατολισμό της αρπάγης. Οι τρεις εναπομένουσες αρθρώσεις προκύπτουν από την κλειστής μορφής αντιστροφή ZYZ. Κάθε σημείο του πενταδιάστατου χώρου απεικονίζεται επομένως σε πλήρη διαμόρφωση επτά αρθρώσεων.

Τρεις ιδιότητες προκύπτουν άμεσα. Ο περιορισμός ικανοποιείται ταυτοτικά **ως προς το ονομαστικό κινηματικό μοντέλο**, καθώς κάθε δείγμα προέρχεται από σύνολο διαμορφώσεων που ήδη τον ικανοποιούν, δηλαδή δεν μεσολαβεί απόρριψη, προβολή ή ανοχή. Ο έλεγχος συγκρούσεων γίνεται **ακριβής**, διότι η απεικόνιση αποδίδει την πλήρη διαμόρφωση και ελέγχεται η πλήρης γεωμετρία της αρπάγης και του σωλήνα. Τέλος, η διάσταση δεν επιλέγεται αλλά επιβάλλεται από το πρόβλημα.

Η συνέχεια του χάρτη απαιτεί ιδιαίτερη μεταχείριση. Οι γωνίες του καρπού ανακτώνται μέσω αντίστροφων τριγωνομετρικών συναρτήσεων και επομένως ανήκουν σε ημιανοικτό διάστημα, δηλαδή διαδρομή που διασχίζει την τομή κλάδου παράγει προσανατολισμό απολύτως συνεχή ενώ η τιμή της άρθρωσης μεταπηδά κατά 2π . Το σφάλμα δεν εμφανίζεται σε κανένα σημείο ελέγχου αλλά αποκλειστικά μεταξύ αυτών. Η αντιμετώπιση συνίσταται στο ξετύλιγμα των ανακτηθεισών γωνιών ως προς το προηγούμενο σημείο και στον έλεγχο της διελυσσιμότητας επί της ακμής μεταξύ δειγμάτων και όχι εκ των υστέρων επί της ολοκληρωμένης λύσης.

Εναλλακτική Προσέγγιση

Εξετάστηκε επιπλέον μία συντηρητική τετραδιάστατη μείωση, στην οποία το σύνολο των τμημάτων πέραν του κέντρου του καρπού αντικαθίσταται από ενιαίο στερεό σώμα που περικλείει κάθε δυνατή θέση τους. Η εγγύηση είναι μονόπλευρη αλλά ορθή: διαδρομή ελεύθερη συγκρούσεων ως προς το σώμα είναι ελεύθερη συγκρούσεων και για τον πραγματικό βραχίονα.

Ο περιορισμός του κατακόρυφου προσανατολισμού καθιστά την ιδέα εφαρμόσιμη, μετατρέποντας το περικλείον σώμα από σφαίρα σε επίπεδο δακτύλιο και μειώνοντας τον όγκο του από 49,3 σε 26,4 λίτρα. Ωστόσο, η μέθοδος δεν παρήγαγε κανένα αξιοποιήσιμο σχέδιο στις μετακινήσεις της εργασίας, όχι λόγω σφάλματος στην κατασκευή, αλλά εξαιτίας του λόγου μεγεθών: η έκταση του τελικού στοιχείου δράσης είναι συγκρίσιμη με την ίδια την εμβέλεια του βραχίονα, οπότε ο δακτύλιος καταλαμβάνει το μεγαλύτερο μέρος του προσβάσιμου χώρου και, τοποθετημένος επάνω από τις βάσεις των σωλήνων, περικλείει πάντα γειτονικό σωλήνα.

Το αποτέλεσμα γενικεύεται ως εξής: το συντηρητικό φράγμα κλιμακώνεται με την εμβέλεια του εργαλείου, ενώ ο διαθέσιμος ελεύθερος χώρος με την εμβέλεια του βραχίονα. Η τεχνική υποβαθμίζεται ακριβώς εκεί όπου το φορτίο είναι μακρύ σε σχέση με τον χειριστή, και θα ήταν πλήρως λειτουργική σε βιομηχανικό βραχίονα με συμπαγές εργαλείο.

Το Ολοκληρωμένο Σύστημα

Το σύστημα αναπτύχθηκε στο Robot Operating System 2, με τον σχεδιασμό κίνησης να εκτελείται μέσω του MoveIt 2 και της βιβλιοθήκης OMPL και τους ελέγχους συγκρούσεων μέσω της FCL. Ο λογικός έλεγχος του βραχίονα υλοποιήθηκε σε C++.

Ο εντοπισμός των αντικειμένων επιτυγχάνεται με δείκτες αναφοράς ArUco. Η κάμερα βαθμονομείται σύμφωνα με τη μέθοδο του Zhang, και η θέση κάθε δείκτη ανακτάται από τις θέσεις των τεσσάρων γωνιών του στην εικόνα. Κάθε δείκτης εκφράζεται ως μετατόπιση από έναν δείκτη αναφοράς, τοποθετημένο σε μετρημένη θέση ως προς τη βάση του ρομπότ, ώστε τα σφάλματα που είναι κοινά σε όλες τις παρατηρήσεις να αλληλοαναιρούνται και η απόλυτη θέση της κάμερας να μην υπεισέρχεται στο αποτέλεσμα. Η οπτική ανατροφοδότηση χρησιμοποιείται για

τον εντοπισμό των αντικειμένων πριν από την εκτέλεση των κινήσεων, και όχι για συνεχή οπτική καθοδήγηση (visual servoing) κατά τη διάρκειά τους.

Ο ενσωματωμένος υπολογιστής του βραχίονα χρησιμοποιείται αποκλειστικά ως "γέφυρα" δικτύου, ενώ το σύνολο του υπολογιστικού φόρτου εκτελείται σε εξωτερικό Η/Υ. Η επιλογή αυτή υπαγορεύθηκε από τους περιορισμένους πόρους του ενσωματωμένου συστήματος.

Η αλληλεπίδραση του χρήστη με τον βραχίονα γίνεται μέσω γραφικής διεπαφής, που λειτουργεί ως αυτοτελής κόμβος ROS 2 και προσφέρει δύο τρόπους λειτουργίας: αυτόματο, όπου ο χρήστης επιλέγει το τελικό προϊόν και εκτελείται η αντίστοιχη αποθηκευμένη ακολουθία κινήσεων, και χειροκίνητο, όπου ο χρήστης συνθέτει ο ίδιος την ακολουθία ή εκτελεί μεμονωμένες κινήσεις.

Αποτελέσματα

Επειδή η μοντελοποιημένη σκηνή είναι γνωστό ότι διαφέρει από τον πραγματικό χώρο εργασίας, το σύστημα αξιολογήθηκε επί του φυσικού βραχίονα, μέσω δεκαέξι πλήρων μεταφορών κατανεμημένων στους πέντε δοκιμαστικούς σωλήνες.

Στην πειραματική αξιολόγηση, δεκατέσσερις από τις δεκαέξι μεταφορές ολοκληρώθηκαν χωρίς σφάλμα. Το αποτέλεσμα αφορά τη συγκεκριμένη επίδειξη σκοπιμότητας και όχι τη γενική αξιοπιστία του συστήματος. Οι δύο εξαιρέσεις αφορούσαν την τοποθέτηση του σωλήνα λίγα χιλιοστά εκτός της βάσης του· κανένας σωλήνας δεν έπεσε, καμία σύλληψη δεν απέτυχε και καμία δοκιμή δεν χρειάστηκε παρέμβαση.

Οι δεκαέξι μεταφορές περιλαμβάνουν σαράντα οκτώ επιμέρους μετακινήσεις. Ο χάρτης μειωμένης διάστασης διαχειρίστηκε τις τριάντα τρεις, σχεδιάζοντας κάθε μία σε 7,7 έως 80,5 ms με μέγιστη κλίση του σωλήνα στη σχεδιασμένη τροχιά 0,00035°. Η περιορισμένη αναζήτηση στον πλήρη χώρο των αρθρώσεων διαχειρίστηκε τις υπόλοιπες δεκαπέντε, επί του ίδιου πάγκου και εντός των ίδιων συνεδριών, σε 457 έως 7055 ms με κλίσεις από 10,6° έως 33,0°. Οι τιμές κλίσης αφορούν τη σχεδιασμένη τροχιά, καθώς δεν έγινε ανεξάρτητη μέτρηση της πραγματικής κλίσης του βραχίονα. Οι δύο κατανομές δεν επικαλύπτονται σε καμία από τις δύο ποσότητες. Επειδή στις δοκιμές οι δύο μέθοδοι δεν εφαρμόστηκαν στις ίδιες μετακινήσεις, πραγματοποιήθηκε επιπλέον κοινή σύγκριση στις ίδιες ακριβώς μετακινήσεις, με δέκα επαναλήψεις ανά μέθοδο. Σε αυτήν ο χάρτης σχεδίασε περίπου είκοσι φορές ταχύτερα από τη γενική μέθοδο (διάμεσος 44 ms έναντι 923 ms), και οι κατανομές της κλίσης δεν επικαλύφθηκαν.

Η διαφορά ως προς την κλίση δεν είναι διαφορά βαθμού αλλά είδους: ο χάρτης αδυνατεί να παραγάγει διαμόρφωση που παραβιάζει τον περιορισμό στο ονομαστικό κινηματικό μοντέλο, ενώ η γενική μέθοδος σέβεται μία ανοχή που επαληθεύεται εκ των υστέρων — και σε δύο περιπτώσεις υπερέβη αυτήν την ανοχή.

Η επιλογή της μεθόδου για κάθε μετακίνηση καθορίζεται από τη γεωμετρία του στόχου. Ο χάρτης απέρριψε κάθε μετακίνηση προς τους δύο πλησιέστερους σωλήνες και καμία προς τους τρεις απομακρυσμένους, σε κάθε επανάληψη, διότι στις πλησιέστερες θέσεις κάθε διαθέσιμη διαμόρφωση στόχου βρίσκεται σε μηχανικό όριο άρθρωσης. Πρόκειται για τον περιορισμό που η εργασία προβλέπει σε δομικό επίπεδο, εδώ παρατηρημένο ως σαφής διαχωρισμός σε πέντε θέσεις του ίδιου πάγκου.

Η επαναληψιμότητα του εντοπισμού εντός μίας σάρωσης είναι καλύτερη από 0,6 mm. Η απόλυτη ακρίβεια εκτιμήθηκε έναντι μετρήσεων με το χέρι, από τις οποίες οι θέσεις του συστήματος αποκλίνουν κατά 4,6 έως 17,6 mm. Οι αποκλίσεις διαφέρουν από δείκτη σε δείκτη και είναι της ίδιας τάξης με την αβεβαιότητα της ίδιας της μέτρησης αναφοράς, η οποία εκτιμάται σε περίπου ± 6 mm ανά συντεταγμένη.

Τέλος, η μείωση του χρόνου σχεδιασμού δεν μειώνει αναλόγως τη διάρκεια της μεταφοράς. Στο 15 % της ονομαστικής ταχύτητας, μία πλήρης μεταφορά διαρκεί 55 έως 95 δευτερόλεπτα, εκ των οποίων ο σχεδιασμός καταλαμβάνει μερικές δεκάδες χιλιοστά του δευτερολέπτου· το κόστος μετατοπίζεται στην αντίστροφη κινηματική της σύλληψης, στην επιλογή θέσης προσέγγισης και στην ίδια την κίνηση του βραχίονα.

Συμπεράσματα και Μελλοντικές Βελτιώσεις

Η εργασία δείχνει ότι, όταν ο μηχανισμός διαθέτει εκμεταλλεύσιμη δομή, ένας περιορισμός διαδρομής δεν χρειάζεται να αντιμετωπιστεί ως εμπόδιο κατά την αναζήτηση, αλλά μπορεί να ενσωματωθεί στον ίδιο τον χώρο αναζήτησης. Το αποτέλεσμα είναι μέθοδος ακριβέστερη και, στις μετακινήσεις όπου εφαρμόστηκε, ταχύτερη από τη γενική τεχνική.

Οι κύριοι περιορισμοί της προσέγγισης είναι τρεις. Ο χάρτης διατηρεί σταθερό τον κλάδο του καρπού, ώστε κίνηση της οποίας τα άκρα ανήκουν σε διαφορετικούς κλάδους δεν αναπαρίσταται εντός ενός χάρτη. Η μέθοδος προϋποθέτει ακριβή αποσύζευξη, η οποία ισχύει κατασκευαστικά για τον συγκεκριμένο βραχίονα. Τέλος, η μείωση αφορά περιορισμούς που αφήνουν παραμετροποιήσιμη υπολειπόμενη ελευθερία.

Ως προς το σύστημα, η σημαντικότερη έλλειψη είναι ότι η διεπαφή δεν εμφανίζει στον χειριστή την πρόοδο και το αποτέλεσμα κάθε εντολής και δεν διαθέτει δυνατότητα διακοπής, γεγονός που εμποδίζει τη λειτουργία χωρίς επίβλεψη.

Οι φυσικές επεκτάσεις είναι αντιστοίχως: άτλαντας δύο χαρτών ενωμένων στην ιδιομορφία του καρπού, ώστε να αρθεί ο περιορισμός του κλάδου: εμφάνιση της κατάστασης κάθε εντολής στη διεπαφή μαζί με δυνατότητα διακοπής, ανάγνωση των πραγματικών θέσεων των αρθρώσεων από τους κωδικοποιητές και προσθήκη ογκομετρικού ελέγχου, ώστε οι συνταγές να καθορίζουν ποσότητες κι όχι μόνο σειρά.

Λέξεις-Κλειδιά: ρομποτικός βραχίονας, πλεονάζοντες βαθμοί ελευθερίας, σχεδιασμός κίνησης υπό περιορισμούς, πολλαπλότητα περιορισμού, αντίστροφη κινηματική κλειστής μορφής, δομή S-R-S, δείκτες ArUco, ROS 2, εργαστηριακή αυτοματοποίηση.

Table of Contents

1. Introduction	1
1.1. General Context and Motivation for the Thesis	1
1.2. Problem Statement	1
1.3. Scope of the Thesis	1
1.4. Aim and Objectives	2
1.5. Contributions of the Thesis	3
1.6. Implementation Methodology	3
1.7. Thesis Structure.....	4
2. Background and State of the Art	5
2.1. The Evolution of Laboratory Automation	5
2.2. Categories of Robotic Systems in the Chemistry Laboratory	6
2.3. The Operation and Scientific Value of Robotics in Laboratory Chemistry	7
2.4. Computer Vision	8
2.5. The Robot Operating System (ROS)	9
2.6. Motion Planning under Task Constraints	10
3. Mathematical Modelling and Kinematic Analysis.....	12
3.1. Introduction to Robot Kinematics	12
3.2 Geometric Descriptions and Transformations	14
3.3. Denavit-Hartenberg (DH) Kinematic Analysis	15
3.4. Forward Kinematics	20
3.5. Inverse Kinematics	23
3.6. Jacobian Analysis and Singularities	25
4. Problem Formulation.....	28

4.1 The Task and its Governing Requirement.....	28
4.2 Formal Statement of the Constrained Planning Problem	30
4.3 The Constraint as the Source of the Difficulty	32
4.4 The Dimension of the Admissible Set	35
4.5 Dynamic Parameters and the Collision Model	37
4.6 Payload Safety Criteria.....	39
4.7 The Computational Pipeline.....	41
5. Hardware and Experimental Setup	43
5.1. Description of the Experimental Setup	43
5.2. Main Robotic Equipment	44
5.3. Machine Vision Subsystem.....	48
5.4 Peripheral Equipment and Labware.....	49
5.5. Communication Architecture and Interfacing	50
5.6. Safety and Constraints.....	52
5.7 Summary of System Components.....	53
6. Computer Vision	55
6.1 Introduction and Description of the Experimental Setup	55
6.2 Mathematical Model and Camera Calibration	56
6.3 Detection of ArUco Markers and Object Identification	58
6.4 Pose Estimation	59
6.5 Expression in the Robot's Coordinate Frame.....	61
6.6 Integration with the Control System via ROS 2 and tf2.....	63
6.7 Summary.....	63
7. Motion Planning on the Constraint Manifold	65
7.1 Motion Architecture and Anatomy of the Transfers	65

7.2 Baseline: Constrained Planning in the Full Joint Space.....	67
7.3 Implementation and Validation of the Closed-Form Solver	68
7.4 Reduction I: A Conservative Swept-Volume Surrogate.....	70
7.5 Reduction II: Exact Planning on the Constraint Manifold	73
7.6 Continuity of the Analytic Chart	77
7.7 Integration, Dispatch and Descent Feasibility	78
7.8 The Collision Environment	81
7.9 Time Parameterisation and Execution	82
7.10 Command Handling, Dispensing and Status Reporting.....	84
7.11 Scope and Evaluation.....	85
7.12 Summary.....	87
8. Graphical User Interface.....	89
8.1 Design Philosophy and Interface Architecture.....	89
8.2 ROS 2 Network Integration and Node Architecture	90
8.3 Auto Synthesis Mode.....	91
8.4 Manual Control and Batch Processing.....	93
8.5 Safety Mechanisms and their Limits	94
8.6 Summary.....	95
9. System Integration and Experimental Results	96
9.1 Experimental Protocol	96
9.2 Localisation Accuracy	98
9.3 Results of the Transfers	101
9.4 Planning Behaviour.....	102
9.5 Failures, Anomalies and Sources of Error	104
9.6 Summary.....	106

<i>10. Conclusions and Future Work</i>	<i>108</i>
10.1 Conclusions	108
10.2 Limitations	109
10.3 Future Work	111
10.4 Closing	112
<i>11. References.....</i>	<i>114</i>
<i>12. Appendix.....</i>	<i>120</i>

1. Introduction

1.1. General Context and Motivation for the Thesis

Modern scientific research in chemistry, biology and the pharmaceutical sciences is characterised by the need to execute a large volume of processes with high speed, accuracy and repeatability. Traditionally, experimental procedures, such as the synthesis of new materials and sample analysis, are time-consuming and vulnerable to human error, fatigue and operator dependency [1].

The formation of Lab 4.0 or Smart Lab [2], [3] extends the 4th Industrial Revolution [4] to the laboratory environment. The digitisation of data and the automation of processes through modern robotic systems and AI are expected to raise productivity and to change the way experiments are executed.

Robotics, and liquid-handling manipulators in particular, are central to this transition. Beyond processing rate, automation also reduces safety risks, since it removes the researcher from direct contact with toxic or hazardous substances [5]. A further requirement follows from the nature of the samples: an open vessel containing liquid constrains the entire motion of the arm that carries it, not just the positions at which that motion begins and ends. This distinction is the subject of the present work.

The ability of robotic systems to perform demanding tasks with precision and speed opens the way to the consistency of experimental protocols and to laboratory environments in which routine procedures run without human intervention. Given the rapid development of these systems and their use in increasingly complex experimental settings [6], such a transition is an achievable near-term prospect rather than a theoretical one.

1.2. Problem Statement

Despite the technological advances described above, many routine procedures in the chemistry laboratory are still performed manually. This places limits on both safety and repeatability: direct human intervention increases the risk of exposure to hazardous substances, and it makes experimental results dependent on operator technique and subjective judgement [1].

The difficulty, however, is not exclusive to human involvement. Existing automation is itself of limited availability. Commercially available liquid-handling systems are expensive and built on closed architectures that permit little modification, so that adapting them to a new protocol is often impossible for the laboratory that owns them [7].

Another limitation of these systems is that they operate without perception. They execute motions defined by prefixed coordinates and have no capacity to establish where the objects they manipulate actually are. As a result, any change to the layout of the workbench causes the procedure to fail or leads to collision and damage to equipment [8].

Removing this limitation, however, introduces a requirement of its own. A system that locates its targets at run time cannot rely on trajectories recorded in advance. It must generate them itself, for poses that are not known until the moment of execution. For a manipulator carrying an open vessel this is not a matter of reaching the right endpoints, since the liquid constrains the motion along its whole length. The problem addressed in this thesis is therefore the generation of such motions: planning for a redundant manipulator that must satisfy an orientation constraint at every point of the path, at a computational cost low enough for use within a laboratory work cycle.

1.3. Scope of the Thesis

This diploma thesis presents the development of an automated robotic system in which a 7-DoF manipulator acts as a "Lab Assistant". The system locates five test tubes and a mixing vessel through fiducial markers, and according to the user's selection, transfers the chosen tubes from their support bases and pours their contents into the mixing vessel.

The term visual feedback, as used in the title of this thesis, has a specific and limited meaning. The camera is used to localise the objects on the workbench before the transfers are executed, and every

motion is planned against the positions so obtained. The loop is therefore closed on the arrangement of the workbench, not on the motion of the arm. The system does not perform visual servoing: the camera is not consulted while the arm moves, and no trajectory is corrected against it during execution. In the terminology of the visual-control literature, the arrangement is a static look-and-move scheme [9].

Two problems are treated. The first is the construction of the application itself: the localisation of the workbench objects, the definition of the task sequence, and the control of the arm and its gripper through the stages of grasping, transfer and pouring. The second, which forms the principal contribution of the thesis, is the motion planning problem that the task creates. Because the tubes are open and contain liquid, the manipulator must hold each tube vertical along the whole of its path and not merely at its endpoints. Enforcing such a constraint within a general-purpose planner is expensive and yields only approximate satisfaction of it. The thesis develops an alternative formulation: an analytic parameterisation of the constraint manifold, obtained from the kinematic structure of the particular arm, on which motion planning is performed in a reduced five-dimensional space, so that the constraint is satisfied exactly, with respect to the nominal kinematic model, and by construction, at a reduced computational cost.

The work covers the full sequence of stages required to reach a working system: the kinematic analysis and modelling of the manipulator, the formulation of the constrained planning problem, the design of the planning and control algorithms, the description of the experimental setup, the experimental verification of the system's operation, and the conclusions drawn from it.

1.4. Aim and Objectives

The aim of this diploma thesis is the design and setup of a proof-of-concept system, demonstrating the utility of robotic automation at laboratory scale, and the practical benefit that flexible automation of this kind offers. The thesis further aims to show that the transition towards a "Smart Lab" is achievable with affordable hardware and open-source software, so that such technologies become accessible to research laboratories operating under budget constraints.

In order to achieve this goal, the following objectives were defined and carried out:

- **Kinematic Modelling:** Derivation of the forward kinematics of the manipulator and of a closed-form solution to the inverse problem, exploiting the arm's spherical-shoulder/revolute-elbow/spherical-wrist structure, so that end-effector poses can be converted to joint configurations exactly and at minimal computational cost.
- **Constrained Motion Planning (principal contribution):** Development of a planning method for transfers of open tubes, in which the requirement that the tube remain upright is treated as a constraint on the entire path rather than on its endpoints. The constraint manifold is parameterised analytically, so that the search is conducted in a five-dimensional space in which every sample corresponds, by construction, to a configuration satisfying the constraint exactly in the nominal kinematic model. The method is compared against constrained planning in the full joint space.
- **Object Localisation:** Integration of fiducial-marker detection for the automatic determination of the positions of the test tubes and the mixing vessel, so that the system adapts to changes in the layout of the workbench without reprogramming.
- **Task Sequencing:** Implementation of the control logic governing the order of operations – selection of a tube, grasping, transfer, pouring and return to its support base. The system operates in two modes: in manual mode the user selects individual tubes, and the corresponding transfers are executed on demand, while in automatic mode a stored recipe, that is, a predefined ordered sequence of tubes, is executed in full without further intervention.
- **Collision Avoidance:** Integration of a geometric model of the workbench into the planning process, so that planned motions are free of collision with the surrounding equipment and between the links of the arm itself.

- **Graphical User Interface (GUI):** Design of an interface offering both modes of operation, through which the user either selects tubes individually or chooses one of the available recipes, and which sends the corresponding commands to the main control algorithm for execution.
- **Experimental Verification:** Evaluation of the complete system on the physical arm, and numerical evaluation of the planning method in terms of success rate, planning time and the accuracy with which the orientation constraint is maintained.

1.5. Contributions of the Thesis

The principal contribution of this thesis is a motion-planning formulation in which the upright requirement is removed from the search rather than enforced during it. It comprises three elements:

- **An analytic chart of the constraint manifold:** Exploiting the S-R-S structure of the manipulator established in Chapter 3, the set of configurations at which the carried tube is exactly vertical is parameterised by five coordinates $(q_1, q_2, q_3, q_4, \varphi)$, where φ is the rotation of the tool about the vertical, and each point of that domain maps in closed form to a complete seven-joint configuration (Sections 4.4 and 7.5).
- **Motion planning in the reduced space:** A sampling-based planner searches the five-dimensional domain directly, so that, within the domain and wrist branch set out in Section 7.5.5, the constraint is satisfied identically in the nominal kinematic model rather than to within a tolerance, and collision checking is performed against the full geometry of the gripper and the carried tube. The conditions under which the chart remains continuous – unwrapping of the recovered wrist angles, a fixed wrist branch, and traversability tested on the edges between samples – are identified and enforced within the planner (Sections 7.5 and 7.6).
- **An evaluation on the physical arm:** The method is compared against constrained search in the full joint space, on the physical arm and on a common benchmark of identical queries (Chapter 9). A conservative four-dimensional swept-volume reduction is also constructed and reported as a negative result, together with the geometric ratio that causes it (Section 7.4).

The remaining components – marker-based localisation, task sequencing and the graphical interface – are the means by which the method is placed in a working system and evaluated on a real task.

1.6. Implementation Methodology

The system was developed using established open-source robotics tools, for both the software and the hardware components.

The Robot Operating System 2 (ROS 2), in the Humble Hawksbill distribution [10], was used as the central middleware. Its node-based architecture allows the sub-systems – object detection, motion planning, arm control and user interface – to be developed independently and to communicate through distinct interfaces, with each executing as a separate process.

Software

The control logic of the arm was written in C++, for performance and for direct access to the planning libraries described below.

Motion planning and collision avoidance were conducted with MoveIt 2 [11], the most widely used open-source motion planning framework in the ROS ecosystem. Sampling-based planning within MoveIt is provided by OMPL [12], and the RRTConnect planner [13] was used as the baseline against which the method developed in this thesis is compared. Collision queries are resolved by the Flexible Collision Library (FCL) [14], which MoveIt employs as its collision back-end.

Visualisation of the state of the system and of the planning scene was performed in RViz 2 [15], part of the ROS 2 ecosystem.

Image acquisition and processing were implemented with OpenCV [16]. The positions of the objects on the workbench are recovered from ArUco fiducial markers [17], which were selected because each

marker carries a unique identifier, allowing individual tubes to be distinguished, because a single planar marker determines a full six-degree-of-freedom pose, and because detection is computationally inexpensive and available as a mature implementation within OpenCV.

Hardware

Motion is executed by an Elephant Robotics myArm 300 Pi [18], a 7-DoF manipulator selected for its low cost, compact dimensions, light weight and native support for ROS, properties that make it an example of equipment accessible to a laboratory of modest budget.

Handling of the tubes is performed by a custom 3D-printed gripper (Thor Gripper) [19] mounted on the arm's mounting face.

Perception of the workspace is provided by a commodity USB web camera, a Creative Live! Cam Sync 1080p V2 [20], with a 77° field of view, chosen deliberately in preference to a specialised industrial or depth camera in keeping with the affordability objective of the work.

1.7. Thesis Structure

- *Chapter 2* presents the necessary background: the evolution of laboratory automation and the concept of "Lab 4.0", the categories of robotic systems used in the chemistry laboratory, the technologies on which this implementation relies, and the state of the art in motion planning under task constraints.
- *Chapter 3* develops the mathematical model of the manipulator. The general theory of robot kinematics is presented first, followed by the kinematic analysis of the 7-DoF arm used in this work, including the structural properties on which the later chapters depend.
- *Chapter 4* formulates the planning problem. The transfer task is stated formally, the requirement that the tube remain upright is expressed as a constraint on the whole path, the resulting set of admissible configurations is characterised, the dynamic and collision models under which planning is performed are defined, and safety criteria for the liquid payload are derived.
- *Chapter 5* describes the architecture of the system, with emphasis on the hardware components and their integration.
- *Chapter 6* presents the object recognition procedure: the detection of the fiducial markers, the recovery of the positions of the tubes and the mixer in the workspace, and the way this information is supplied to the rest of the system.
- *Chapter 7* presents the motion planning method that forms the principal contribution of this thesis: the analytic chart of the constraint manifold and planning in the reduced space it defines, together with the constrained baseline and the swept-volume reduction against which it is set. The sequencing logic of the mixing procedure is also described.
- *Chapter 8* describes the design and operation of the graphical user interface, through which the process is controlled and supervised.
- *Chapter 9* sets out the experimental evaluation of the complete system on the physical arm, including the comparison of the planning methods.
- *Chapter 10* presents the conclusions of the thesis and proposes directions for future extension and improvement of the system.

2. Background and State of the Art

This chapter presents the background necessary to place the present work in context. The history of laboratory automation is reviewed first, followed by the modern concept of the "Lab 4.0". The categories of robotic systems used in the chemistry laboratory and their respective roles are then examined, and the fundamental principles of the technologies employed in this implementation, namely computer vision and the ROS environment, are presented. The chapter closes with a review of motion planning under task constraints, against which the method developed in this thesis is positioned.

2.1. The Evolution of Laboratory Automation

Laboratory automation is defined as the strategic use of technological equipment and software to optimise or replace manual procedures in scientific research [8]. The history of the field is not merely one of substituting machines for the human hand, but of a continuous progression towards the management of complexity and the processing of data on an increasing scale [7].

2.1.1. The Limitations of Manual Work and the Origins of Automation

Laboratory procedures in synthetic chemistry have traditionally depended entirely on the skill of the researcher performing them. Routine operations such as the transfer of liquids by pipette or the mixing of solutions have always been carried out by hand. This approach is flexible and requires no specialised equipment, but it is subject to a number of significant limitations [21].

Simple operations repeated continuously induce fatigue and lapses of concentration in the researcher performing them, and errors of this kind propagate into incorrect results and incorrect conclusions [21]. The repeated performance of routine tasks also constitutes an inefficient use of human resources, which could instead be directed towards more demanding work such as data analysis, the evaluation of results and the design of new experiments [21].

The most serious limitation, however, is the hazardous nature of manual work in the chemical laboratory. Direct contact with the substances being handled, which may be toxic, radioactive or sensitive to external factors, complicates the experimental procedure and places the researcher at risk [5]. Protective measures are consequently required, both to protect the researcher and to protect the sample from the researcher.

The removal of the human operator from certain procedures therefore became necessary. Continuous-flow analysers, introduced in the late 1950s [22], represented the first substantial step in this direction, automating the sequential processing of samples in clinical analysis. Programmable laboratory robots followed two decades later: the Zymate system, introduced by Zymark in 1982 [23], was a reprogrammable arm capable of performing laboratory unit operations such as pipetting, pouring and the operation of instruments [24], and within a few years several hundred such systems were in use [25]. During the 1990s these robots were incorporated into the integrated High-Throughput Screening (HTS) platforms of the pharmaceutical industry, which combined robotic manipulation, miniaturised assays and automated data management to test very large numbers of compounds [26].

However, these early installations were highly specialised machines, each dedicated to a single operation performed at high speed. They functioned as "islands of automation", isolated from one another, and human intervention was necessary in order to transfer samples between them [25].

2.1.2. "Lab 4.0" and the "Smart Lab"

The present era is one of profound change in the laboratory, described by the terms "Lab 4.0" [27] or the "Smart Lab". Both indicate how the concepts of the Fourth Industrial Revolution (Industry 4.0) are applied to the laboratory environment [3].

The model is not defined by individual automated instruments, but by an ecosystem of devices interconnected with one another. Its principal characteristics are the following:

- **Interconnectivity (Internet of Things – IoT):** The creation of an environment in which instruments, such as balances, centrifuges, robotic arms and others, communicate, cooperate and exchange data in real time [3].
- **Robotic Flexibility:** The transition from rigid, single-purpose machines to articulated robotic arms and mobile robots, which may be reprogrammed for a variety of tasks and adapted to changes in the procedure. Flexibility of this kind requires that the robot can define the state of its own workspace and therefore requires sensing rather than the execution of predetermined motions [28].
- **Digitisation and Data:** The automatic recording of every experimental parameter, such as temperature, mixing time and quantities of material, into digital databases, which makes possible the application of AI methods to the analysis of results and the prediction of outcomes [28].

The furthest extension of this development is the cloud laboratory. In this model the physical presence of the researcher is not necessary. The experimental parameters are transmitted remotely, and the experiment is executed by robots, sensors and automated instruments inside a controlled environment [29]. Once completed, the resulting data are uploaded and made available to the researcher digitally. Such facilities are, however, centralised installations requiring significant capital investment, and access to them is mediated by the platform of the provider, which is why low-cost systems, operating inside the individual laboratory, are still relevant.

The present diploma thesis is situated within this framework: it develops a robotic assistant of the kind on which such laboratories depend, at a scale accessible to an individual research group.

2.2. Categories of Robotic Systems in the Chemistry Laboratory

In the automation of liquid handling and of the laboratory environment in general, the robotic systems employed fall into three categories, distinguished by their mode of operation and by the degrees of freedom they provide.

1. **Cartesian Coordinate Systems (Gantry Systems):** These are the most conventional and most widely used category of liquid handler. They move by linear translations along three orthogonal axes (X, Y, Z) within a strictly delimited workspace [30], normally enclosed within a protective case.

Such machines are characterised by high speed and high positioning accuracy, which makes them well suited to high-throughput screening applications [31]. Their architecture makes them large and demanding of installation space, their purchase and maintenance costs are high, and human intervention remains necessary during the procedure, since samples must be put inside and removed from the workspace by hand. Modification is also difficult and expensive, whether of the software or of the hardware, if the user needs to alter the operations performed or to integrate additional equipment [6]. While adequate regarding speed and accuracy, these systems are, as a result, less suitable for laboratory environments that are dynamic and require flexibility and frequent changes of protocol.

2. **SCARA Robots:** The SCARA (Selective Compliance Assembly Robot Arm) [32] is an intermediate category, employed principally for pick-and-place operations. Its structure consists of two revolute joints whose axes are vertical and parallel, which position the tool within a horizontal plane, a prismatic joint providing vertical translation, and a fourth revolute joint permitting rotation of the tool about the vertical axis.

SCARA robots are faster than articulated arms in planar transfer operations, and the rigidity of their vertical axis affords high precision in insertion tasks such as pipetting into vials. The consequence of this structure is that the orientation of the end-effector is restricted: the tool may be rotated about the vertical axis, but its axis cannot be inclined away from the vertical [33], so approaches to an object are possible only from above.

It is worth noting that this kinematic structure enforces, mechanically, without any computational effort, the constraint with which the present thesis is concerned: a vessel carried by such a robot cannot be tipped, and its contents cannot be spilled. However, that guarantee is obtained at the cost of the corresponding capability, since a robot which cannot tilt its tool can neither pour nor stir, and cannot approach an object horizontally. These operations require a manipulator that can freely tilt its end-effector. Therefore, the control system, and not the mechanism itself, must keep it upright.

3. **Articulated Robotic Arms:** This category consists of robots with revolute joints that offer multiple degrees of freedom. It is the category with which the present thesis is concerned, as the system is developed on a 7-DoF robotic arm.

Robots of this type simulate most closely the motion of the human arm, and their development is the most active area of laboratory robotics at present. They are able to approach an object from any direction, to perform delicate operations, and to manipulate laboratory glassware designed for human use, grasping a test tube, transporting it, inclining it, or transferring it from one station to another, without the purpose-built consumables that other categories require [6]. An important subcategory is that of collaborative robots, which incorporate safety sensing and are therefore able to operate within the same workspace as the researcher without protective enclosure, giving practical form to the concept of the digital laboratory assistant [34].

The category is, however, inferior in execution speed to both gantry systems and SCARA robots, and the complexity of its kinematics makes the programming and parameterisation of such systems more demanding [35]. Despite this, the capacity of articulated arms to reproduce human agility and to adapt to complex procedures makes them the most appropriate choice for the development of laboratory assistants.



Figure 2.1: Gantry System: Hamilton Microlab STAR [36]

Figure 2.2: SCARA System: "EPSON GXI-C" [37]

Figure 2.3: Articulated Robotic Arm: KUKA KR AGILUS [38]

2.3. The Operation and Scientific Value of Robotics in Laboratory Chemistry

The development of the present robotic system and its application to the handling and mixing of liquid chemicals is not aimed only at the solution of a technical problem; it also addresses needs appearing in modern research labs. The transition towards the "Lab 4.0" requires systems capable of operating with high speed, precision and safety [3]. The operational and scientific value of the Lab Assistant developed in this thesis may be considered from four perspectives:

1. **Reliability and Accuracy:** The validity of experimental results depends on the stability of the conditions under which the procedure is executed. Such procedures require a high degree of repeatability and the greatest achievable precision [39]. A major cause of errors is the loss of concentration caused by repetitive and boring tasks. Also, at small scales, the agility of the human hand is frequently insufficient to attain the precision required [40]. Robotic systems address both difficulties. The repeatability of a robotic arm is not affected by fatigue or the number of repetitions performed, although periodic maintenance is required because of mechanical wear and drift in calibration [3]. Moreover, the parameters of the procedure are controlled by a predefined program, so that each execution reproduces the preceding one. This way, the probability of error is reduced, along with the possibility that an experiment or a measurement will have to be repeated [41].

2. **Safety and Hygiene:** The introduction of robotic systems into the laboratory provides protection to the personnel, to the equipment, and to the experimental samples. Procedures involving toxic, radioactive or hazardous substances can be assigned to the manipulator, which executes them under the direction of the researcher. Direct contact with these materials is thereby limited to those stages that remain outside the scope of the automated procedure, such as the loading of samples [5]. The risk of contamination of samples by the operator is also reduced. In particularly sensitive experiments even exhaled breath or contact with the skin may affect the outcome, which is why protective clothing is often required [42]. A robotic system removes the operator from the majority of such interactions, and, when combined with an appropriate enclosure, permits the sample and the associated instrumentation to be maintained under controlled conditions [1].
3. **Efficiency and Workflow Optimisation:** Commercial systems of this kind require significant initial investment: the machines are expensive, and their installation and configuration demand specialised expertise. This cost is one of the main reasons why automation remains restricted to well-funded facilities, and it is the reason why systems assembled from low-cost hardware and open-source software, like the one developed in this thesis, deserve investigation. Where the volume of work justifies it, however, the investment is recovered [8], [43]. The execution of procedures is accelerated, without the interruptions that attend manual work and the reduction in error leads to an equivalent saving of material, since errors of manipulation often damage equipment or make the repetition of an experiment necessary, consuming reagents.
4. **Digital Transformation and Future Readiness:** A further contribution of robotic systems is that they build the infrastructure upon which the full digitisation of the laboratory depends. Every motion and every operation may be recorded digitally as it occurs, with an accuracy and consistency impossible to achieve using manual records. Information that would be lost is retained in structured databases and remains available for analysis [7]. Robotic systems are the physical means through which algorithmic methods, including machine learning, act upon the laboratory: a "Smart Lab" relies on an automated mechanism able both to execute experiments and to supply the resulting data to such algorithms [44]. The present thesis is situated within this context. The transition to automated procedures requires the appropriate equipment and the algorithmic means to exploit it, and this work is concerned with both. A scenario of substance mixing, carried out in cooperation with the user, is implemented and executed on physical hardware, as a demonstration that a flexible lab assistant can be developed with available methods.

2.4. Computer Vision

Computer Vision is the field that allows robotic systems to extract geometric information from digital images. In robotic applications, object detection is not the only important goal. Instead, the accurate Pose Estimation of objects in three-dimensional space is critical [45]. For accurate Pose Estimation in unpredictable environments, Fiducial Markers (ArUco Markers) are often used [46].

2.4.1. Reference Markers (ArUco Markers)

The Reference Markers are objects with standard geometric characteristics, positioned in the camera field of view, inside the workspace of the robotic system, which act as fixed reference points. The most common class of Reference Markers used in robotics are ArUco Markers, which are square markers with an interior grid that acts as a binary code for their identification [17]. Their use offers advantages, as each marker carries a unique identifier that distinguishes it from the others, they are detected at low computational cost, and they provide the full position and orientation of the object from a single image.

2.4.2. Calibration and Spatial Transformation

For the software to convert the pixels of a 2-D image into real-world coordinates in 3-D space, camera calibration is essential, based on the Pinhole Camera model. Through this process, the system calculates the intrinsic optical parameters of the lens [47].

Once calibration is done, the recognition of an ArUco marker allows the algorithm to calculate the translation and rotation vectors of the marker relative to the camera. Through successive coordinate transformations – from the marker to the camera, and finally to the robot's base – the position and orientation of each object can be expressed relative to the base of the arm [35].

2.5. The Robot Operating System (ROS)

The Robot Operating System (ROS) is the most widely used open-source software for developing robotic applications. ROS is not a traditional operating system. It is middleware that runs on top of a host OS, in this case Ubuntu Linux [48]. The main role of ROS is to provide an environment, which allows the user to control complex hardware such as servo motors, cameras and sensors, without writing low-level code.

Communication Architecture

The main innovation of ROS lies in its distributed architecture. A robotic system in ROS does not rely on a single and centralised program. Instead, it is based on small and independent programs that communicate with each other. The main theoretical fundamentals that compose this network are the following [49]:

- **Nodes:** These are the executable processes that perform computational work. In this case, one node handles the computer vision algorithms, a second one the kinematics, a third one the GUI, etc.
- **Discovery:** In ROS 2 there is no central registry process. Communication is built upon the Data Distribution Service (DDS) standard, through which each node announces its presence on the network and discovers the others automatically.
- **Topics and Messages:** This is the main mechanism for asynchronous communication. Nodes exchange data through channels called Topics [50]: a node that generates data, such as the markers' position, is a Publisher, and a node that needs this data is a Subscriber, which registers to the corresponding Topic. The Publisher and the Subscriber don't have to be aware of each other's existence, which makes the system much more flexible.
- **Services:** ROS also provides a synchronous communication mechanism of the Request/Reply type, called a Service. Services are used when a node needs to request a specific action and wait for confirmation of its completion, such as opening the gripper's fingers [51]. The Client sends a Request and waits until the Server provides a Response.

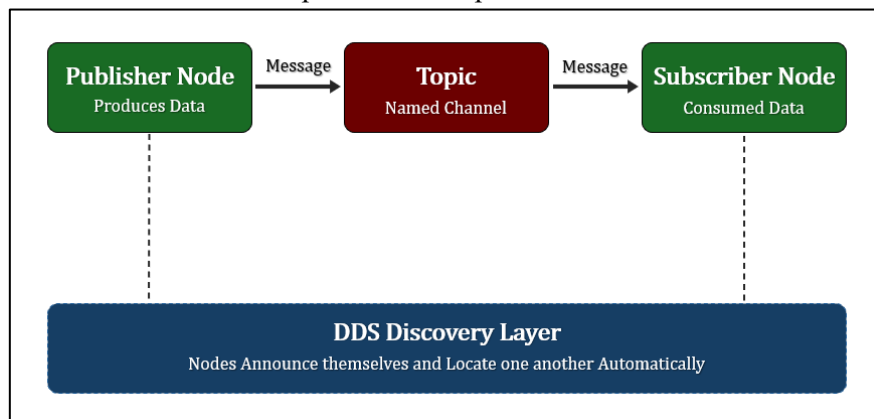


Figure 2.4: Communication architecture in ROS 2. Nodes exchange messages through Topics, while discovery is performed by the DDS layer without a central registry process.

The selection of ROS for the present diploma thesis was considered essential because of its modular structure. It allows easy integration of different subsystems, such as the camera, the robotic arm, and the GUI, into a unified network and provides the essential tools for visualisation (RViz [15]), motion planning (MoveIt [11]) and physics-based simulation (Gazebo [52]) which assist in the development of the system.

2.6. Motion Planning under Task Constraints

Sampling-based planners, of which RRTConnect [13] and the planners collected in OMPL [12] are representative, search the configuration space of a manipulator by drawing random configurations, discarding those in collision, and connecting the remainder into a tree or graph. Their appeal lies in their generality. They require only the ability to sample a configuration and to test it, and they make no assumption about the geometry of the mechanism or of the obstacles. That generality rests on an assumption that is easily overlooked: that the set of admissible configurations has non-zero volume, so that a random sample has a positive probability of falling inside it.

Many manipulation tasks violate that assumption. A robot carrying a tray, opening a door or, as in the present work, transporting an open vessel of liquid must satisfy a condition on the pose of its end-effector at every point of its path. Such a task constraint confines the admissible configurations to a lower-dimensional subset of the configuration space, a constraint manifold, which a random sample reaches with probability zero. The formal treatment of this for the present task is given in Section 4.3. The methods developed to plan on such manifolds fall into three families, as reviewed by Kingston et al. [53]:

1. **Relaxation:** The constraint is replaced by a band of finite tolerance about the manifold, and samples falling outside the band are rejected. The approach is simple and compatible with any planner. However, the fraction of samples accepted falls rapidly as the tolerance narrows, and the constraint is satisfied only to within the band.
2. **Projection:** Each sample is moved onto the manifold by iterative numerical correction, typically along the pseudoinverse of the constraint Jacobian. The Constrained Bi-directional RRT (CBiRRT) of Berenson et al. [54], which projects samples onto the manifold defined by constraints on the end-effector pose, is a well-known example. Stilman [55] compared several such schemes for planning in joint space under task constraints. Projection yields samples that satisfy the constraint to within a numerical threshold, at the cost of an iterative solve for every sample and for the intermediate states along every edge.
3. **Atlas and tangent-space methods:** Rather than projecting samples individually, these methods build local charts of the manifold, approximating it by its tangent spaces at points already reached, and sample within those charts. AtlasRRT [56] is the principal example. The framework of Kingston et al. [57], implemented in OMPL and available through MoveIt, unifies projection, atlas and tangent-bundle methods behind a common constrained state space.

The three families share one property: the manifold is known only implicitly, as the zero set of a constraint function that can be evaluated at a configuration but not inverted. Any chart of it is therefore constructed numerically, locally and approximately, since nothing more is available to a method that must serve any manipulator and any constraint.

A separate line of work concerns the inverse kinematics of redundant manipulators. Pieper [58] showed that a closed-form inverse solution exists when three consecutive joint axes intersect at a common point. For seven-joint arms of spherical-shoulder, revolute-elbow, spherical-wrist structure, the remaining redundancy is conventionally parameterised by the arm angle, the rotation of the elbow about the line joining the shoulder and the wrist centre. Shimizu et al. [59] derived a complete analytical solution for such arms in terms of this parameter, including the effect of joint limits, and applied it to redundancy resolution. Where no such structure exists, general numerical solvers such as TRAC-IK [60] are used instead.

In the works cited, the two lines of work serve different purposes. Analytical inverse kinematics is used to find, or to choose among, the configurations that reach a given pose. Constrained planning treats the manifold as an implicitly defined set to be approximated. The present thesis combines the two. For an S-R-S manipulator carrying an object whose axis must remain vertical, the closed-form inverse solution

supplies an explicit chart of the constraint manifold, one for each wrist branch. A sampling-based planner can therefore search the manifold directly in five coordinates, without rejection, projection or tolerance. The method is less general than those reviewed above, since it depends on the kinematic structure of the particular arm and on the form of the constraint; where those conditions hold, however, it satisfies the constraint exactly with respect to the nominal kinematic model. Its construction is developed in Chapter 7, and its conditions of applicability are set out in Section 7.11.

3. Mathematical Modelling and Kinematic Analysis

This chapter develops the mathematical foundation and geometric model of the manipulator, which are the requirements for its precise control. The objective is to describe the relationship between the robot's joint variables and the pose of its end-effector in three-dimensional space. The necessary coordinate frames and homogeneous transformation matrices are defined first, the Denavit-Hartenberg convention [35] is then applied to derive the forward kinematics, and the inverse problem, whose resolution is essential for trajectory planning, is examined last.

Beyond the standard treatment, particular attention is given to the structural properties of this mechanism, and to the kinematic redundancy caused by its seven degrees of freedom. It is shown that the manipulator is of the spherical-revolute-spherical type [33], that both of its joint triples form ZYZ Euler sets, and that it satisfies Pieper's condition for a closed-form inverse solution [58]. These properties are established here as facts about the kinematics. They are what permits the reduction in planning dimension developed in Chapter 7, and their consequences for the formulation of the planning problem are established in Chapter 4.

3.1. Introduction to Robot Kinematics

Kinematics is the branch of mechanics that studies the motion of bodies without considering the forces or torques that cause it. In robotics, kinematic analysis is essential for establishing the analytical relationship between the manipulator's individual joint configurations and the Cartesian position and orientation of its end-effector in the workspace [35].

3.1.1. Joint Space and Cartesian Space

The motion of a serial robotic arm is described using two distinct spaces:

- **Joint Space:** This space is described by the vector of joint variables:

$$\mathbf{q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]^T \quad (3.1)$$

In the case of revolute joints, these variables correspond to the angular positions of the respective joints.

- **Cartesian Space:** This space is described by the position and orientation of the end-effector relative to a fixed reference frame, typically represented by a vector:

$$\mathbf{x} \in \mathbf{R}^m \quad (3.2)$$

For a manipulator operating in 3D space, m is typically 6, representing 3 positional coordinates and 3 orientation parameters.

The general problem of robot kinematics is divided into two fundamental branches:

- **Forward Kinematics (FK):** This is the process of calculating the exact pose of the end-effector (in this case the gripper) when the joint angles are known and its solution is always unique [35].
- **Inverse Kinematics (IK):** This is the reverse process: the calculation of the joint angles required for the end-effector to reach a desired pose. It is more demanding, since it may return no solution, when the target is outside the reachable workspace or cannot be attained within the joint limits, or several. For a redundant manipulator such as we are using, the solutions form a one-parameter family, which can be exploited to satisfy secondary requirements such as obstacle avoidance [33].

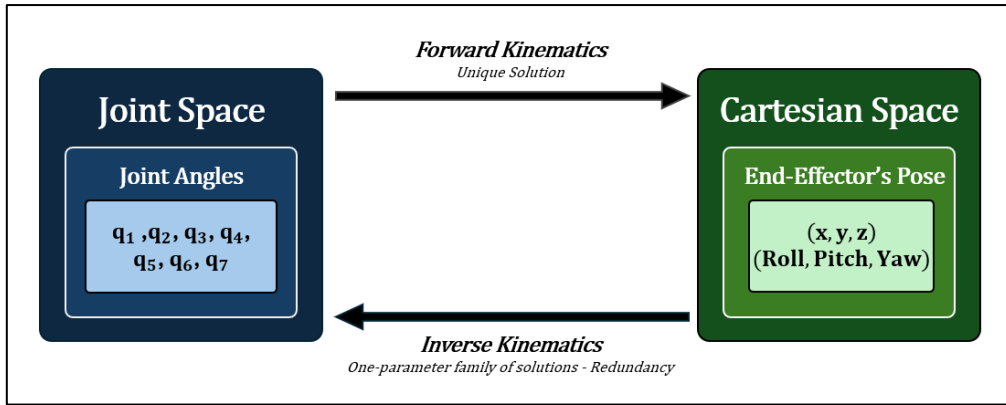


Figure 3.1 : The relationship between Joint Space and Cartesian Space. Forward Kinematics maps a Joint Configuration to a unique end-effector pose; Inverse Kinematics maps a pose back to a family of configurations

3.1.2. Description of the Robotic System

The robotic system developed in this thesis is based on the myArm 300 Pi manipulator, manufactured by Elephant Robotics; a lightweight articulated arm designed for educational purposes, which mimics the kinematic structure of the human arm [18].

The most significant feature of the myArm is its seven degrees of freedom. Unlike standard 6-axis manipulators, the seventh degree of freedom gives the system kinematic redundancy as mentioned before. This redundancy arises because the robot's seven degrees of freedom ($n = 7$) exceed the 6 degrees of freedom required to fully define any pose in 3-D space ($m = 6$).

This property offers the system significant advantages [33]:

- **Enhanced Flexibility:** The ability to reach the same target pose using different arm configurations.
- **Obstacle Avoidance:** The ability for "elbow motion" (self-motion in the null space), enabling the avoidance of collisions in constrained workspaces, such as a chemistry laboratory.
- **Singularity Avoidance:** The ability to reconfigure internally in order to avoid singular configurations. The proximity of the manipulator to such configurations is examined later in this chapter [61] and Chapter 7 describes how the system treats them.

The manipulator has a maximum reach of 300 mm as quoted by the manufacturer, measured to the mounting face - the figure relevant to the kinematic analysis is derived in Section 3.3.4 – and a payload capacity of 200 g, which the carried tube and its contents do not approach. The manipulator's constraints in this application are therefore geometric rather than dynamic.



Figure 3.2: Elephant Robotics myArm 300 Pi Robotic Arm [18]

3.2 Geometric Descriptions and Transformations

To mathematically describe the geometry of a robotic manipulator, it is essential to define Coordinate Frames attached to each link of the arm. In kinematic analysis, we are primarily concerned with describing the pose of a moving frame {B} with respect to a fixed reference frame {A}.

In the case of the myArm 300 Pi, three coordinate frames are of major importance:

- **Base Frame {0}**: Stationary and securely fixed at the base of the robot.
- **Mounting Face Frame {7}**: Attached to the mounting face of the last link. This is the frame at which the Denavit-Hartenberg chain terminates, and the frame to which the manufacturer's reach specification refers.
- **Gripper Frame {E}**: Attached to the point at which the fingers close, which is the operational working point of the manipulator. It is displaced from {7} by a fixed transformation determined by the gripper rather than by the arm.

The distinction between {7} and {E} is not a formality. The two are separated by a distance larger than the diameter of the objects being handled, so a pose specified with respect to one and analysed with respect to the other has an error larger than the tolerance of the task, causing major consequences.

3.2.1. Homogeneous Transformation Matrices

To fully describe the relationship between two coordinate frames in 3-D space, a 4×4 Homogeneous Transformation Matrix is utilised. This matrix integrates both rotational and translational information into a unified mathematical structure.

The general form of the transformation matrix T_A^B , which describes frame {B} relative to {A}, is given by the following equation:

$$T_A^B = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{P}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Where:

- **$\mathbf{R}_{3 \times 3}$ (Rotation Matrix)**: It is an orthogonal matrix describing the orientation of the axes of frame {B} relative to frame {A}.
- **$\mathbf{P}_{3 \times 1}$ (Position Vector)**: A vector $[p_x, p_y, p_z]^T$ describing the Cartesian position of the origin of frame {B} relative to frame {A}.
- **$\mathbf{0}_{1 \times 3}$ and $\mathbf{1}$** : These elements serve as a mathematical convenience, allowing affine transformations (the combination of rotation and translation) to be computed as a single linear matrix multiplication, enabling the consecutive chaining of transformations.

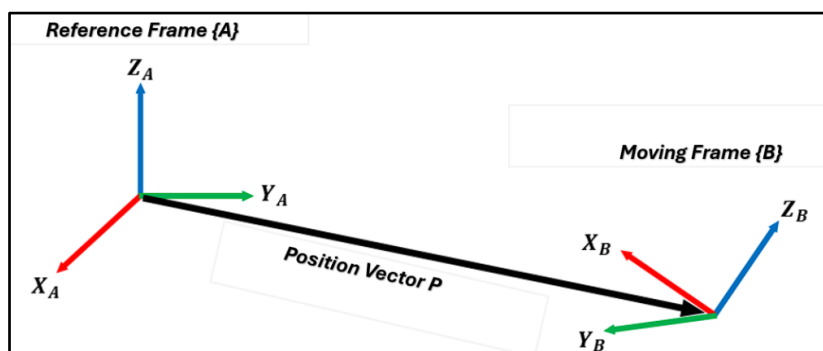


Figure 3.3 : Geometric representation of the homogeneous transformation between the reference frame {A} and the moving frame {B}. The vector P denotes the translation between the origins of the frames, while the orientation of the axes X_B, Y_B, Z_B denotes the rotation R .

3.2.2. Elementary Rotations

The rotation matrix R can be decomposed into three basic rotations around the axes x , y and z . These elementary matrices are fundamental for the calculation of the Denavit – Hartenberg parameters and are defined as follows:

- **Rotation around the x -axis by an angle γ (Roll):**

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \quad (3.4)$$

- **Rotation around the y -axis by an angle β (Pitch):**

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (3.5)$$

- **Rotation around the z -axis by an angle α (Yaw):**

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

In the case of myArm, each joint performs a rotation around its local z -axis, an operation which is geometrically described by the matrix $R_z(\alpha)$, one of several parameterisations of $SO(3)$; the choice matters, and Chapters 3 and 4 return to it.

3.2.3. Transformation Chaining and ROS Implementation

The main advantage of homogeneous transformations lies in their mathematical property of composition. If the transformation from the base frame to joint 1 (T_0^1) and from joint 1 to joint 2 (T_1^2) are known, the pose of joint 2 relative to the base is derived from the product of the two matrices:

$$T_0^2 = T_0^1 \cdot T_1^2 \quad (3.7)$$

Extending this principle for the 7-DOF manipulator, the final pose of the end-effector is calculated by sequentially multiplying all consecutive transformation matrices from the base to the mounting face. In the ROS environment, system-wide coordinate tracking is abstracted and managed by the `tf2` library [62]. This library maintains a continuous, real-time transform tree, allowing the system to query the pose of the robot relative to external reference frames, such as the camera. However, during motion planning using planners like OMPL, the high-frequency calculation of the manipulator's forward kinematics is handled by optimised kinematic solvers. These solvers rapidly compute the sequential matrix multiplications in order to evaluate large numbers of candidate configurations.

3.3. Denavit-Hartenberg (DH) Kinematic Analysis

To derive the kinematic equations of the myArm 300 Pi, the modified DH methodology is adopted. This method is the standard for the systematic geometric description of robotic systems, as it allows the determination of the position and orientation of each link through four parameters [35].

3.3.1. Parameter Definitions

According to the modified DH convention, a coordinate frame $\{i\}$ is attached to each joint i . The transformation from frame $\{i-1\}$ to frame $\{i\}$ is fully described by the following four parameters:

- **Link Length (a_{i-1}):** The minimum distance between the z_{i-1} and z_i axes, measured along their common normal, which is the x_{i-1} axis. For manipulators in which these axes intersect, this parameter is zero.

- **Link Twist (α_{i-1}):** The angle of rotation around the x_{i-1} axis required to make the z_{i-1} axis parallel to the z_i axis. It determines the relative inclination between successive joints.
- **Link Offset (d_i):** The distance from the intersection of the z_i and x_{i-1} axes to the origin of frame $\{i\}$, measured along the z_i axis. This parameter typically corresponds to the physical length of the manipulator's links.
- **Joint Angle (θ_i):** The angle of rotation around the z_i axis required to align the x_{i-1} axis with the x_i axis. In revolute manipulators this acts as the controllable joint variable (q_i).

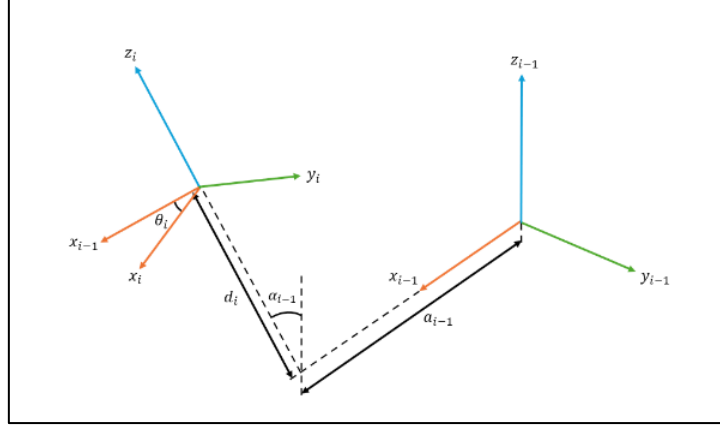


Figure 3.4: Graphical representation of the four Denavit-Hartenberg (Modified DH) parameters. The parameters a_{i-1} and α_{i-1} are defined with respect to frame $\{i-1\}$ while d_i and θ_i are defined with respect to frame $\{i\}$.

3.3.2. Geometric Characteristics of the myArm 300 Pi

To determine the DH parameters, the manufacturer's technical manuals [18] are examined. Figure 3.5 illustrates the external dimensions of the manipulator as provided by the manufacturer. These values define the effective workspace volume of the robot.

In the detailed kinematic diagram shown in Figure 3.6, the coordinate frames $\{0\}$ through $\{7\}$ are defined according to the Modified DH methodology. When the manipulator is in its upright home position, the arm extends entirely along the z -axis. Furthermore, the kinematic design dictates that consecutive joint axes (z_{i-1} and z_i) intersect orthogonally or are collinear. Because the link length parameter (a_{i-1}) measures the common normal distance between these consecutive axes, this physical intersection mathematically zeroes out the link lengths ($a_{i-1} = 0$). Therefore, the physical dimensions of the manipulator's links are represented purely as joint offsets (d_i) along their respective local z -axes.

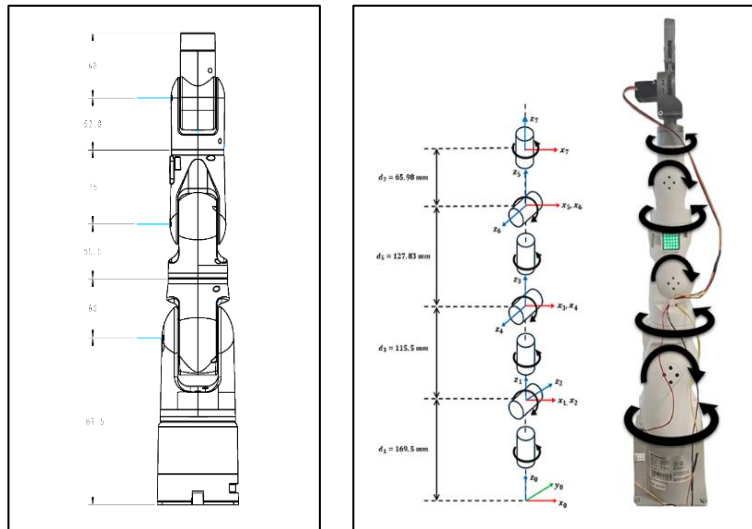


Figure 3.5 : Technical drawing illustrating the physical dimensions of the myArm 300 Pi robotic arm in millimetres [18].

Figure 3.6 : Kinematic coordinate diagram of the myArm, showing the seven rotation axes ($q_1 - q_7$), the local coordinate frames, and the link offsets (d_1, d_3, d_5, d_7) of the DH table. The axes are placed on the physical manipulator on the right.

3.3.3. DH Parameter Table

Based on the kinematic model shown in Figure 3.6, the DH Parameter Table is established. The d distance values match exactly to the physical dimensions detailed in the manufacturer's diagram [18].

Table 3.1: The DH parameter table

Link (i)	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)	θ_i (rad)
1	0	0	169.5	q_1
2	$\pi/2$	0	0	q_2
3	$-\pi/2$	0	115.5	q_3
4	$\pi/2$	0	0	q_4
5	$-\pi/2$	0	127.83	q_5
6	$\pi/2$	0	0	q_6
7	$-\pi/2$	0	65.98	q_7

Parameter Analysis:

- **Link Twists (α_{i-1}):** The presence of $\pm \pi/2$ angles confirms the orthogonal configuration of the actuators. Each motor is mounted at a right angle to the previous, enabling motion across all rotational degrees of freedom (Roll-Pitch-Yaw) of the end-effector.
- **Zero Link Lengths ($a_{i-1} = 0$):** All a parameters are zero. In the Modified DH convention, the link length is the common normal distance between two successive joint axes, so a zero value indicates that each pair of consecutive z -axes intersects or is collinear. This property is characteristic of compact manipulators. Its consequences extend beyond pairs of axes, however: combined with the zero offsets d_2 and d_6 , it causes two groups of three consecutive axes to meet at a single point each, a structural property examined in Section 3.3.4 and exploited throughout this work. The zero offset d_4 places the axes z_3 , z_4 and z_5 in concurrence at the elbow, but this is a consequence of z_3 and z_5 lying along the upper arm and the forearm respectively and does not constitute a spherical joint.
- **Vertical Offsets (d_i):** The non-zero values in the d column (169.5, 115.5, 127.83, 65.98 mm) represent the physical distances along the manipulator's axis. These values coincide with the lengths of the structural elements connecting the joints. Inversely, the zero d values (at joints 2, 4 and 6) indicate that these joints act as pivot points without adding physical length to the arm.
- **Joint Angle Offsets (θ_i):** The parameters in the θ column consist strictly of the variable joint angles (q_1 through q_7). Since the myArm is composed of revolute joints, these parameters represent the system's degrees of freedom. The absence of constant mathematical offsets in this column indicates a direct 1:1 mapping between the theoretical kinematic model and the physical hardware, so that the θ_i values correspond to the angular displacements recorded by the motor encoders, governing the dynamic posture of the manipulator.

3.3.4. Structural Properties of the Kinematic Chain

The parameter table established before, is not just a compact description of the manipulator's dimensions. It also reveals a specific structural property of the myArm that distinguishes it from an arbitrary seven-axis mechanism and that has direct consequences for the inverse kinematics problem examined later in this chapter.

Concurrency of consecutive joint axes

In the Modified Denavit-Hartenberg convention, joint i rotates about the axis z_i , and the origin of frame $\{i\}$ is located relative to the origin of frame $\{i-1\}$ by:

$$\mathbf{O}_i = \mathbf{O}_{i-1} + \mathbf{a}_{i-1} \cdot \mathbf{x}_{i-1} + \mathbf{d}_i \cdot \mathbf{z}_i \quad (3.8)$$

Because every link length of the robotic arm vanishes ($a_{i-1} = 0$ for all i), the first term disappears and each successive origin is displaced from its predecessor purely along the axis of the corresponding joint:

$$\mathbf{O}_i = \mathbf{O}_{i-1} + \mathbf{d}_i \cdot \mathbf{z}_i \quad (3.9)$$

Two consequences follow from this relation. First, if $d_i = 0$, then O_i coincides with O_{i-1} . Second, if $d_i \neq 0$ the origin O_{i-1} nevertheless still lies on the axis line z_i , since the two origins are separated along precisely that direction. Therefore, a non-zero link offset displaces the frame origin without displacing the axis line through it.

Combining the two observations yields a general criterion. Three consecutive joint axes z_i , z_{i+1} and z_{i+2} intersect at a common point whenever:

$$\mathbf{a}_i = \mathbf{a}_{i+1} = \mathbf{0} \text{ and } \mathbf{d}_{i+1} = 0 \quad (3.10)$$

in which case all three axes pass through the point $O_i = O_{i+1}$.

Applying this criterion to the DH parameter table, the condition is satisfied three times, but only two of the three correspond to spherical joints of the mechanism:

- **Joints 1,2 and 3:** Here $d_2 = 0$, so $O_2 = O_1$ and the axes z_1, z_2, z_3 all pass through O_1 . This point lies on the base axis at a height of $d_1 = 169.5$ mm and is defined as the shoulder point S.
- **Joints 5,6 and 7:** Here $d_6 = 0$, so $O_6 = O_5$, and the axes z_5, z_6, z_7 all pass through O_5 . This point is defined as the wrist centre W.
- **Joints 3,4 and 5:** Here $d_4 = 0$, so $O_4 = O_3$, and the axes z_3, z_4, z_5 pass through O_3 . This concurrence is a consequence of the limb geometry rather than a further spherical joint: z_3 is the line joining the shoulder to the elbow and z_5 the line joining the elbow to the wrist centre, so both pass through the elbow point. Since joints 3 and 5 already belong to the shoulder and wrist assemblies, and a joint cannot be shared between two spherical groups, this triple does not enter the decomposition.

The remaining joint, joint 4, therefore belongs to neither spherical assembly. Its axis z_4 passes through O_3 , which is displaced from the shoulder by $d_3 = 115.5$ mm. It acts as a single revolute elbow E separating the two spherical assemblies.

The manipulator is therefore of the S-R-S (spherical-revolute-spherical) type, a structure that mimics the human arm and that is shared by several well-known redundant manipulators [33]. This classification is exact for the myArm rather than an approximation adopted for convenience, since it follows from the vanishing link lengths and the zero offsets d_2 and d_6 alone.

Table 3.2: Decomposition of the kinematic chain into spherical shoulder, revolute elbow and spherical wrist

Assembly	Joints	Axes	Common Point	Distance from previous point
Spherical shoulder	q_1, q_2, q_3	z_1, z_2, z_3	$S = O_1$	169.5 mm above the base
Revolute elbow	q_4	z_4	$E = O_3$	$L_1 = d_3 = 115.5$ mm
Spherical wrist	q_5, q_6, q_7	z_5, z_6, z_7	$W = O_5$	$L_2 = d_5 = 127.83$ mm

The geometry is easily misread, and the error is worth explaining because it leads to the opposite conclusion. Both joint triples contain a non-zero link offset that appears to break the intersection: the

shoulder triple carries $d_3 = 115.5$ mm and the wrist triple carries $d_7 = 65.98$ mm. Neither offset displaces the corresponding axis line, because each is directed along the axis of the joint to which it belongs.

The correct test is therefore not whether the offsets vanish, but whether the mutual distance between consecutive axis lines, measured along their common normal, vanishes. That distance is exactly the link length a_{i-1} , and it is zero throughout the chain. Testing the offsets gives the wrong answer, but testing the common normals gives the right one.

Consequences for the Kinematic Problem

Two properties follow from the S-R-S structure, and together they determine the treatment of the inverse kinematics problem.

The wrist centre depends only on the first four joints. Since joints 5,6 and 7 rotate the tool about the fixed-point W , they alter the orientation of the end-effector and nothing else. The position of the wrist centre is consequently a function of q_1 through q_4 alone,

$$W = f(q_1, q_2, q_3, q_4) \quad (3.11)$$

so, the problem of positioning the wrist centre is four-dimensional, embedded within the seven-dimensional configuration space of the complete manipulator.

The shoulder-to-wrist distance depends only on the elbow. A spherical shoulder can reorient the triangle formed by the shoulder, elbow and wrist centre, but it cannot change the shape of that triangle, since the two sides L_1 and L_2 are fixed link dimensions. The distance $|W - S|$ is therefore governed by the elbow angle alone and follows from a single application of the law of cosines:

$$|W - S| = \sqrt{(L_1^2 + L_2^2 + 2 \cdot L_1 \cdot L_2 \cdot \cos q_4)} \quad (3.12)$$

with the sign convention fixed by the home posture, in which the arm is fully extended along the base axis and the two links are collinear. At $q_4 = 0$ the expression reduces to $L_1 + L_2 = 243.3$ mm, the maximum distance the wrist centre can attain. The corresponding lower bound is set by the mechanical travel limit of Joint 4. For the myArm the limit declared in the URDF file is $|q_4| \leq 1.972$ rad, which results in a minimum separation of 134.7 mm. The wrist centre is therefore confined to a spherical shell of inner radius 134.7 mm and outer radius 243.3 mm, centred on the shoulder point.

This figure should not be confused with the 300 mm maximum reach quoted by the manufacturer. The datasheet value is measured to the mounting face, which lies a further $d_7 = 65.98$ mm beyond the wrist centre, giving $115.5 + 127.8 + 65.98 \cong 309$ mm from the shoulder point to the mounting face along the fully extended axis. This exceeds the manufacturer's quoted 300 mm by roughly 3 %, most likely because the datasheet figure is measured from a different reference point. The derived values are used throughout this work, since they follow directly from the kinematic model against which all planning is performed.

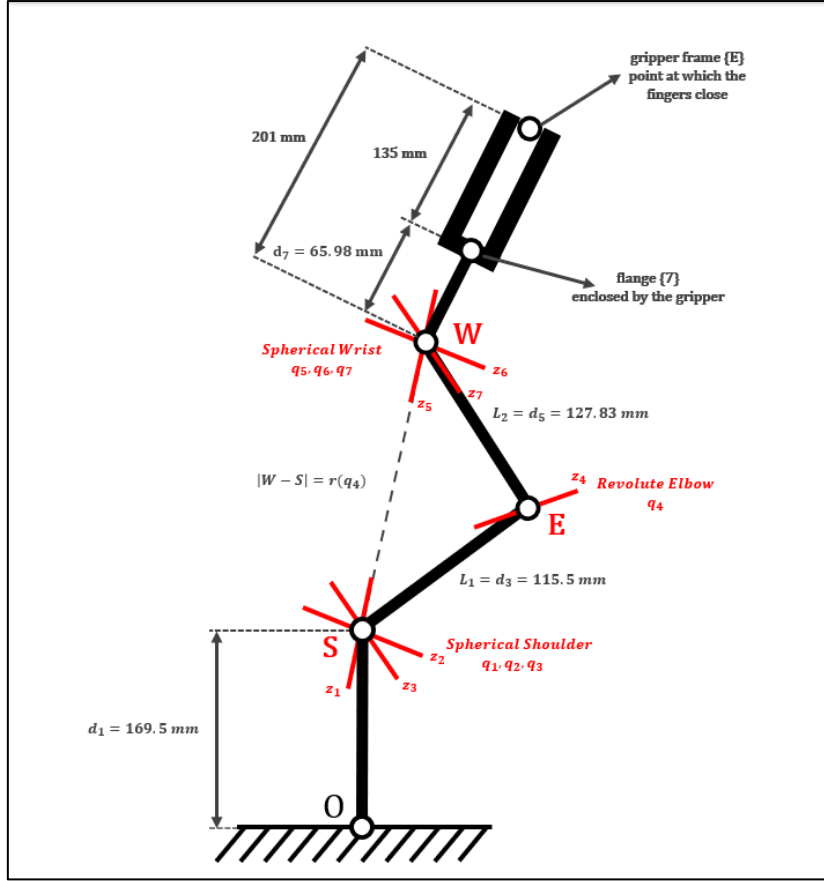


Figure 3.7: S-R-S structure of the myArm. Joint axes z_1, z_2, z_3 intersect at the Shoulder point S , axes z_5, z_6, z_7 intersect at the wrist centre W . The single revolute elbow axis z_4 passes through E . Because the link lengths L_1 and L_2 are fixed, the distance $|W - S|$ depends on the elbow angle q_4 . The mounting face $\{7\}$, at which the DH chain terminates, lies $d_7 = 65.98$ mm from the wrist centre and the gripper assembly extends a further 135 mm to the gripper frame $\{E\}$, at which task poses are specified, so the fingers close 201 mm from the wrist centre. Dimensions in millimetres

The structural properties established here are exploited in two distinct ways in the rest of this work. In Section 3.5 they provide the geometric basis for the kinematic decoupling of the position and orientation sub-problems, allowing the inverse kinematics to be treated analytically rather than numerically. In Chapter 7 they permit the dimension of the space searched by the motion planner to be reduced below that of the joint space itself, without loss of exactness.

3.4. Forward Kinematics

Forward Kinematics solves the problem of determining the pose (position and orientation) of the end-effector in the Cartesian space when the joint angles of the manipulator are known.

$$\mathbf{q} = [q_1, q_2, \dots, q_7]^T \quad (3.13)$$

3.4.1. Links Transformation Matrices

By substituting the numerical values from the DH Parameter Table into the general Modified DH transformation formula:

$$\mathbf{T}_{i-1}^i = \begin{bmatrix} \cos q_i & -\sin q_i & 0 & a_{i-1} \\ \sin q_i \cos \alpha_{i-1} & \cos q_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -d_i \sin \alpha_{i-1} \\ \sin q_i \sin \alpha_{i-1} & \cos q_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & d_i \cos \alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

the individual transformation matrices for each link of the myArm are derived.

(We use the following symbolisms: $c_i = \cos q_i$, $s_i = \sin q_i$).

1. Transformation T_0^1 (Base $\rightarrow$ Link 1): $\alpha_0 = 0$, $a_0 = 0$, $d_1 = 169.5$, $\theta_1 = q_1$

$$T_0^1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 169.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

2. Transformation T_1^2 (Link 1 $\rightarrow$ Link 2): $\alpha_1 = \frac{\pi}{2}$, $a_1 = 0$, $d_2 = 0$, $\theta_2 = q_2$

$$T_1^2 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

3. Transformation T_2^3 (Link 2 $\rightarrow$ Link 3): $\alpha_2 = -\frac{\pi}{2}$, $a_2 = 0$, $d_3 = 115.5$, $\theta_3 = q_3$

$$T_2^3 = \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ 0 & 0 & 1 & 115.5 \\ -s_3 & -c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

4. Transformation T_3^4 (Link 3 $\rightarrow$ Link 4): $\alpha_3 = \frac{\pi}{2}$, $a_3 = 0$, $d_4 = 0$, $\theta_4 = q_4$

$$T_3^4 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

5. Transformation T_4^5 (Link 4 $\rightarrow$ Link 5): $\alpha_4 = -\frac{\pi}{2}$, $a_4 = 0$, $d_5 = 127.83$, $\theta_5 = q_5$

$$T_4^5 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & 1 & 127.83 \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.19)$$

6. Transformation T_5^6 (Link 5 $\rightarrow$ Link 6): $\alpha_5 = \frac{\pi}{2}$, $a_5 = 0$, $d_6 = 0$, $\theta_6 = q_6$

$$T_5^6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.20)$$

7. Transformation T_6^7 (Link 6 $\rightarrow$ Link 7): $\alpha_6 = -\frac{\pi}{2}$, $a_6 = 0$, $d_7 = 65.98$, $\theta_7 = q_7$

$$T_6^7 = \begin{bmatrix} c_7 & -s_7 & 0 & 0 \\ 0 & 0 & 1 & 65.98 \\ -s_7 & -c_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.21)$$

3.4.2. Final Pose and Orientation Matrix

The overall kinematic chain is established by sequentially multiplying the following transformation matrices:

$$\mathbf{T}_0^7 = \mathbf{T}_0^1 \cdot \mathbf{T}_1^2 \cdot \mathbf{T}_2^3 \cdot \mathbf{T}_3^4 \cdot \mathbf{T}_4^5 \cdot \mathbf{T}_5^6 \cdot \mathbf{T}_6^7 \quad (3.22)$$

The result is a 4×4 homogeneous transformation matrix that contains the Cartesian position vector (p_x, p_y, p_z) and the rotation matrix $\mathbf{R}$:

$$\mathbf{T}_0^7 = \begin{bmatrix} \mathbf{r}_{11} & \mathbf{r}_{12} & \mathbf{r}_{13} & \mathbf{p}_x \\ \mathbf{r}_{21} & \mathbf{r}_{22} & \mathbf{r}_{23} & \mathbf{p}_y \\ \mathbf{r}_{31} & \mathbf{r}_{32} & \mathbf{r}_{33} & \mathbf{p}_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.23)$$

The product above terminates at frame $\{7\}$, the mounting face. The pose of the gripper frame $\{E\}$, at which task targets are specified, follows by composition with the fixed transformation:

$$\mathbf{T}_0^E(\mathbf{q}) = \mathbf{T}_0^7(\mathbf{q}) \cdot \mathbf{T}_7^E \quad (3.24)$$

Because of the system's seven degrees of freedom, the analytical expansion of the terms inside this final matrix results in highly complex trigonometric expressions. For this reason, the explicit mathematical derivation is generally avoided in manual calculations. Instead, the calculation of the end-effector's pose is performed algorithmically by the kinematics solvers of the planning framework.

Since $\mathbf{T}_7^E$ carries no joint variable, it contributes nothing to the differential kinematics of Section 3.6 and does not affect the structural properties of Section 3.3.4. It only matters considerably where a numerical pose is concerned.

3.4.3 Euler Structure of the Shoulder and Wrist Triples

The concurrency of the joint axes established before, in Section 3.3.4, is a statement about position. A second property, concerning orientation, follows from the twist parameters of the same table and completes the geometric representation.

From the Modified DH transformation, the rotational part of each link transformation is:

$$\mathbf{R}_{i-1}^i = \mathbf{R}_x(\mathbf{a}_{(i-1)}) \cdot \mathbf{R}_z(\mathbf{q}_i) \quad (3.25)$$

so that the rotation produced by the shoulder triple is

$$\mathbf{R}_0^3 = \mathbf{R}_z(\mathbf{q}_1) \cdot \left[\mathbf{R}_x\left(\frac{\pi}{2}\right) \mathbf{R}_z(\mathbf{q}_2) \mathbf{R}_x\left(-\frac{\pi}{2}\right) \right] \cdot \mathbf{R}_z(\mathbf{q}_3) \quad (3.26)$$

The bracketed product is a similarity transformation: conjugating a rotation by $\mathbf{R}_x\left(\frac{\pi}{2}\right)$ rotates its axis by the same amount. Since $\mathbf{R}_x\left(\frac{\pi}{2}\right)$ maps the unit vector $\hat{z}$ onto $-\hat{y}$, the middle term is a rotation about the negative y-axis, and the expression simplifies to:

$$\mathbf{R}_0^3 = \mathbf{R}_z(\mathbf{q}_1) \cdot \mathbf{R}_y(-\mathbf{q}_2) \cdot \mathbf{R}_z(\mathbf{q}_3) \quad (3.27)$$

which is a ZYZ Euler set in the three shoulder variables. The same reduction applies to the wrist triple, where the alternating twists give:

$$\mathbf{R}_4^7 = \mathbf{R}_x\left(-\frac{\pi}{2}\right) \cdot \mathbf{R}_z(\mathbf{q}_5) \cdot \mathbf{R}_y(-\mathbf{q}_6) \cdot \mathbf{R}_z(\mathbf{q}_7) \quad (3.28)$$

that is a ZYZ Euler set preceded by a constant frame rotation that carries no joint variable and can be absorbed into the reference frame.

The sign of the middle angle in each expression depends on the sign convention adopted for the twist parameters. The forms above correspond to the DH frames of Section 3.3.3 and differ by fixed rotations from the frames used in the URDF description of the robot model.

The significance is that the ZYZ Euler decomposition is one of the classical parameterisations of $SO(3)$ for which the inverse problem admits an explicit solution. Given a desired rotation matrix, the three angles are recovered directly through inverse trigonometric functions, with two solution branches distinguished by the sign of the middle angle. Both the shoulder and the wrist of the myArm therefore possess closed-form orientation inverses.

3.5. Inverse Kinematics

Inverse Kinematics (IK) deals with the complex problem of determining the specific joint angles (q) required to position the end-effector at a desired pose in Cartesian space. Unlike forward kinematics, which always results in a single, straightforward result, the solution to the inverse problem is highly complex, non-linear, and often non-unique [33].

3.5.1. Kinematic Redundancy

The myArm features seven degrees of freedom ($n=7$), while only six degrees of freedom ($m=6$) are required to fully define the position and orientation of an object in three-dimensional space. This property makes the manipulator kinematically redundant, as the number of available joints exceeds the required task-space dimensions ($n > m$).

Kinematic redundancy dictates that there is a one-parameter family of solutions (combinations of joint angles) to achieve the same end-effector pose. This provides the system with significant advantages:

- **Physical Significance (Elbow Motion):** The robotic arm possesses the physical capability to alter its internal structural configuration, specifically by moving its elbow position, while keeping the end-effector completely stationary at the target point [33]. For a manipulator of the S-R-S type, this internal motion is not arbitrary but follows a single, well-defined path. With the wrist centre held fixed, the shoulder and wrist points remain stationary, and the elbow is constrained to trace a circle about the axis joining them. The infinite family of solutions is therefore parameterised by one scalar quantity, conventionally termed the arm angle ψ , which specifies the position of the elbow along that circle. Selecting a value of ψ resolves the redundancy completely, so that the remaining joint values are determined by the target pose alone.
- **Null Space Motion:** This internal reconfiguration, which does not disrupt the primary task (the end-effector's pose), is mathematically described as motion within the system's null space. In advanced robotics, null space motion is heavily exploited to execute secondary objectives, such as obstacle avoidance in cluttered environments or joint torque optimisation.

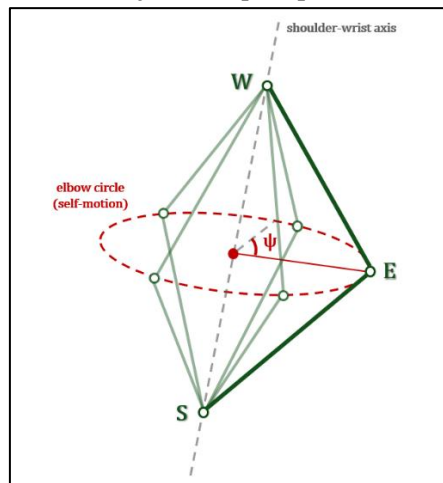


Figure 3.8: Self-motion of the redundant degree of freedom. With S and W fixed, the elbow E traces a circle about the shoulder-wrist axis, parameterised by the arm angle ψ . All the faded postures yield the identical end-effector pose.

3.5.2. Numerical Solution Method

For a manipulator of arbitrary geometry, the inverse kinematics equations are highly non-linear and admit no closed-form solution in the general case. An explicit solution exists only when the structure of the mechanism satisfies particular geometric conditions, examined later, in Section 3.5.3. Kinematic redundancy complicates the problem, since the solution is not just difficult to express but non-unique, so that any explicit formulation must be accompanied by a criterion for selecting among infinitely many candidates. For these reasons the general-purpose solver employed by the planning framework is numerical, resolving the inverse problem iteratively and in a uniform manner for any requested pose. The standard approach is based on differential kinematic control utilising the Jacobian matrix J . The fundamental relationship connecting the joint velocities $\dot{q}$ to the end-effector's Cartesian velocity $\dot{x}$ is given by:

$$\dot{x} = J(q)\dot{q} \quad (3.29)$$

Because the Jacobian matrix J for a 7-DOF redundant manipulator is not square (it is a 6×7 matrix), it cannot be directly inverted. Instead, the general solution is obtained using the Moore-Penrose pseudoinverse $J^\dagger$:

$$\dot{q} = J^\dagger \dot{x} + (I - J^\dagger J)\dot{q}_0 \quad (3.30)$$

This kinematic equation consists of two components:

1. $J^\dagger \dot{x}$: The minimum-norm solution that calculates the required joint velocities to drive the end-effector accurately to the primary target pose.
 2. $(I - J^\dagger J)\dot{q}_0$: The projection into the null space. This term permits the continuous optimisation of secondary objectives (e.g. joint limit avoidance, singularity avoidance, or posture optimisation) by applying an arbitrary joint velocity vector $\dot{q}_0$, without altering the primary pose of the end-effector.
- Within the context of this thesis, these complex, computationally heavy matrix operations are not computed manually. Instead, they are executed algorithmically and in real-time by the TRAC-IK [60], an IK solver, which is integrated into the MoveIt motion planning framework of the ROS.

3.5.3 Conditions for a Closed-Form Solution

The numerical treatment described in Section 3.5.2 is general: it makes no assumption about the geometry of the mechanism, so it is applicable to any manipulator. Its generality is also its cost, since it proceeds by iteration and depends on an initial estimate. It is worth asking whether, for the particular geometry of the myArm, the analytical route is available after all.

Whether a manipulator admits a closed-form inverse solution is not a matter of algebraic persistence but a property of its geometry. The classical result is due to Pieper [58], who showed that a sufficient condition for a closed-form solution is that three consecutive revolute axes either intersect at a common point or are mutually parallel. When this condition is met, the inverse kinematics problem separates into two smaller problems, each of which can be solved explicitly.

Kinematic Decoupling

The separation exploited here is the one produced by a spherical wrist. Because the last three axes intersect at the wrist centre W , rotating those joints moves the tool frame about W but leaves W itself unchanged. The position of the wrist centre can therefore be obtained from the desired end-effector pose alone, without knowing the wrist joint values, by retreating along the tool axis:

$$W = p_7 - d_7 \cdot \hat{z}_7 \quad (3.31)$$

where p_7 is the position of the mounting face, frame $\{7\}$, and $\hat{z}_7$ its axis direction. A target pose is normally specified at the gripper frame $\{E\}$ rather than at the mounting face and must first be mapped

back through the fixed transformation between the two frames, which depends on the gripper geometry rather than on the arm. The retraction distance applied in practice is therefore the sum of d_7 and that tool offset: 65.98 mm and 135 mm respectively, or 201 mm in total. Once the wrist centre is known, the problem decomposes into two independent sub-problems:

- **A position sub-problem:** Determine the proximal joints that place the wrist centre at W
- **An orientation sub-problem:** Determine the wrist joints that produce the desired tool orientation, given the orientation already contributed by the proximal joints.

The second sub-problem is well posed precisely because a spherical wrist can realise any residual rotation. The proximal joints do influence the final tool orientation, since they rotate the frame in which the wrist acts; but that contribution is known once the position sub-problem has been solved, and the wrist can always cancel it. This is why the decoupling holds exactly rather than merely approximately. The two preceding sections have established that both requirements are met by this manipulator, and met exactly:

- 3.3.4 showed that joints 1-3 concur at the shoulder point S and joints 5-7 concur at the wrist centre W, so Pieper's condition is satisfied at both the shoulder and the wrist.
- 3.3.4 further showed that the shoulder-to-wrist distance is governed by the elbow angle alone, so q_4 follows from the law of cosines once W is known.
- 3.4.3 showed that both joint triples form ZYZ Euler sets, so each admits an explicit inversion through inverse trigonometric functions.

The position sub-problem is therefore solved by the shoulder and elbow, and the orientation sub-problem by the wrist, both in closed form.

One qualification is required, and it is a consequence of the redundancy. Placing a point in space requires three coordinates, whereas four joints ($q_1 - q_4$) contribute to the position of the wrist centre. The position sub-problem is consequently under-determined: its solutions form the one-parameter family described by the arm angle, along which the elbow sweeps a circle while W remains fixed.

A closed-form solution is therefore obtained for each fixed value of the redundant parameter. Once it is specified, the remaining joints follow explicitly, and only a finite set of solutions remains, generated by three sign ambiguities in the inverse trigonometric functions: the elbow arccosine, which admits both signs and gives the "elbow-up" and "elbow-down" postures, and the middle angle of each of the two ZYZ inversions of Section 3.4.3. The candidates are therefore enumerable, and the appropriate one is selected by applying the joint limits and choosing the configuration closest to the current posture.

The practical consequences significantly distinguish this route from the numerical one. A closed-form solution requires no initial estimate and cannot fail to converge: it executes in constant time, it is exact to machine precision rather than to a convergence tolerance and it returns the complete set of solutions rather than the single solution nearest a seed. The last of these is the most useful in the motion planning strategy of Chapter 7, and the division of labour between the analytical and the general numerical solver is the subject of Section 4.7.1.

3.6. Jacobian Analysis and Singularities

While forward kinematics describes the positional relationship between the joints and the end-effector, differential kinematics studies the relationship between the joint velocities and the end-effector's velocity in Cartesian space. The central mathematical tool for this analysis is the Jacobian Matrix.

3.6.1. The Jacobian Matrix

The Jacobian Matrix J is a matrix of partial derivatives that linearises the velocity relationship around the current operating configuration q . The fundamental equation is given by:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (3.32)$$

Where:

- $\dot{\mathbf{x}} \in \mathbb{R}^6$: The Cartesian velocity vector of the end-effector, comprising both linear ($\mathbf{v}$) and angular ($\boldsymbol{\omega}$) velocity.
- $\dot{\mathbf{q}} \in \mathbb{R}^7$: The velocity vector of the seven joints.
- $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6 \times 7}$: The geometric Jacobian matrix.

In the case of the myArm, since the robot possesses $n=7$ joints, the Jacobian matrix $\mathbf{J}$ has dimensions of 6×7 . This is the differential expression of the kinematic redundancy established before: the null space of $\mathbf{J}$ is non-trivial, and its dimension — one at a regular configuration, and larger where $\mathbf{J}$ loses rank — is the number of internal freedoms available at that configuration.

For a manipulator composed only of revolute joints, the geometric Jacobian matrix can be constructed column-by-column.

The i -th column of the Jacobian, $\mathbf{J}_i$, which relates the velocity of the joint i to the end-effector velocity, is computed as:

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{z}_i \times (\mathbf{p}_e - \mathbf{p}_i) \\ \mathbf{z}_i \end{bmatrix} \quad (3.33)$$

Where:

- $\mathbf{z}_i$ is the unit vector along the axis of rotation for joint i (extracted from the rotation block of T_0^i).
- $\mathbf{p}_i$ is the position vector of the origin of frame $\{i\}$ (extracted from the translation block of T_0^i)
- $\mathbf{p}_e$ is the position vector of the end-effector (extracted from T_0^E).

3.6.2. Kinematic Singularities

One of the most critical challenges in the control of robotic manipulators is the occurrence of Kinematic Singularities. A singularity is defined as a specific robot configuration $\mathbf{q}$ in which the Jacobian matrix loses its full rank (becomes rank-deficient). For the myArm, whose Jacobian has six rows, this condition reads:

$$\text{rank}(\mathbf{J}) < 6 \quad (3.34)$$

When the manipulator approaches or enters a singular configuration, the following physical effects occur:

- **Loss of Degrees of Freedom:** The end-effector loses the ability to move in one or more specific Cartesian directions.
- **Infinite Joint Velocities:** To produce a finite Cartesian velocity in the restricted or "forbidden" direction, theoretically infinite joint velocities ($\dot{\mathbf{q}} \rightarrow \infty$) are required, which can lead to hardware failure or erratic behaviour.

Singularities are classified into two categories:

- **Boundary Singularities:** These occur when the manipulator is fully extended or completely folded at the absolute limit of its workspace. In such a posture, the robot cannot move any further outward.
- **Internal Singularities:** These occur within the reachable workspace, typically when two or more rotational axes become collinear. For the myArm the characteristic case is the "wrist lock" condition at $q_6 = 0$: the middle wrist joint is what separates the axes of joints 5 and 7, so when it passes through zero those axes become collinear and the wrist loses the ability to rotate the tool about the direction perpendicular to them. The two joints then produce identical motion, and the contribution of each can no longer be distinguished.

3.6.3. Manipulability Analysis

To numerically evaluate the robot's capacity to move evenly in all directions from a given posture, Yoshikawa's Manipulability Measure (w) is utilised [61]:

$$\mathbf{w} = \sqrt{\det(\mathbf{J}(\mathbf{q})\mathbf{J}(\mathbf{q})^T)} \quad (3.35)$$

The value of w drops to zero precisely at singular configurations. A redundant manipulator such as the myArm can reconfigure internally in order to increase this measure while holding the end-effector pose fixed, an option unavailable to a conventional six-axis robot.

The redundant degree of freedom is what makes such reconfiguration possible, and null-space optimisation of w is the classical means of exploiting it. The present system resolves the redundancy differently: rather than driving the arm angle by a secondary objective, it treats it as an explicit coordinate, enumerated over when generating goal configurations and searched over during planning. The reasoning behind that choice, and the criteria that replace a manipulability objective, are given in Chapter 7.

4. Problem Formulation

Chapter 3 established the kinematic framework of the manipulator: the transformations that relate its joints to the pose of its end-effector, the structural properties of its mechanism, and the conditions under which those relations may be inverted. That framework describes what the arm can do. It does not describe what the arm is being asked to do, which determines the difficulty of the task.

This chapter states the problem. It describes the physical transfer the system must perform, identifies the requirement that distinguishes it from a conventional pick-and-place operation, and expresses that requirement mathematically as a condition on the set of configurations through which the manipulator may pass. It also characterises the resulting search problem, its dimension, the structure of the set to be searched, explains why that structure is the source of the difficulty, and specifies the dynamic and geometric parameters within which any solution must operate.

The chapter focuses exclusively on the statement of the problem. The planning architecture that solves it, the costs of each formulation, and the reduction that is the principal contribution of this work are the subject of Chapter 7. The purpose here is to define precisely the object that Chapter 7 searches, so that the claims made there can be read against a requirement fixed in advance instead of one developed in parallel with the method that satisfies it.

4.1 The Task and its Governing Requirement

The difficulty of this task does not lie where it might be expected. The motions are short, the stations are fixed and known during the execution of the transfers, and the workspace is neither large nor cluttered by the standards of industrial manipulation. What makes the problem hard is a single requirement arising from the payload, which converts a routine sequence of transfers into a constrained search.

4.1.1 The Physical Transfer

The system performs liquid transfers between fixed stations on a laboratory bench. Five bases hold five test tubes, each 100 mm long and 16 mm in diameter, standing upright with a spacing of approximately 30 – 40 mm between neighbours. A mixing vessel of 95 mm height and 75 mm diameter is positioned nearby. The manipulator is the myArm, the 7-DoF arm described above, with a gripper attached to its mounting face.

A single transfer consists of the following stages. The arm travels from its current configuration to a standoff pose above a designated tube, it descends vertically to the grasp height and closes the gripper on the tube, it lifts the tube clear of the base, it carries the tube to the mixing vessel, it tips the tube to discharge its contents, and it returns the tube to the base, descending, releasing and retreating. The transfer is therefore not a single motion but a sequence of motions of several kinds, with different requirements from stage to stage.

Two features of the physical arrangement bear directly on the difficulty of the problem and are recorded here because they are properties of the task rather than of any method chosen to address it. The first is one of scale: the carried tube is 100 mm long on an arm whose reach is 243 mm, so the payload is not small relative to the manipulator carrying it, and the gripper that holds it extends further still beyond the wrist. The second is one of clearance: with neighbouring tubes 30 – 40 mm apart, the free space around a grasped tube is small, and the margin available to any motion in the neighbourhood of the bases is correspondingly narrow.

4.1.2 The Sources of Difficulty

The execution of this task is constrained by three elements, which interact:

1. **The Kinematics of the Manipulator:** The pose of the end-effector in Cartesian space is an element of $SE(3)$ so it is specified by six quantities, while the arm provides seven joints. The manipulator is redundant: a given end-effector pose is attained not by one joint configuration but by a one-

parameter family of configurations, along which the arm may reconfigure internally while holding the tool still. In a workspace containing obstacles this is an advantage, since it supplies alternative postures from which to reach the same target. However, it is also a cost, as the additional freedom must be resolved, either by a choice made explicitly or by whatever choice a solver happens to make, and a configuration that reaches the target correctly may still be unusable if it lies against a joint limit or in a posture from which the next motion cannot proceed.

2. **The Spatial Restrictions of the Workspace:** The bench, the bases with the tubes, the mixing vessel and adjacent fixtures occupy the space through which the arm must move, and the spacing noted above is small in relation to the carried payload. Any configuration the manipulator adopts must be free of collision, not only at the endpoints of a motion but throughout it.
3. **The Instability of the Payload:** The tube is open and contains liquid. This is the element that separates the present problem from a conventional pick-and-place operation, and it acts in two different ways. It restricts both the orientation of the tube throughout the motion and the accelerations applied to the tube, since inertial forces displace the free surface of the liquid even when the tube remains upright.

4.1.3 The Requirement as a Path Constraint

The requirement imposed by the liquid is that the axis of the tube remain vertical while the tube is carried. What matters for the difficulty of the problem is not the requirement itself, but where it applies. A conventional manipulation task specifies the configuration at which a motion begins and the configuration at which it ends and leaves the intervening motion unspecified. The planner is free to reach the goal by any collision-free route. Such a specification is a goal constraint. If the upright requirement was of this kind, it would be satisfied without effort. Holding the tube upright in both the initial and the final pose is easily accommodated by the arm's seven degrees of freedom, and any motion between them would be tolerable.

However, it is not of this kind. Liquid spills at the moment the tube tips, irrespective of the final orientation of the tube, so the requirement must hold at every moment of the motion and not only at its endpoints. It is a path constraint. This distinction is the central fact of the problem addressed in this thesis, and its consequence is that the set of configurations available to the planner is no longer the whole of the joint space but a subset of it, one which, as Section 4.3 shows, is small in a sense that can be made precise.

The requirement is also directional rather than uniform. A tube is a body of revolution, so rotating it about its own axis moves neither its contents nor their centre of mass relative to gravity, and that rotation may be left entirely free. What must be restricted are the two directions in which the tube tips away from the vertical. The requirement is therefore anisotropic: two of the three orientational degrees of freedom of the tool are constrained and one is not. The formal expression of this is given in Section 4.2, where the free axis is fixed relative to the tool frame. This is of particular consequence, as it is not the axis suggested by conventional labelling.

4.1.4 Statement of the Problem

The problem addressed by the control architecture can be expressed as follows:

1. The system must produce motions along which the tube remains upright at every configuration, not merely at the configurations at which those motions begin and end, and it must do it while avoiding collision inside a workspace whose gaps are small in relation to the carried payload.
2. The system must therefore resolve the kinematic redundancy of the 7-DoF manipulator, identifying configurations that satisfy the immediate requirements while guaranteeing the execution of the remaining trajectory.

3. The system must accomplish both within a time acceptable to an interactive operator, since the arm plans each stage of a transfer in response to a command rather than replaying a trajectory prepared offline.

The three are not independent, and the tension between them makes the problem non-trivial: the first restricts the configurations available, the second must choose among what the first leaves, and the third bounds the effort that may be spent doing so. The following sections express the first of these mathematically and establish why one of these dominates the cost.

4.2 Formal Statement of the Constrained Planning Problem

The requirement identified in Section 4.1 is stated informally as a condition on the tube. To reason about its cost, it must be expressed as a condition on the manipulator, and specifically as a condition on the set of joint configurations the planner is permitted to visit. This section constructs that set. Nothing here assumes any particular method of searching it. The definitions are made regarding the manipulator and the task alone, so that the formulations compared in Chapter 7 may be measured against a common requirement.

4.2.1 Configuration Space and Forward Kinematics

The manipulator has seven revolute joints, whose values are collected in the vector $\mathbf{q} = (q_1, \dots, q_7)^T$. Each joint is bounded by the mechanical limits of the arm, so the configuration space is the box:

$$\mathbf{Q} = \{\mathbf{q} \in \mathbb{R}^7: \mathbf{q}_i^{\min} \leq \mathbf{q}_i \leq \mathbf{q}_i^{\max}, i = 1, \dots, 7\} \quad (4.1)$$

The forward kinematics established in Chapter 3 map a configuration to the pose of the gripper frame $\{E\}$:

$$\mathbf{f}(\mathbf{q}) = \mathbf{T}_0^E(\mathbf{q}) = (\mathbf{p}(\mathbf{q}), \mathbf{R}(\mathbf{q})) \quad (4.2)$$

where $\mathbf{p}(\mathbf{q}) \in \mathbb{R}^3$ is the position of the point at which the fingers close and $\mathbf{R}(\mathbf{q}) \in SO(3)$ the orientation of that frame, both expressed in the world frame.

4.2.2 The Tube Axis and the Tilt Function

The requirement concerns the orientation of the carried tube, which is fixed rigidly in the gripper and therefore has a constant direction in the tool frame. Let $\hat{\mathbf{a}}$ denote the unit vector along the tube's long axis, directed from its closed base towards its opening, expressed in the tool frame. For this manipulator and gripper:

$$\hat{\mathbf{a}} = \hat{\mathbf{x}} = (\mathbf{1}, \mathbf{0}, \mathbf{0})^T \quad (4.3)$$

This identification is stated directly because it is not the one the conventional labelling of a tool frame would suggest. The z axis of a tool frame is conventionally taken to be its approach direction, and it is natural to assume that a tube held by the gripper lies along it. Here it does not: the tube is grasped transversely to that convention, and its axis coincides with the tool frame's x direction. The consequences of assuming otherwise are not subtle and are documented in Section 7.4.4.

The frame in which (4.3) holds is the gripper frame $\{E\}$ of Section 3.2. Frames $\{7\}$ and $\{E\}$ differ by a fixed translation and therefore share an orientation, so the identification is unaffected by which one is used. The position $\mathbf{p}(\mathbf{q})$ is not and differs between the two by the extent recorded in Section 7.8.2.

In the world frame the tube axis is therefore:

$$\mathbf{a}(\mathbf{q}) = \mathbf{R}(\mathbf{q})\hat{\mathbf{a}} \quad (4.4)$$

and the deviation of the tube from vertical is the angle between $\mathbf{a}(\mathbf{q})$ and the world vertical $\hat{\mathbf{z}}_w = (\mathbf{0}, \mathbf{0}, \mathbf{1})^T$:

$$\beta(\mathbf{q}) = \arccos(\mathbf{a}(\mathbf{q}) \cdot \hat{\mathbf{z}}_w), \quad \beta: \mathbf{Q} \rightarrow [0, \pi] \quad (4.5)$$

The quantity $\beta(q)$ is the single scalar in which the entire orientation requirement is expressed. Three of its properties are worth recording, since they are the reasons, this measure is used throughout the thesis in preference to the per-axis tolerances a planning framework supplies of its own accord.

It is invariant under rotation of the tube about its own axis. If the tool orientation is post-multiplied by a rotation through any angle φ about $\hat{a}$:

$$\mathbf{a}(q) \mapsto \mathbf{R}(q)\text{Rot}(\hat{a}, \varphi)\hat{a} = \mathbf{R}(q)\hat{a} = \mathbf{a}(q) \quad (4.6)$$

so β is unchanged. This is the formal expression of the anisotropy noted in Section 4.1.3: the freedom that spills nothing is invisible to the measure, exactly as it should be.

On M the rotation $\text{Rot}(\hat{a}, \varphi)$ acts in the world frame as a rotation about the vertical, since $\mathbf{R}(q)\hat{a} = \hat{z}_w$ implies $\mathbf{R}(q)\text{Rot}(\hat{a}, \varphi) = \text{Rot}(\hat{z}_w, \varphi)\mathbf{R}(q)$. The parameter φ is therefore not merely a convenient label for the residual freedom but a rotation about a fixed world axis, which is what allows it to serve as a search coordinate in Chapter 7.

It is well defined where a decomposition into per-axis deviations is not. An orientation error resolved into Euler components degenerates when one of those components is permitted to span a half-turn, which by construction is the case here: the inner product of equation (4.5) has no such degeneracy.

It is independent of the planner. β is a function of the configuration alone and makes no reference to how that configuration was produced, so it yields a number comparable across methods that enforce the requirement by different mechanisms – which is what makes the evaluation of Chapter 7 meaningful.

4.2.3 The Constraint Manifold

The set of configurations at which the tube is exactly vertical is the zero level set of the tilt function:

$$\mathbf{M} = \beta^{-1}(0) = \{q \in Q : \mathbf{a}(q) = \hat{z}_w\} \quad (4.7)$$

This set is the constraint manifold: the surface within the joint space on which the requirement of Section 4.1.3 holds identically. A motion that remains within M carries the tube upright at every configuration it passes through, and no motion that leaves M does so, however briefly it departs.

Equation (4.7) fixes only the direction of the tube axis and leaves the tool orientation undetermined to the extent described by (4.6). At any $q \in M$ the orientation $\mathbf{R}(q)$ is therefore determined up to a single rotation about the world vertical, since $\hat{a}$ is carried onto $\hat{z}_w$ and rotations about the image axis leave that image fixed. The admissible tool orientations at a given point thus form a one-parameter family rather than a single orientation – a residual freedom that the constraint leaves untouched, and whose availability is taken up in Section 4.4.

A relaxed form of the requirement is also needed, because a planner that cannot restrict itself to M exactly may still be able to restrict itself to a neighbourhood of it. For a tolerance $\varepsilon > 0$ define:

$$\mathbf{M}_\varepsilon = \beta^{-1}([0, \varepsilon]) = \{q \in Q : \beta(q) \leq \varepsilon\} \quad (4.8)$$

a band about M of angular half-width ε . The system's nominal tolerance is $\varepsilon = 0.30$ rad, approximately 17° , a planning tolerance chosen well inside the hard limit derived in Section 4.6.1. Note that $M \subset M_\varepsilon$ for every ε , and that M is recovered as $\varepsilon \rightarrow 0$.

4.2.4 Collision-Free Configurations

The workspace described in Section 4.1.1 is represented by a set of rigid obstacles $\mathcal{O}$ fixed in the world frame. Writing $\mathcal{A}(q)$ for the region of space occupied by the links of the manipulator, together with the gripper and the carried tube, when the arm is in configuration q , the collision-free subset of the configuration space is:

$$\mathbf{C}_{\text{free}} = \{q \in Q : \mathcal{A}(q) \cap \mathcal{O} = \emptyset\} \quad (4.9)$$

Two aspects of this definition are worth making explicit, since both bear on results reported later. The occupied region $\mathcal{A}(q)$ includes the carried tube, so a configuration that is admissible with an empty gripper need not be admissible while a tube is held. And C_{free} is defined against the modelled obstacles $\mathcal{O}$ rather than against the physical workspace; every guarantee obtained from it is therefore conditional on the model being faithful, a dependence examined in Section 7.8.

4.2.5 The Problem

The planning problem addressed in this thesis may now be stated. Given a start configuration q_{start} and a goal configuration q_{goal} , both lying in $M \cap C_{\text{free}}$, find a continuous path:

$$\begin{aligned}\tau : [0, 1] &\rightarrow M \cap C_{\text{free}} \\ \tau(0) &= q_{\text{start}} \\ \tau(1) &= q_{\text{goal}} \quad (4.10)\end{aligned}$$

or report that none exists.

Two features of this statement distinguish it from the planning problem ordinarily posed. The codomain is $M \cap C_{\text{free}}$ rather than C_{free} alone: the path is required to lie within the constraint manifold along its whole length, not merely to begin and end there. And τ is required to be continuous as a map into configuration space, which is a stronger condition than the requirement that each of a finite set of sampled configurations lie in $M \cap C_{\text{free}}$. The distinction between these two is not pedantic – it is the subject of Section 7.6, in which several defects are described that satisfy the sampled condition while violating the continuous one.

Where a method cannot meet (4.10) exactly, it may instead attempt the relaxed problem obtained by substituting M_ϵ for M , yielding a path along which the tube tips by no more than ϵ . The two problems differ in kind and not merely in degree, and Section 4.3 examines why.

4.3 The Constraint as the Source of the Difficulty

The problem stated in equation (4.10) differs from an ordinary motion planning problem only in that its codomain is $M \cap C_{\text{free}}$ rather than C_{free} . That is a small change to write down and a large one to solve. This section explains why, in terms of the structure of M alone and without reference to any particular planning algorithm, and it distinguishes the two kinds of solution that the difference between M and M_ϵ makes available.

4.3.1 Measure Zero and its Consequence for Sampling

The requirement $a(q) = \hat{z}_w$ is a condition on a unit vector and therefore amounts to two independent scalar equations rather than three. Writing $\hat{x}_w$ and $\hat{y}_w$ for the two horizontal directions of the world frame, define

$$h : Q \rightarrow \mathbb{R}^2, \quad h(q) = (a(q) \cdot \hat{x}_w, a(q) \cdot \hat{y}_w) \quad (4.11)$$

so that $M = h^{-1}(0)$, together with the requirement that $a(q)$ point upwards rather than downwards. Where 0 is a regular value of h , the implicit function theorem gives M the structure of a smooth submanifold of Q of codimension two, and hence of dimension $7-2=5$. That dimension is developed in Section 4.4. What matters here is the codimension.

It is preferable to argue through h rather than directly through the tilt function β of equation (4.5), because β fails to be differentiable precisely on M itself – arccos has infinite derivative at unit argument – so the zero level set of β is not one to which the regular value theorem may be applied directly. The vector-valued h has no such defect. The scalar β remains the correct measure of deviation from the constraint, which is the use to which it is put throughout; h is the correct expression of the constraint itself.

The consequence of positive codimension is immediate, and it is the whole of the difficulty: M is a set of Lebesgue measure zero within Q . A configuration drawn at random from the seven-dimensional joint space lies in M with probability zero.

Sampling-based planners operate by drawing configurations from Q and attempting to connect them. Applied to the problem of equation (4.10), such a planner will, with probability one, never draw an admissible sample at all. This is not a deficiency of any particular algorithm and cannot be remedied by sampling more densely or for longer; it is a property of the set being sampled. Two general remedies exist, and both purchase admissible samples at a price.

4.3.2 The First Remedy: Relaxation and its Cost

The first remedy is to replace M by the band M_ϵ of equation (4.8) and to discard samples falling outside it. Since M_ϵ has non-empty interior in Q , samples now land within it at a positive rate, and the planner proceeds by rejection.

The rate, however, is small, and it is instructive to estimate it. The set of directions lying within ϵ of the vertical forms a spherical cap of solid angle $2\pi(1 - \cos \epsilon)$ out of the 4π of the whole sphere, so if the tube axis $a(q)$ were distributed uniformly over the sphere of directions, the fraction of configurations admitted would be:

$$P(\beta \leq \epsilon) = \frac{(1 - \cos \epsilon)}{2} \approx \frac{\epsilon^2}{4} \text{ for small } \epsilon \quad (4.12)$$

Uniform sampling of the joint box does not induce an exactly uniform distribution of $a(q)$, so (4.12) is an idealisation rather than a prediction of the rate any particular planner will observe. It nonetheless gives the correct order of magnitude and, more importantly, the correct scaling. Evaluated at the three tolerances the system employs:

Table 4.1: Admitted fraction at each tilt tolerance

Tolerance ϵ	In Degrees	Admitted Fraction
0.30 rad	17.2°	2.2 %
0.50 rad	28.6°	6.1 %
0.70 rad	40.1°	11.8 %

Three readings follow. At the nominal tolerance roughly forty-four samples in forty-five are discarded before one is retained, and the discarded samples are not free: each must be generated and evaluated before it can be rejected. The quadratic scaling means the cost worsens rapidly as the requirement tightens – a halving of the tolerance quarters the admitted fraction – so the approach degrades in exactly the regime where the constraint matters most. And the admitted fraction tends to zero with ϵ , recovering the measure-zero result of Section 4.3.1 in the limit: relaxation does not remove the difficulty, it trades a probability of zero for one that is merely small.

The table also exhibits, in advance, the reasoning behind the graded sequence of attempts described in Section 7.2.2, in which a query that fails at 0.30 rad is retried at 0.50 rad. The second attempt is not a different method but the same method operating on a set roughly three times larger.

4.3.3 The Second Remedy: Projection and its Cost

The second remedy retains M but abandons the attempt to sample it directly. A configuration is drawn from Q as before and is then moved onto the constraint manifold by iterative root-finding on h , typically by repeated correction along the Jacobian pseudo-inverse until $\|h(q)\|$ falls below a numerical threshold. Each sample so treated is admissible, and the planner may then connect samples in the ordinary way.

This is the mechanism a general-purpose constrained planner employs, and it is the one against which the contribution of this thesis is measured. Its costs are of two kinds:

1. **It is Expensive:** In that every sample requires an iterative solve rather than an evaluation, and every edge between samples requires the same treatment at each intermediate state along it, since a straight line between two points of a curved manifold does not remain on the manifold.
2. **It is Approximate:** In that root-finding terminates on a numerical threshold rather than on exact satisfaction. The configurations produced lie near M rather than on it, so what is obtained is once again a path in some M_δ – with δ now determined by the solver’s convergence tolerance rather than chosen on physical grounds.

Neither cost is a defect of implementation. Both follow from the manifold being described only implicitly, as the level set of a function that can be evaluated but not inverted. Where an explicit description is available the costs do not arise, and this is the observation on which Chapter 7 is built.

4.3.4 Two Kinds of Guarantee

The distinction between a path lying in M and a path lying in M_ϵ is easily read as a difference of degree, as though the second were the first with an error bar attached. It is better understood as a difference in the kind of statement each supports, and the distinction is worth fixing here because the evaluation in Chapter 7 turns on it.

A path constructed within M satisfies the requirement by construction. Uprightness is a property of the space in which the search takes place, so no configuration violating it can be produced, and no test is required to establish that none was. There is no tolerance, because there is no quantity left to bound.

A path constructed within M_ϵ satisfies the requirement within a bound, subject to verification. Three qualifications attach to it, none of which attaches to the first case. The bound is a specification the planner is asked to respect rather than a property the path possesses, so whether it was respected is a question that must be answered by measurement. That measurement can be made only at the configurations actually examined, which for a sampling-based planner is a finite set of waypoints; the motion between two adjacent waypoints is not tested by it and need not inherit the bound. And the value of ϵ is itself a judgement about the payload – how full the tube is, how viscous its contents, how much margin is prudent – rather than a fact about the manipulator, so a path admissible under one filling is not necessarily admissible under another.

The practical consequence is that the two formulations require different things of the surrounding system. A planner operating on M_ϵ must be paired with an independent check on its output, and the system must have a defined response for the case in which that check fails. A planner operating on M requires neither. This difference is not visible in a success rate or a planning time, which is why the two are reported separately in Section 7.11.

Exactness, here and throughout the thesis, is relative to the nominal kinematic model of the manipulator. The manifold M is defined through the forward kinematics of equation (4.2), which are computed from the parameters of Table 3.1 and the robot description. A path within M holds the tube vertical in that model, to the precision of floating-point arithmetic. It does not follow that the physical tube is vertical. The physical tilt differs from the modelled one by the discrepancy between model and mechanism: manufacturing tolerances in link dimensions and axis alignments, the rounding of the model’s own parameters (Section 7.3), backlash in the joints (Section 9.5.3), and the seating of the tube in the printed gripper, whose axis is assumed to coincide with the tool frame’s x direction. The distinction drawn in this section is therefore between a guarantee that holds exactly in the model and one that holds only to within a tolerance in the model. Neither is a guarantee about the physical arm, which no planner can supply without measurement.

4.3.5 Alternative Explanations and their Exclusion

The argument above attributes the difficulty of the problem entirely to the codimension of M , and therefore to the constraint. Two alternative explanations must be excluded before that attribution can be relied upon, since either would account for the same observed behaviour.

The first is that the workspace itself is hard: that the obstacles are dense, the clearances tight, and the queries would be difficult even unconstrained. The second is that the difficulty is an artefact of measurement – a planner misconfigured, a tolerance set wrongly, a scene incorrectly published – rather than a property of the problem at all.

Neither can be settled by argument. Both would be addressed by the same experiment: the identical set of queries planned a second time with the orientation requirement removed altogether, so that the constraint is the only quantity differing between the two runs. Such a control is not a candidate method – it is free to tip the tube, and does – but it would separate the three hypotheses. It was not conducted in the present work (Section 9.1.5), and the attribution of the difficulty to the constraint therefore rests on the analysis of this section, a gap recorded in Section 10.2.6.

The formulation set out in this chapter is what makes such a comparison possible. Because β of equation (4.5) is defined on configurations rather than on planner internals, the same number may be computed for every method, regardless of how – or whether – it enforces the constraint at all.

4.4 The Dimension of the Admissible Set

Section 4.3 established that M is small in the sense of measure: a set of measure zero within Q , which is why it cannot be sampled directly. It is also small in a second and more useful sense. M is a five-dimensional object, and five is two fewer than the dimension of the space in which sampling-based planners ordinarily search. This section establishes that number, notes what it does and does not assert, and identifies the condition under which it can be exploited rather than merely observed.

4.4.1 Two Counts of the Same Number

The dimension of M may be obtained in two ways, which proceed from different premises and are worth setting out separately:

1. **Counting from the Constraint:** The manipulator has seven degrees of freedom. The requirement of Section 4.2.3 is a condition on the direction of a single axis, and a direction in three-dimensional space is specified by two quantities rather than three; equivalently, h of equation (4.11) has two components. Two of the three orientational freedoms of the tool are therefore fixed and one is left free, giving:

$$\dim M = 7 - 2 = 5 \quad (4.13)$$

This is the codimension argument of Section 4.3.1 restated as a count.

2. **Counting from the Geometry:** The same number arises from the structure of the mechanism, without reference to the constraint function at all. Chapter 3 established that the position of the wrist centre is determined by the four proximal joints alone, and that this positioning task is redundant: three quantities locate the wrist centre in space and a fourth, the arm angle of Section 3.5.1, describes the rotation of the elbow about the shoulder-wrist axis that leaves that position unchanged. To these four must be added the residual orientational freedom identified in Section 4.2.3 – the rotation of the tool about the world vertical, which the constraint leaves entirely unrestricted.

The total is again:

$$3(\text{wrist centre}) + 1(\text{arm angle}) + 1(\text{rotation about the vertical}) = 5 \quad (4.14)$$

That two arguments proceeding from unrelated starting points – one from the algebra of the constraint, one from the kinematic structure of the arm – arrive at the same number is worth more than either argument alone. It indicates that five is intrinsic to the problem rather than an artefact of how the

problem was described, and it means that any method claiming to search a space of this dimension is not making an approximation but matching the dimension of the set it is searching.

The quantities appearing in (4.14) have a further property that (4.13) does not exhibit: they are individually meaningful. Each is a geometric quantity with a physical interpretation – a position, an elbow rotation, a tool rotation – rather than an abstract coordinate on a level set. Section 4.4.3 returns to why this matters.

4.4.2 Dimension of the Search Space, not of the Mechanism

The claim of equation (4.13) is easily misread, and the misreading is worth forestalling at the point the claim is first made.

The manipulator has seven degrees of freedom throughout. Nothing in this chapter locks a joint, immobilises a link, or restricts the arm’s capability; all seven joints move during a constrained motion, and the trajectory that results is a trajectory in $\mathbb{R}^7$ like any other. What is five-dimensional is the set of configurations satisfying the requirement – a surface embedded in the seven-dimensional joint space, on which the arm is free to move in five independent directions rather than seven.

The reduction, where a planner achieves one, is accordingly a change of coordinates and not a simplification of the mechanism. This distinction is developed further in Section 7.5.2, at the point where a planner exploiting it is actually constructed.

A further reduction is available if one of the two redundant freedoms in (4.14) is given up. Fixing the arm angle removes one coordinate and yields a four-dimensional description, at the cost of a portion of the reachable workspace; whether that cost is acceptable is an empirical question about the particular task, and it is measured in Section 7.3.

4.4.3 From Dimension to Chart

The dimension count above is a property of M and holds irrespective of how M is described or searched. It is therefore available to every method considered in this thesis, including those that derive no benefit from it whatsoever.

This point deserves emphasis, because it locates precisely where the difficulty of the problem survives the analysis of this chapter. A projection-based constrained planner of the kind described in Section 4.3.3 operates on a manifold it knows to be five-dimensional and nonetheless draws its samples from the seven-dimensional box Q and corrects them afterwards. It does so because dimension is not what such a planner lacks. What it lacks is a chart: an explicit map from a five-dimensional parameter domain onto M , by which a point of that domain may be converted into a configuration of the arm directly, without iteration and without the possibility of failing to converge.

The distinction between the two situations is the distinction between an implicit and an explicit description of the same set. Equation (4.11) describes M implicitly; given a configuration, it can be tested for membership. It does not describe how to produce a member. A chart would do the second, and the difference is decisive for a planner, whose inner loop consists almost entirely of producing candidate configurations.

Whether such a chart exists is not a question the present chapter can settle, because it is not a question about the constraint. It is a question about the manipulator: about whether the inverse kinematics of the arm admit a closed-form solution, and whether the freedom the constraint leaves can be expressed as a coordinate rather than merely bounded. The structural properties established in Chapter 3 – the concurrency of the shoulder and wrist axes, the ZYZ form of both joint triples, and the decoupling that follows from them – are what make the question answerable for this manipulator, and the construction of the chart itself is the subject of Chapter 7.

Two remarks close the matter. The first is that the number five is a property of this constraint and not a general one: a requirement fixing all three orientational freedoms would leave a four-dimensional set, and one confining the tool axis to the surface of a cone rather than to a single direction would leave six.

The benefit obtainable from a reduction therefore depends on how much freedom the constraint leaves, and the upright requirement is close to the most favourable case. The second is that a chart, where one exists, does more than lower a dimension. It changes what the planner can guarantee, in the manner set out in Section 4.3.4 – for a search conducted entirely within the parameter domain of a chart cannot produce a configuration off the manifold, there being none to produce.

4.5 Dynamic Parameters and the Collision Model

The problem stated in Section 4.2 is purely geometric. Its solution is a path – an ordered set of configurations – and a path carries no information about how quickly it is to be traversed, or about what the manipulator is permitted to occupy while traversing it. Both must be supplied before the problem is fully specified, and both follow from requirements already identified: the second restriction imposed by the liquid payload, noted in Section 4.1.2 and deferred to here, and obstacle set $\mathcal{O}$ introduced in the definition of C_{free} at equation (4.9).

4.5.1 The Second Restriction Imposed by the Payload

The orientation requirement of Section 4.2 constrains the shape of a motion. It does not constrain its speed, and a motion that satisfies it perfectly may still spill the tube’s contents.

The reason is that the free surface of a liquid orients itself normal to the total specific force acting on it, which is the vector sum of gravity and the negative of the acceleration imposed on the vessel. A tube held exactly upright but accelerated horizontally at magnitude α presents to its contents an effective vertical tilted from the true one by:

$$\beta_{\text{eff}} = \arctan\left(\frac{\alpha}{g}\right) \quad (4.15)$$

where g is the acceleration due to gravity. The liquid behaves as though the tube were tilted through β_{eff} even though $\beta(q) = 0$ throughout, and it spills accordingly. A horizontal acceleration of 0.3 g is thus equivalent, from the liquid’s point of view to a static tilt of roughly 16.7° - comparable to the entire nominal tolerance of Section 4.2.3.

Two consequences follow for the formulation. The requirement $\beta(q) = 0$ is necessary but not sufficient, and a bound on acceleration must accompany it. And the two requirements are not interchangeable: a planner cannot compensate for an aggressive velocity profile by holding the tube more precisely upright, because (4.15) is unaffected by β . They constrain different aspects of the motion and must be imposed separately – the first on the path, by the constraint manifold, and the second on the timing subsequently applied to it.

4.5.2 Time Parameterisation and the Scaling Factors

Converting a path into an executable trajectory requires assigning to each configuration a time, and thereby a velocity and an acceleration. The manipulator declares a maximum velocity v_i^{max} and a maximum acceleration a_i^{max} for each joint, and any admissible trajectory must respect both at every instant.

To satisfy the requirement of Section 4.5.1, the system does not operate at these declared limits. Two scaling factors $\lambda_v, \lambda_a \in (0,1]$ are applied, so that the trajectory is generated against the reduced bounds:

$$v_i^{\text{cmd}} = \lambda_v \cdot v_i^{\text{max}}, \quad a_i^{\text{cmd}} = \lambda_a \cdot a_i^{\text{max}} \quad (4.16)$$

The factors are selected by what the gripper holds rather than by which kind of motion is being performed, since it is the payload and the geometry of the path that determines what accelerations are tolerable. Motions with an empty gripper are scaled at $\lambda_v = \lambda_a = 0.30$ and every motion made while carrying liquid at $\lambda_v = \lambda_a = 0.15$.

The timing itself is produced by time-optimal trajectory generation: given the path and the bounds (4.16), the algorithm fits the fastest timing consistent with them, respecting the acceleration bounds along the whole trajectory rather than merely interpolating between waypoints. One property of this choice should be stated plainly, since it is easily overstated. The parameterisation bounds velocity and acceleration; it does not bound jerk, the derivative of acceleration, which is left unconstrained. The motion is therefore smooth in the sense that it commands no step change in velocity, but it is not jerk-limited, and any residual excitation of the payload by rapid changes of acceleration is controlled only indirectly, through the magnitude of λ_a .

4.5.3 The Binding Factor

The pair (λ_v, λ_a) appears symmetric in (4.16), and it is natural to treat the two as jointly determining how fast a motion runs. Over the distances involved here they do not, and the asymmetry is worth deriving because it determines which parameter is worth tuning.

Consider a joint required to move through a displacement d under bounds $v = \lambda_v v^{\max}$ and $a = \lambda_a a^{\max}$. If the joint accelerates at a for half the motion and decelerates for the remainder, its peak velocity is $\sqrt{(a d)}$ and the total duration is:

$$t = 2\sqrt{(d/a)} \quad (4.17)$$

This triangular profile is admissible provided the peak does not exceed the velocity bound, that is, provided:

$$d \leq \frac{v^2}{a} = (\lambda_v^2 / \lambda_a) \cdot ((v^{\max})^2 / a^{\max}) \quad (4.18)$$

Below this displacement the velocity bound is never reached and (4.17) applies: the duration varies with the inverse square root of the acceleration scale and is independent of λ_v altogether. Above it the joint cruises at its velocity bound for part of the motion and λ_v becomes significant.

Equation (4.18) has a feature that is easy to miss. Its right-hand side is proportional to $\frac{\lambda_v^2}{\lambda_a}$, so scaling both factors down together by a common λ lowers the threshold in proportion to λ . Slowing the system down therefore makes its motions more likely to fall in the acceleration-limited regime, not less. For this manipulator the declared limits are equal at 3.0 in both quantities, so at the carrying scale $\lambda=0.15$ the threshold lies at 0.45 rad – a displacement larger than most individual joints undergo during a station-to-station transit.

The practical reading is that λ_a governs the duration of a motion and λ_v has almost no effect on it, so the acceleration factor is the one that must be chosen with reference to (4.15) and the velocity factor is largely nominal. That this holds for the motions the system actually performs is not established by the argument above, which fixes only the criterion; it is confirmed by measurement in Section 7.9.2.

4.5.4 The Collision Model

The set C_{free} of equation (4.9) was defined against an obstacle set $\mathcal{O}$ and an occupied region $\mathcal{A}(q)$, neither of which was specified. Both are models, and the requirements they must meet are part of the problem statement:

1. **The Obstacles:** $\mathcal{O}$ comprises the bench surface, the fixtures adjacent to the working area, the tube rack and its occupants, and the mixing vessel. Each is represented by a bounding primitive – a box or a cylinder – rather than by a detailed mesh, on the grounds that a primitive enclosing the true object yields a conservative C_{free} , and that the error introduced runs in the safe direction. The bench is the most consequential member of $\mathcal{O}$, since it extends beyond the manipulator's reach in both horizontal directions and therefore bounds the workspace from below at every configuration.

2. **The Manipulator and its Payload:** $\mathcal{A}(q)$ comprises the links of the arm, the gripper, and – while a tube is held – the tube itself, in the pose in which it is being carried. The last of these is a requirement rather than a detail: the carried tube is the part of the assembly furthest from the wrist and the part most likely to strike something, so a model omitting it would define as C_{free} that is not conservative but simply wrong. It follows that C_{free} is not a fixed set. It changes at the instant of a grasp, and a configuration admissible with an empty gripper need not remain so once a tube is held.
3. **Fidelity:** C_{free} is defined against $\mathcal{O}$ and not against the physical workspace, and every guarantee derived from it is conditional on the two agreeing. This dependence is of a different character from the others in this chapter, because it cannot be discharged by any test the planner is able to perform: a planner will report, correctly and confidently, that it has avoided an obstacle that is not where the model places it. The verification must come from outside the planning system, and the arrangements by which it is obtained are described in Section 7.8.

4.6 Payload Safety Criteria

The liquid imposes two requirements on the motion, and they are treated separately because they act through different mechanisms and are enforced at different points of the pipeline. The first concerns the orientation of the tube and is enforced on the path. The second concerns the accelerations applied to it and is enforced by the timing. Section 4.2.3 introduced a tolerance ε for the first. This section derives the hard limit within which that tolerance must lie, and then examines the second, including the dynamic response of the liquid.

4.6.1 The Hard Tilt Limit

The tolerance ε of Section 4.2.3 is a specification given to the planner, not a safety limit. What matters physically is the tilt at which the liquid reaches the rim, and this follows from the geometry of the tube and the fill level.

Consider a tube of internal radius r and internal depth H , with a hemispherical base, holding a volume V . Upright, the liquid stands to a depth

$$h = r + (V - \frac{2}{3}\pi r^3)/(\pi r^2) \quad (4.19)$$

When the tube is tilted by an angle θ , the free surface remains horizontal and therefore meets the wall obliquely. As long as it meets the wall around its whole circumference, the volume is conserved by rotating the surface about the point at which it crosses the axis, which remains at depth h , so the surface rises by $r \tan \theta$ on the lower side. Liquid reaches the rim when $h + r \tan \theta = H$, which gives the static spill angle

$$\theta_{\text{spill}} = \arctan((H - h)/r) \quad (4.20)$$

This holds provided the surface does not also uncover the base, that is, provided $h - r \tan(\theta_{\text{spill}}) \geq r$.

The tubes used here have an external diameter of 16 mm and a wall of approximately 1 mm, so that $r \approx 7$ mm, with $H \approx 100$ mm and a fill of $V = 9$ ml. Equation (4.19) then gives $h \approx 61$ mm, so that the liquid stands some 39 mm below the rim, and (4.20) gives $\theta_{\text{spill}} \approx 80^\circ$. The condition of validity is met with room to spare: at that angle the surface on the upper side still lies about 22 mm above the base. The result is not sensitive to the estimate of the wall thickness, since an internal radius of 6.5 mm or 7.5 mm changes θ_{spill} by about one degree.

The hard limit is set well inside this value:

$$\beta_{\text{max}} = 0.70 \text{ rad } (40.1^\circ) \quad (4.21)$$

Every trajectory executed while a tube is held must satisfy $\beta(q) \leq \beta_{\text{max}}$ at every waypoint, with β the tilt function of equation (4.5). The check is independent of the planner that produced the trajectory

and of the tolerance that planner was given, and a trajectory that violates it is rejected before execution. The limit leaves a margin of roughly 40° against static spillage. That margin is reserved for effects that β does not capture: the accelerations of Section 4.6.2, and the difference between the modelled and the physical tilt (Section 4.3.4).

The tilt values used in the system are therefore distinct in kind:

- $\varepsilon = 0.30$ rad, and its relaxation to 0.50 rad, are tolerances requested of the constrained planner.
- β_{max} is the limit against which every carrying trajectory is checked.
- θ_{spill} is the physical threshold that β_{max} protects.

The chart of Chapter 7 satisfies all three trivially, its planned tilt being zero by construction. The criterion matters for the baseline and its fallbacks, where it is the only guarantee.

4.6.2 Accelerations and Sloshing

The tilt limit bounds the orientation of the tube but says nothing about the forces acting on its contents. These act through two mechanisms, which differ in timescale.

Quasi-static response

As set out in Section 4.5.1, a horizontal acceleration α tilts the effective vertical by $\arctan(\alpha/g)$, and the liquid responds to the sum of this and the tube's own tilt. If the liquid follows the effective vertical without lag, the condition for retaining the contents is

$$\beta + \arctan(\alpha/g) \leq \theta_{spill} \quad (4.22)$$

Dynamic response

The liquid does not follow the effective vertical instantaneously. Its free surface is an oscillator, whose fundamental antisymmetric mode in an upright circular cylinder has the frequency

$$\omega_1^2 = (g \xi_{11} / r) \cdot \tanh(\xi_{11} h / r), \quad \xi_{11} \approx 1.841 \quad (4.23)$$

[63]. For $r \approx 7$ mm and $h \approx 61$ mm the hyperbolic tangent is indistinguishable from one, and $f_1 = \omega_1/2\pi \approx 8$ Hz. At a diameter this small, surface tension is not negligible relative to gravity. It raises the frequency further and adds damping, so the value given by (4.23) is a lower bound.

Two consequences follow. First, the frequency lies well above the rate at which the accelerations of a transfer vary, since a transfer occupies between 55 and 95 seconds and its individual motions several seconds each. The motion therefore does not drive the liquid near resonance, and the quasi-static description of (4.22) is the appropriate one. Second, the time parameterisation of Section 4.5.2 does not bound jerk, so a step change in acceleration excites the mode transiently. For a lightly damped second-order system the response to a step overshoots its final value by up to a factor of two. The conservative form of (4.22) therefore doubles the dynamic term:

$$\beta + 2 \arctan(\alpha_{max} / g) \leq \theta_{spill} \quad (4.24)$$

With β at its hard limit of 40° and $\theta_{spill} \approx 80^\circ$, condition (4.24) holds for any horizontal acceleration up to $g \cdot \tan 20^\circ \approx 3.6$ m/s². For a tube carried exactly upright, as on the chart, the same condition allows up to $g \cdot \tan 40^\circ \approx 8.2$ m/s².

What is not established

Neither the tilt of the liquid nor its motion was measured, and the system carries no sensor that could measure them. The only evidence is observational: across sixteen transfers, including one carried at a planned tilt of 33.0°, no liquid was spilled. The analysis above makes that outcome expected rather than fortunate. It rests, however, on a model of the liquid, and a fuller tube, a more viscous payload or a smaller margin would call for measurement.

4.7 The Computational Pipeline

The formulation of the preceding sections is realised on a software stack in which each component is responsible for a distinct part of the problem statement. This section identifies those responsibilities. It describes what each component supplies rather than what it achieves, since what it achieves on this problem is the subject of the measurements in Chapter 7 and is in several cases less than the component's general description would suggest.

4.7.1 Components and their Responsibilities

Table 4.2: Software components of the planning pipeline and the part of the formulation each supplies

Component	Supplies	Part of the Formulation
ROS 2	Process structure, message transport, the transform tree	The interface between the planning system and the hardware and perception layers
MoveIt 2	Planning-scene representation, planning-request handling, time parameterisation, execution interface	The obstacle set $\mathcal{O}$; the scaling of equation (4.16)
OMPL	Sampling-based search, including constrained state space	The search over Q , and the enforcement of the constraint by the mechanisms of Sections 4.3.2 and 4.3.3
TRAC-IK	Numerical Inverse Kinematics for a redundant chain	The map from a pose in $SE(3)$ to a configuration in Q , where no closed-form map is applied
FCL	Collision queries between geometric bodies	The test $\mathcal{A}(q) \cap \mathcal{O} = \emptyset$ defining C_{free} at equation (4.9)

- **MoveIt 2:** It occupies the centre of the pipeline. It maintains the planning scene against which C_{free} is evaluated, mediates between the command layer of Chapter 8 and the planners beneath it, and applies the time parameterisation of Section 4.5.2 to the geometric paths those planners return. It is an orchestrator rather than a solver: the mathematical work is done by the components it dispatches to [64].
- **OMPL:** It provides the sampling-based search. The planner used throughout is RRTConnect, which grows trees from the start and goal configurations and attempts to join them. OMPL also provides the two mechanisms by which a path constraint may be imposed on such a search – rejection of samples violating the constraint, and projection of samples onto the constraint manifold by iterative correction – which are the remedies analysed in Sections 4.3.2 and 4.3.3 respectively. Which of the two is in use is a configuration matter, and one whose effect is not visible in the planner's output, so the question of which is actually active is settled by instrumentation in Section 7.2.2 rather than assumed [12].
- **TRAC-IK:** It supplies inverse kinematics for the redundant chain. It runs two independent numerical procedures concurrently – an iterative Jacobian-based scheme with null-space projection, and a formulation of the same problem as a bounded non-linear optimisation – and returns the first configuration either produces within the joint limits and the error tolerance. It is a general-purpose solver, requiring no structural assumption about the manipulator beyond a description of its links and joints, and it is used wherever a pose must be inverted without one: for grasp and standoff configurations, and at the waypoints of the Cartesian segments [60].

It is worth stating explicitly what this does and does not imply about closed-form solutions. A general numerical solver is required for a general manipulator, because no closed-form inverse exists for an arbitrary seven-joint chain. It does not follow that none exists for this one. Chapter 3 established that the arm satisfies Pieper's condition and admits a closed-form inverse, and Section 4.4.3 identified the availability of such a solution as precisely the condition under which the dimension of M could be exploited rather than merely observed. The two solvers therefore coexist in the system, and the division

of labour between them follows from the structural results of Chapter 3 rather than from any property of the software.

- **FCL:** It performs the collision queries. Given the poses of the manipulator's links, computed from a configuration by forward kinematics, together with the primitives representing $\mathcal{O}$, it reports whether any pair intersects. It is this test, evaluated at sampled configurations and along the interpolations between them, that implements C_{free} [14].

4.7.2 Decisions Left Open by the Toolchain

The components above supply mechanisms. They do not supply the choice among them, and it is worth recording which decisions remain open after the toolchain has been fixed, because those decisions are the substance of Chapter 7.

The toolchain offers no means of searching M directly. Both mechanisms OMPL provides for handling a path constraint operate on the ambient space Q – one by discarding what falls outside a band about M , the other by correcting samples onto it – and both incur the costs set out in Section 4.3. Neither constitutes a chart in the sense of Section 4.4.3, and none of the components listed above supplies one, nor could: a chart depends on the kinematic structure of the particular manipulator, which is information the general-purpose libraries deliberately do not assume.

Nor does the toolchain determine which kind of motion is planned by which mechanism. The transfer described in Section 4.1.1 comprises motions with quite different requirements – a descent whose geometry is prescribed, a transit whose geometry is not, a gripper actuation with no obstacle to avoid – and the assignment of each to an appropriate planner is an architectural decision that the libraries neither make nor constrain. That assignment is set out in Section 7.1.1.

Summary

This chapter has stated the problem the remainder of the thesis addresses and has done so independently of any method for solving it.

The manipulator's configurations form the bounded set Q of equation (4.1). The requirement imposed by the open tube is that its axis, which is the tool frame's x direction and not its z , remain vertical, expressed by the tilt function β of equation (4.5) and by the constraint manifold M of equation (4.7). The planning problem is to find a continuous path lying in $M \cap C_{free}$ between two given configurations, equation (4.10).

The difficulty of that problem is entirely attributable to the constraint. M has codimension two in Q and is therefore of measure zero, so it cannot be sampled directly. The two general remedies, relaxation to the band M_ϵ and projection onto M , are both approximate, satisfying the constraint only to within a tolerance or a convergence threshold. Both are also costly: the first in discarded samples, the second in an iterative solve for every sample. They differ from a search conducted within M itself not in the accuracy of the result but in the kind of guarantee attaching to it, a distinction set out in Section 4.3.4 and returned to throughout Chapter 7.

The admissible set is five-dimensional, by two independent counts. That fact alone confers no advantage, since a planner requires not the dimension of M but an explicit chart of it, and whether a chart exists is a question about the manipulator rather than about the constraint – one to which Chapter 3 has already supplied the structural answer.

Finally, the orientational requirement is necessary but not sufficient: the payload also bounds the accelerations to which it may be subjected, in the manner of equation (4.15), and the trajectory produced from any path must respect the scaled limits of equation (4.16). A hard limit on tilt is derived from the geometry of the tube, and the effect of accelerations, including the dynamic response of the liquid, is bounded separately (Section 4.6). What remains is to construct planners against this specification and to measure what each costs. That is the subject of Chapter 7.

5. Hardware and Experimental Setup

This chapter focuses on the description of the experimental setup and the hardware components that compose the autonomous robotic system. In this chapter, the hardware topology of the system is presented, including the robotic arm, the machine vision subsystem, the central computational subsystem and the remaining laboratory equipment. The objective is to provide a detailed and clear description of the physical subsystems, their technical specifications and the methodologies used to link them into one integrated system.

5.1. Description of the Experimental Setup

The experimental setup was designed and developed with the objective of simulating an automated workstation for executing laboratory procedures, such as the handling of test tubes and the mixing of liquid solutions. The system's architecture is based on visual feedback: the positions of the objects on the bench are determined from camera images before the operations, and the motion of the robotic arm is planned against the positions so obtained rather than against coordinates fixed in advance. **Nothing is observed while a trajectory is executed.**

Physically, the setup is positioned on a solid laboratory workbench. ArUco fiducial markers distributed across the working surface establish the global coordinate frame in which the robot's operational workspace is mapped.

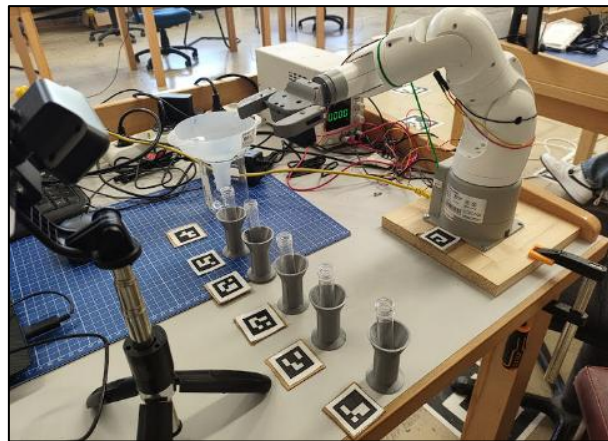


Figure 5.1: Overall view of the experimental setup on the workbench. Displayed are the myArm 300 Pi robotic arm equipped with the custom end-effector, the ArUco markers used for object localisation, and the laboratory mixing equipment.

The main system is architecturally structured around three discrete but interacting subsystems:

- **Robotic Arm Subsystem:** This operates as the mechanical part of the setup. The robotic manipulator is positioned at the right margin of the workspace. Its stabilisation is achieved by a wooden base, which is braced to the workbench. This mounting strategy suppresses operational vibrations and maintains the positional stability of the arm during motion execution. The arm's position was selected so that all labware is inside the workspace.
- **Vision Subsystem:** This subsystem acts as the visual sensor of the setup. A digital camera is mounted on a tripod, above the workspace surface, operating in parallel with a network of fiducial markers (ArUco Markers) distributed throughout the workspace, so that the positions of the objects are recovered from the images.
- **Central Computational Subsystem:** Data management and computational processing are executed by a central computer which acts as the system's main controller. It hosts the ROS 2 environment, processes the camera's video stream via the OpenCV library, generates collision-free trajectories utilising MoveIt, and transmits the low-level motion commands to the arm's joint controllers.

This three-tier architecture separates sensing, computation and actuation into independently reachable components, and it allows the passive laboratory equipment on the bench to be operated automatically.

5.2. Main Robotic Equipment

The central subsystem for motion and physical interaction within the experimental setup consists of the primary robotic arm and the attached end-effector: the myArm 7-DoF robotic manipulator, manufactured by Elephant Robotics [18], equipped with the Thor Gripper, a 3D-printed, open-source hardware design created by Angel L.M [19]. The combined operation of these two mechanical components serves as the physical actuator of the setup, bridging the gap between high-level algorithmic path planning and low-level physical execution.

5.2.1. The Robotic Manipulator

The primary actuator of the experimental setup is the myArm 300 Pi robot. Its compact design differentiates it from traditional industrial robots, integrating technologies that make it suitable for rapid prototyping and delicate laboratory automation tasks.

Arm Specifications (Hardware)

Table 5.1: The myArm hardware specifications

Index	Specification
Degrees of Freedom	7-DOF
Maximum Working Radius	300 mm
Maximum Payload	200 g
Net Weight	1.5 kg
Repeatability Accuracy	± 0.5 mm
Lifespan	500 h
Adapter Specifications	DC 12V-5A
Communication Interface	Ethernet/WLAN

Its mechanical characteristics are detailed as follows:

- **Kinematic Structure:** The manipulator is a symmetrical 7-DoF arm. The addition of the 7th axis provides the system with kinematic redundancy, which allows the robot to reach the exact same point in the workspace using multiple different joint configurations, offering flexibility in obstacle avoidance and in accessing restricted workspaces.

Table 5.2: The range of the joints

Joint	Range
Joint 1	(-160°) - (+160°)
Joint 2	(-80°) - (+80°)
Joint 3	(-165°) - (+165°)
Joint 4	(-100°) - (+80°)
Joint 5	(-165°) - (+165°)
Joint 6	(-110°) - (+110°)
Joint 7	(-165°) - (+165°)

The values above are those specified by the manufacturer's manual [18]. The limits actually enforced during motion planning are those declared in the robot's URDF description, which differ from the

manufacturer's figures in some joints. It is the URDF values that are used in the kinematic analysis in Chapter 3 and in the planning of Chapter 7.

- **Servo Motors:** The motion of the seven joints is executed through integrated, compact, high-precision servo motors. These operate on a closed-loop control logic. Through a built-in encoder, they continuously read the actual position of the rotor and feed this signal back to the controller, allowing it to correct any deviation between the current and the target position, so that the executed trajectory follows the commanded one more closely than open-loop actuation would permit.
- **Material Construction:** The outer shell and the main structural components are plastic, and this keeps the total weight of the manipulator at 1.5 kg, categorising it as a lightweight robotic arm. The reduced mass lowers the torque demanded of the joint servo motors and limits the kinetic energy involved in any accidental contact, which makes the arm better suited to open-workbench operation than a heavier manipulator of the same reach. The polymer structure also damps high-frequency vibration more effectively than a metal one of the same geometry, which shortens the settling time of the gripper after a rapid motion. The same material choice reduces structural stiffness, so the trade-off is between damping and rigidity rather than an unqualified advantage.

Operational Specifications

- **Working Radius:** The arm features a maximum effective working radius of 300 mm [18], measured to the mounting face. The corresponding figure for the wrist centre, which is the quantity relevant to the kinematic analysis, is derived in Section 3.3.4. All labware was placed well within this radius, so that the transfers are executed away from the workspace boundary where the manipulator loses the ability to move outward.

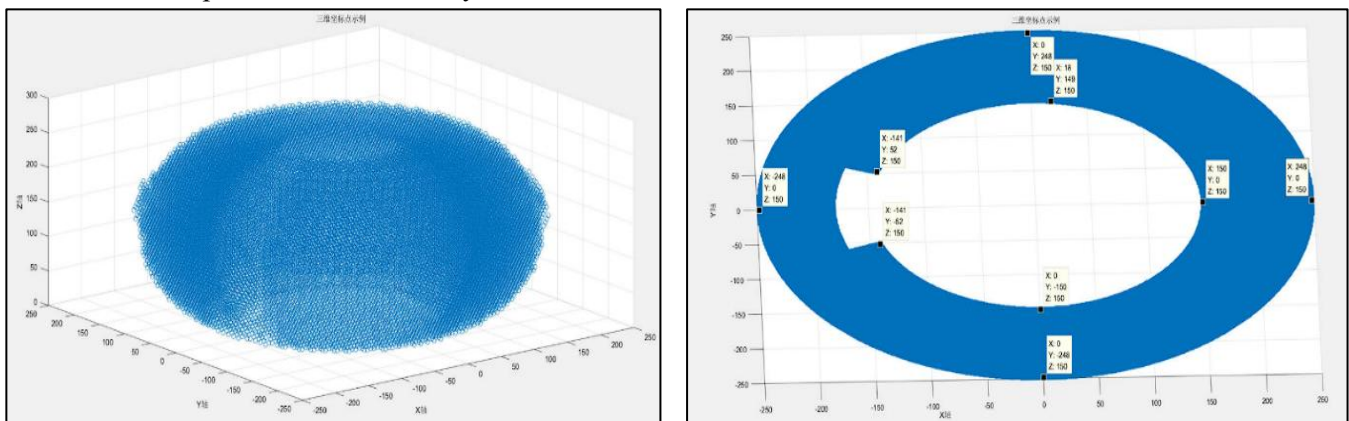


Figure 5.2: Visualisation of the myArm 300 Pi's workspace. The top section presents a 3D point cloud defining the total reachable volume. The bottom section depicts a 2D cross-section of the workspace, highlighting the boundary coordinates that verify the maximum effective working radius of 300 mm [18].

- **Payload Capacity:** The maximum payload capacity of the end-effector is 200 g [18]. This payload is fully sufficient for accommodating the combined mass of the custom end-effector, the labware components and the liquid contents, none of which approaches the limit.
- **Repeatability:** The manufacturer specifies a repeatability of ± 0.5 mm [18]. Repeatability describes the consistency with which the arm returns to a commanded configuration and is distinct from absolute accuracy, which depends additionally on the fidelity of the kinematic model and of the calibration. The tolerances required by the grasping operations are examined in Chapter 9.

Embedded Computing System and Interfaces

The "Pi" suffix in the arm's name indicates the integration of a Raspberry Pi microcomputer in the device. This computer functions as the system's primary hardware gateway and low-level controller.



Figure 5.3: The front interface and the left side of the base, where the Raspberry Pi is housed [18]

The main controller's hardware and software specifications are the following [18]:

Table 5.3: The Raspberry Pi specifications

Index	Specification
Embedded Controller	Raspberry Pi 4B 4G
Core CPU	Broadcom BCM2711, 64-bit 1.5GHz quad-core
Core Memory	4 GB 64-bit LPDDR4 25.6 GB/s
Core Network Port	RJ45
System Version	Ubuntu Mate 20.04
ROS/Python	Built-in ROS1, ROS2, Python

The embedded computer runs Ubuntu MATE 20.04, and the ROS distribution supplied with it is Foxy. The planning and control software developed in this thesis runs instead on ROS 2 Humble, hosted on an external personal computer, and the two systems are therefore not required to share a distribution. The reasons for this division are set out in Section 5.5.3. The value of the embedded computer to the present work lies less in its processing capacity than in two other capabilities: its ability to run the network-bridging software through which that communication is established, and the physical availability of general-purpose input and output pins on the robot's base.

Specifically, the following hardware ports were interfaced with the software environment:

- 1. Network Port (RJ45 Ethernet):** The built-in Ethernet port was utilised for the direct wired connection between the manipulator and the main personal computer. Through this port, the distributed Local Area Network (LAN) communication architecture was established, enabling the transmission of control commands from the PC's ROS 2 environment to the actuators.
- 2. General Purpose Input/Output (GPIO) Ports:** The base of the robot provides direct access to a great number of the Raspberry Pi's pins. For the requirements of the experiment, the control cable of the custom Thor Gripper's servo motor was connected to Pin 12 (GPIO 18). By initiating the "pigpiod" daemon, it became possible to transmit the PWM (Pulse Width Modulation) pulses directly to the gripper's motor over the network, thereby eliminating the need for any additional external microcontrollers.

5.2.2. The Gripper

In this thesis, the end-effector utilised is the Thor Gripper, an open-source 3D-printable robotic gripper designed and developed by Angel L.M. [19]. The primary reason for this selection is its low mass, as the robotic arm's payload is limited to 200 g. Furthermore, its two-finger parallel kinematic design

provides the necessary stability for securely grasping and transferring cylindrical test tubes without applying excessive or uneven lateral forces to the fragile glassware.

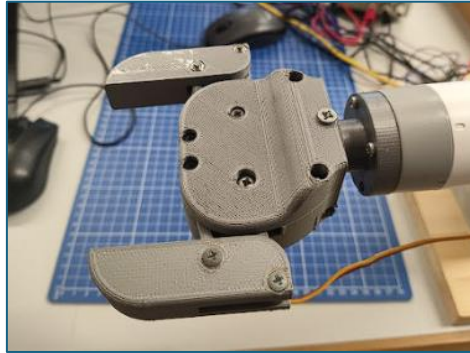


Figure 5.4: Close-up view of the 3D-printed Thor Gripper rigidly mounted to the mounting face of the myArm 300 Pi manipulator.

Actuation and Wiring

The actuation of the gripper – specifically the opening and closing of the fingers – is driven by an MG955 servo motor operating at 5V. This motor is securely bolted to the side of the gripper.

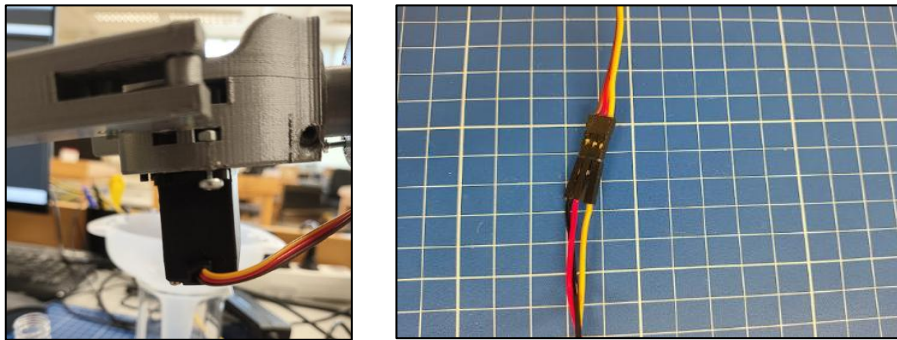


Figure 5.5: The MG955 servo motor securely bolted to the side of the gripper's housing and the 3-pin wiring of the servo motor: the control signal (yellow), the power supply (red) and the ground (black).

The motor interfaces with the control system through a 3-pin connector with the wiring configured as follows:

1. **Yellow Wire (Control Signal):** This pin transmits the Pulse Width Modulation (PWM) signal. It is connected directly to GPIO Pin 18 on the embedded Raspberry Pi. This specific port was selected because it supports hardware PWM, which gives a more stable pulse train than a software-generated signal and therefore more precise servo control.
2. **Red Wire (Power):** This pin connects to the positive pole of the power source, supplying the necessary operating current to the gripper.
3. **Black Wire (Ground):** This pin connects to the negative pole of the power source and the GND port of the GPIO closing the circuit and grounding the system.

Power Management and Electrical Isolation

An important hardware modification was the integration of an external regulated DC power supply. Initially, powering the servo directly via the 5V and GND ports of the Raspberry Pi was attempted. However, the current drawn by the motor during rotation caused an abrupt voltage drop, which overheated the microcomputer and resulted in sudden shutdowns of the device.

To guarantee the safety of the embedded hardware and the uninterrupted operation of the system, the motor's power pins were routed through a breadboard to the external power supply. It was also necessary to establish a common ground between the Raspberry Pi and the external supply within the breadboard. This common ground is an electrical requirement for the integrity of the PWM control signal, since the pulse widths are referenced to the ground potential shared by the two devices.

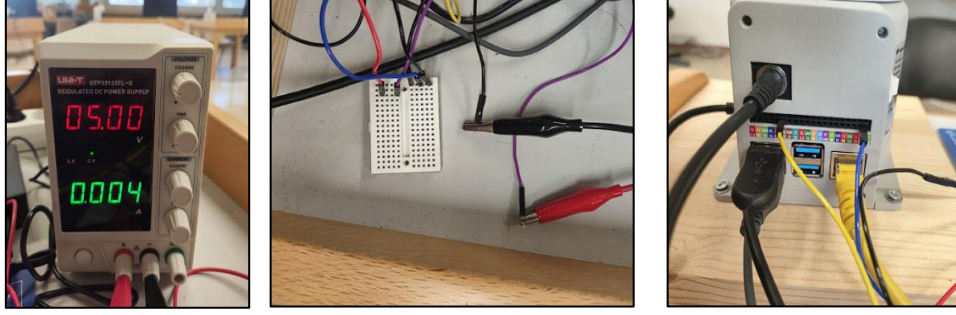


Figure 5.6: The external power management setup.

Mechanical Integration and Adapter Flange Design

Because the Thor Gripper was designed for a different manipulator, its mounting base was incompatible with the 4-M2.5 bolt pattern of the myArm 300 Pi's mounting face. A custom cylindrical adapter flange was therefore designed in FreeCAD and 3D printed to the dimensions of both parts.

The adapter fastens to the arm's mounting face on one side and to the gripper on the other, forming the proximal end of the gripper assembly. That assembly extends 135 mm from the mounting face to the point at which the fingers close, and it is this distance that constitutes the fixed transformation T_7^E between the mounting face $\{7\}$ and the gripper frame $\{E\}$. Since the mounting face itself lies 65.98 mm from the wrist centre, the fingers close 201 mm from it.

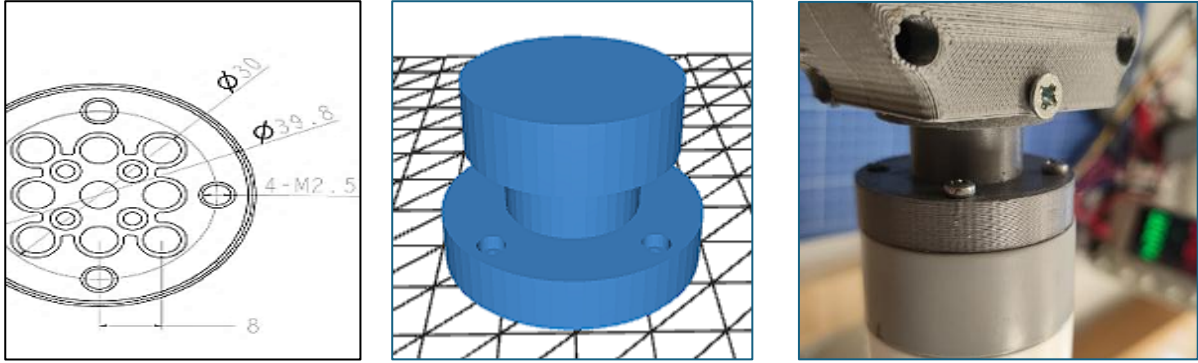


Figure 5.7: The design and implementation of the custom 3D-printed adapter flange.

5.3. Machine Vision Subsystem

The machine vision subsystem constitutes the primary perception sensor of the robotic setup. The selection and positioning of the physical hardware play a decisive role in the quality of the image input streamed to the algorithms, as physical errors (e.g., inadequate lighting, warped markers or camera vibrations) cannot be easily corrected by the software.

5.3.1 Digital Camera and Mounting

For the capture of visual information, the Creative Live! 1080p digital webcam was selected [20]. Its technical specifications include a 1080p image resolution (1920×1080 pixels), a refresh rate of 30 frames per second and a 77° field of view. These specifications were considered adequate for the application's requirements, as they provide the resolution necessary for detecting the edges of the ArUco markers while keeping the computational load at viable levels during real-time image processing. The camera connects to the central computer via a standard USB connection.



Figure 5.8: The Camera and the Field of View to the Markers

The camera was installed on a tripod at an elevated and lateral position relative to the manipulator's base. This specific positioning serves four fundamental purposes:

1. **Complete Coverage of the Workspace:** All seven markers – those identifying the five test tubes and the one identifying the mixer – must lie within a single field of view, since the position of every object is recovered from the same image. The height and inclination of the camera were chosen so that the whole of the working area is visible simultaneously, without the field of view extending so far beyond it that the markers become small in the image.
2. **Minimisation of Distortion:** The elevated placement reduces perspective distortion, thus improving the overall accuracy of the pose estimation algorithm.
3. **Occlusion Avoidance:** The lateral placement ensures that the robotic arm itself does not obstruct the line of sight to the markers with its body during the execution of trajectories.
4. **Equipment Safety:** The camera is located outside the robot's reachable kinematic workspace, so that the manipulator cannot strike it during operation.

5.3.2 Physical Implementation of ArUco Markers

The ArUco Markers were printed at a size of 30×30 mm. A critical element of their setup is their attachment to flat, rigid surfaces (stiff cardboard) placed adjacent to the laboratory glassware. This ensures that the marker's grid remains perfectly flat. Any curvature or bending of the paper would deform the geometry of the square, leading to an inaccurate resolution of the Perspective-n-Point (PnP) problem by the software and, as a result, an inaccurate approach to the object by the arm.

The size of the markers is itself a compromise. A larger marker occupies more pixels in the image and yields a more accurate pose, but the markers must fit inside the workspace without obscuring one another, and the whole set must remain within the field of view described above. The dimension adopted is the largest compatible with the spacing of the rack.

5.4 Peripheral Equipment and Labware

The smooth performance of the chemical synthesis protocols depends on the careful choice of the laboratory equipment and its stabilisation in the workspace of the robotic arm.

5.4.1 Lab Glassware

The experimental design makes use of conventional laboratory glassware, chosen to fit within the payload limitations of the manipulator and the volume constraints of the experiment. The glassware includes:

1. **Test Tubes:** Five cylindrical test tubes, with dimensions of 100×16 mm, used to hold the chemical reagents prior to mixing.
2. **Mixing Vessel:** A 250 ml glass beaker, placed in the reachable area, to be used as a receiving vessel for the liquids to be mixed.

3. **Transfer Funnel:** To allow the precise pouring of the liquids and to avoid spillage or splashing, the funnel is placed over the central beaker.

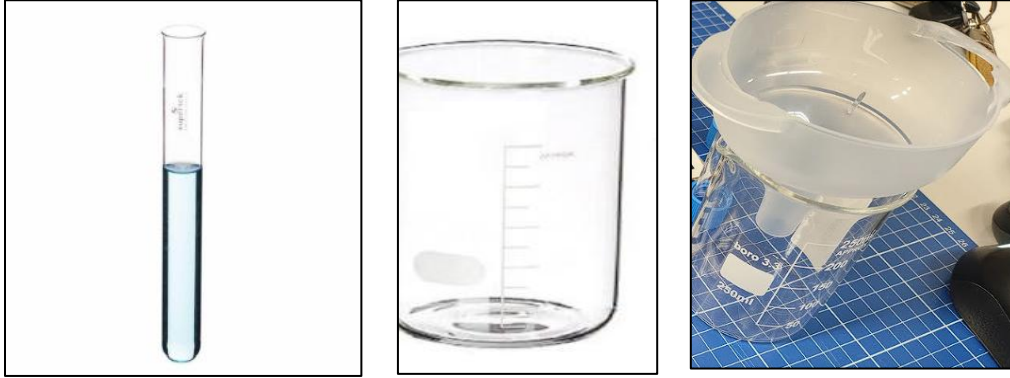


Figure 5.9: Laboratory glassware used in the experimental setup: (A) Test tube used for storing reagents, (B) Central Mixing Vessel, (C) Assembled vessel with integrated transfer funnel for safeguarded liquid transfer during robotic pouring (from left to right).

5.4.2 Custom Support Bases and Stabilisation

Stability of the labware was critical to the safety and repeatability of the robotic pick-and-place operation. Since the custom gripper approaches, grips and returns the tubes vertically, rigid support of the test tubes is essential. 3D-printed support bases were therefore designed for the present application. A critical structural feature is that the top opening of these bases is widened into a funnel shape, forming a chamfered edge. This geometry acts as a mechanical guide: any small deviation of the arm's trajectory from the nominal insertion axis is taken up by the chamfer, which draws the tube towards its resting position during the "place" action and so reduces the likelihood of the tube striking the rim or jamming against it.

The chamfer therefore performs the same function as a mechanical tolerance, absorbing positioning error that the arm's repeatability ± 0.5 mm and the accuracy of the marker-based localisation would otherwise have to meet unaided.

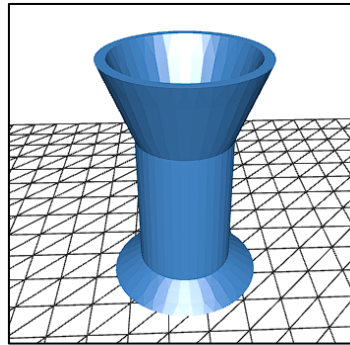


Figure 5.10: 3D design of the custom-fabricated test tube support base. The design features funnel-shaped opening to serve as a mechanical guide for reliable tube re-insertion.

5.5. Communication Architecture and Interfacing

To ensure the smooth and continuous operation of the system – perception, trajectory planning and physical motion – the communication architecture was designed based on the Direct LAN Control model using a bypass logic. In this configuration, the main personal computer acts as the central controller, executing the ROS 2 environment, the computer vision algorithms and the trajectory planning in MoveIt 2. The Raspberry Pi microcomputer, embedded in the base of the manipulator, functions exclusively as a passive bridge, its internal ROS installation being bypassed entirely.

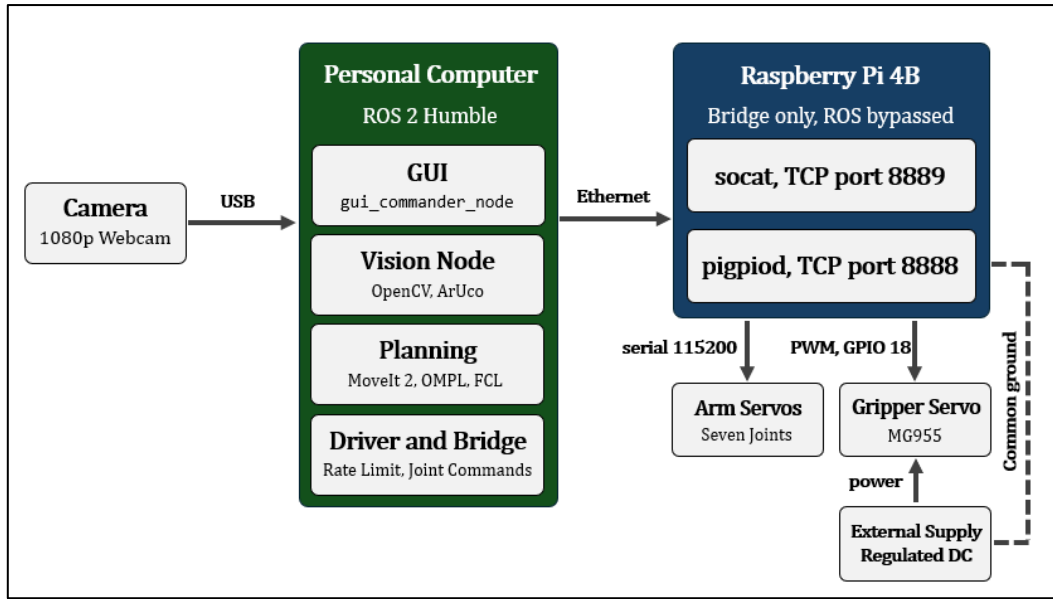


Figure 5.11: Hardware topology of the system. The personal computer performs all image processing and trajectory planning; the Raspberry Pi in the arm's base acts solely as a bridge, its own ROS installation being bypassed. Note that the arm and the gripper are commanded by separate paths: joint angles reach the arm servos over a serial link tunnelled through TCP, while the gripper is driven by a PWM signal generated directly on a GPIO pin.

5.5.1. Physical Connection and Port Forwarding

The physical connection between the robotic arm and the main computer is established via an Ethernet cable, creating an isolated local network. To maximise network stability, static IP addresses were configured: 192.168.123.10 for the PC and 192.168.123.20 for the Raspberry Pi.

To enable the PC to control the motors remotely, the socat utility was used in order to create a virtual serial port. Specifically:

- **On the Raspberry Pi side:** The physical serial port `/dev/ttyAMA0` is forwarded to the TCP network port 8889. The `pigpiod` daemon is then started on port 8888, for direct control of the PWM signal required by the gripper's servomotor. The commands executed on the Raspberry Pi are:

```
$ sudo pigpiod
$ socat TCP-LISTEN:8889,fork,reuseaddr FILE:/dev/ttyAMA0,b115200,raw,echo=0
```

- **On the PC side:** The `socat` utility listens on port 8889 and generates the virtual device `/tmp/virtual_robot`. This allows the hardware driver library to communicate with the robot as if it were directly connected to a local USB port. The command executed on the PC is:

```
$ socat pty,link=/tmp/virtual_robot,raw tcp:192.168.123.20:8889
```

5.5.2. ROS 2 Node Architecture

The translation of planned trajectories into physical motion is performed by two Python nodes running on the PC. These are not the only nodes the system runs — the perception node of Chapter 6 and the user interface of Chapter 8 execute on the same machine — but they are the two through which motion commands reach the hardware. The full source of both is given in Appendix (Chapter 12).

- **MoveIt Bridge:** This node presents itself to MoveIt as the action server from both the arm controller and the gripper controller, receiving `FollowJointTrajectory` goals on each. On receiving an arm trajectory it advances through the requested waypoints, interpolating linearly between them at 50 Hz so that the intermediate states are traversed smoothly rather than jumped between.

The node publishes on two channels for two different consumers. To MoveIt it publishes the current joint state unconditionally at 50 Hz, which is required to keep the action interface from timing out. To the hardware driver it publishes on a separate topic under a rate limit: a message is forwarded only if at

least 50 ms have elapsed since the last transmission, and then only if the joint values have changed by more than a small threshold, with a heartbeat message once per second if nothing has moved. This limits the hardware channel to at most 20 Hz and prevents the serial link, which carries far less bandwidth than MoveIt generates, from being saturated.

It should be noted that the joint state published to MoveIt is the commanded position rather than a value read back from the arm's encoders, since the serial protocol is used here only for transmission. The individual joint servos close their own position loops internally, as described in Section 5.2.1, but no measurement of the executed trajectory is returned to the planning layer. The consequences of this are considered in Chapter 10.

- **Hardware Driver:** The second node subscribes to the rate-limited channel published by the bridge. It reads the target values of the seven arm joints, aligns them with the joint ordering expected by the manufacturer's API, and dispatches them to the manipulator through the virtual serial port using the `pymycobot` library. It also extracts the eighth value, which corresponds to the gripper, converts it into a pulse width within the servo's operating range of 500–2500 μ s, and transmits it directly to the Raspberry Pi through the `pigpio` interface. The mapping is inverted with respect to the raw command value, so that an increasing commanded value closes the fingers.

5.5.3. Justification of the Bypass Architecture

The architecture of the myArm 300 Pi is intended to allow the operating system and control algorithms to run internally on the embedded Raspberry Pi 4. During the initial development of this setup, it became apparent that this arrangement was not viable for the requirements of the present application, which led to the distributed bypass architecture described above. Three considerations motivated the decision.

1. **Computational Load:** The concurrent execution of the ROS 2 core, the trajectory planning of MoveIt 2 and the image processing of OpenCV places a substantial demand on the processor. When these were run together on the Raspberry Pi 4, the system was unable to sustain the video stream and the planning workload at the same time, and processes were observed to throttle and terminate.
2. **Software Compatibility:** The embedded computer is supplied with Ubuntu MATE 20.04 and ROS 2 Foxy, whereas the software developed for this thesis targets ROS 2 Humble. Running the planning stack on the arm would therefore have required either porting it to an earlier distribution or replacing the operating system supplied with the hardware.
3. **Latency:** When motion commands were generated on the Raspberry Pi, delays were observed between the issuing of a command and its execution, which manifested as irregular motion of the manipulator.

In the resulting arrangement the Raspberry Pi is deliberately reduced to a transparent hardware abstraction layer, and the whole of the computational load is placed on the personal computer. The latency of the control path is reduced to the transmission time over the local network, motions are executed without the irregularity observed previously, and the camera stream is maintained at its full frame rate. The performance of the complete system under this architecture is measured in Chapter 9.

5.6. Safety and Constraints

The integration of an automated robotic arm into a synthetic chemistry workflow introduces critical operational and safety requirements of two distinct kinds. Collision avoidance and trajectory planning are handled by the software described in Chapter 7, but the foundational safety of the system rests on its physical architecture.

The mechanical, electrical and workspace constraints imposed by the hardware have been described where the hardware itself was introduced: the wooden base that braces the arm to the workbench and prevents it from overturning under the inertial forces and asymmetric loading that arise in extended postures (Section 5.1), the placement of the camera outside the robot's reachable workspace (Section 5.3.1), the rigid support bases without which the glassware could be displaced or overturned during

insertion (Section 5.4.2), and the separate external DC supply for the gripper servo, sharing a common ground with the Raspberry Pi so that what is separated is the power path rather than the signal reference (Section 5.2.2).

One hardware constraint governs the choice of labware directly. The manipulator's maximum payload of 200 g limits the selection to lightweight glassware and bounds the volume of liquid each tube may carry, so that the servo motors are not loaded beyond their rating. The same lightness of construction that this constraint reflects is also a safety advantage: at a net mass of 1.5 kg, the arm carries little kinetic energy, and the consequences of an unintended contact are correspondingly slight. This is what permits the system to operate on an open bench rather than within the protective enclosure that an industrial manipulator of comparable reach would require.

5.6.1 Constraints Imposed by the Payload

A further class of constraint arises not from the hardware, but from what the hardware carries, and it is enforced in software rather than by physical means.

The tubes are open, and their contents are retained only while the tube remains upright. The requirement that the tube's axis stay vertical throughout every motion is therefore a safety constraint as much as a functional one, and it is expressed formally in Section 4.2 and enforced by the planning method of Chapter 7.

Uprightness alone is not sufficient. As established in Section 4.5.1, a horizontal acceleration displaces the free surface of the liquid exactly as a static tilt would: an acceleration of 0.3 g is equivalent, from the liquid's point of view to an inclination of roughly 16.7°. The velocity and acceleration scaling factors of Section 4.5.2 are set accordingly; the arm being restricted to 15 % of its declared limits whenever a filled tube is held and 30 % otherwise. These values are constraints on the executed motion in the same sense that the payload limit is a constraint on the load, and they are respected by every trajectory the system executes.

5.7 Summary of System Components

The components described individually in the preceding sections are collected in Table 5.4, which gives for each the model or specification adopted, the quantity used, and the section in which it is described. Two characteristics of the assembly are apparent from the table taken as a whole. The first is that no part of it is specialised laboratory automation equipment: the manipulator is an educational arm, the camera a commodity webcam, and the labware conventional glassware of the kind already present in any chemistry laboratory. The second is that the parts which are specific to this application — the gripper, its adapter flange and the tube support bases — were manufactured by 3D printing from open designs or from designs produced for the purpose, at negligible material cost. The setup described in this chapter is therefore reproducible with equipment that is either already available or obtainable without specialised procurement, which is the condition identified in Section 1.4 as necessary if automation of this kind is to be accessible to laboratories operating under budget constraints.

Table 5.4: Summary of the system's hardware components

Component	Model/Specification	Qty	Section
Robotic Manipulator	Elephant Robotics myArm 300 Pi, 7-DoF	1	5.2.1
Embedded Controller	Raspberry Pi 4B, 4GB (integrated in arm base)	1	5.2.1
End-Effector	Thor Gripper, 3-D printed, Open-Source Design	1	5.2.2
Gripper Actuator	MG955 Servo Motor	1	5.2.2
Adapter Flange	Custom, Designed in FreeCAD, 3D-printed	1	5.2.2
External Power Supply	Regulated DC, for Gripper Servo	1	5.2.2
Camera	Creative Live! Cam Sync 1080p V2	1	5.3.1
Camera Mount	Tripod	1	5.3.1
Fiducial Markers	ArUco, 30×30 mm, on rigid backing	7	5.3.2
Test Tubes	Dimensions: 100 × 16 mm	5	5.4.1
Mixing Vessel	Glass Beaker with Transfer Funnel	1	5.4.1
Tube Support Bases	Custom, 3D-printed, chamfered opening	5	5.4.2
Arm Mounting Base	Custom-Fabricated wooden plate, braced to bench	1	5.1
Host Computer	Dell Vostro 3590 (Core i5/8GB RAM/256GB SSD)	1	5.1
Network	Ethernet Cable, Direct PC-to-arm connection	1	5.5.1

6. Computer Vision

This chapter focuses on the deployment and analysis of the machine vision system developed for localising objects within the workspace. Marker-based localisation constitutes a fundamental component of the experimental setup, as it allows the system to perceive the laboratory environment and adapt to changes in the arrangement of the objects it handles. This capability represents the defining difference compared to conventional robotic systems, which operate strictly on the basis of pre-programmed positions.

While the previous chapters analysed the architecture and kinematics of the manipulator, the study in this chapter concentrates on the integration of the Creative Live! Cam Sync 1080p V2 camera and the use of ArUco Markers. The primary objective of this analysis is the accurate pose estimation of the laboratory glassware and objects inside the workspace.

Initially, the camera calibration process is described, based on the pinhole camera model and the algorithm for detecting the markers and extracting their coordinates via the OpenCV library is analysed. The mathematical transformation of the geometrical data from the camera's coordinate frame to the robotic arm's base reference frame is then examined, which is what allows the recovered positions to be used for motion planning.

6.1 Introduction and Description of the Experimental Setup

The visual feedback system was designed to extract geometric information from the workspace before the execution of the experimental procedures. The implementation is based on a digital camera and a set of ArUco markers. These markers are placed adjacent to the experimental labware, enabling the system to determine the position and orientation of each object. The positions are established once, by a scanning procedure performed before operation begins, and are thereafter used for every transfer until the scan is repeated. They are not tracked while a motion is executed.

6.1.1 Hardware and Connectivity

For image capture, the Creative Live! Cam Sync 1080p V2 [20] is employed, connected via a USB port to the central computer responsible for processing the video stream. This camera provides sufficient resolution and frame rate (1080p, 30fps) for the requirements of this study, and its 77° field of view allows the entire working area to be observed from a single viewpoint.

As described in Section 5.3.1, the camera is mounted on a tripod at an elevated and lateral position with respect to the manipulator's base. The elevation limits perspective distortion, while the lateral offset keeps the body of the arm out of the line of sight to the markers during the execution of trajectories. The positions are determined once, by the scanning procedure of Section 6.6, and thereafter held fixed; a displacement of the tripod after that point invalidates every recovered position until the scan is repeated.



Figure 6.1: The Creative Live! Cam Sync 1080p V2 webcam used in the experimental setup [20]

6.1.2 Configuration and Placement of ArUco Markers

The system employs markers generated from the DICT_4X4_50 ArUco dictionary [17], printed at a physical size of 30×30 mm. These markers are affixed at specific locations, adjacent to the experimental labware, serving distinct operational roles:

- **Reference Marker (ID 0).** Positioned at a fixed, measured location on the workbench, 70 mm from the origin of the manipulator's base frame along the negative y axis, at the coordinates (0, −0.07, 0). Its position in the base frame is therefore known in advance, while its pose in the camera frame is recovered by detection. The two together fix the relationship between the two coordinate systems, as described in Section 6.5. Every other marker is expressed as an offset from this one, so the reference marker is used only during the scan and does not appear in the control logic of Chapter 7, which receives the positions of the remaining markers already expressed in the robot's frame.
- **Object Markers (ID 1-5):** Positioned next to the support bases of the test tubes. Each identifier corresponds to a distinct tube, enabling the recognition algorithm to distinguish between tubes containing different chemical substances.
- **Mixer Marker (ID 6):** Positioned next to the mixing vessel into which the liquid solutions are poured and combined.

This encoding scheme allows the control algorithm to translate user commands issued from the graphical user interface into spatial coordinates. If the user selects the transfer of the substance held in the first tube, for example, the system retrieves the pose associated with identifier 1, plans and executes the trajectory required to grasp that tube, pours its contents into the vessel identified by marker 6, and returns the tube to the base from which it was taken.

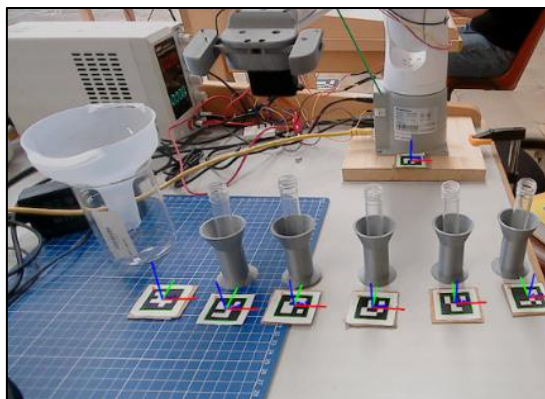


Figure 6.2: ArUco markers from the camera's POV.

6.2 Mathematical Model and Camera Calibration

The precision of a machine vision system depends fundamentally on the accurate geometric modelling of the camera. In this thesis, the procedure includes the mathematical formulation of 3D-to-2D object projection and the experimental extraction of the camera's intrinsic parameters.

6.2.1 The Pinhole Camera Model

The imaging device is modelled as an ideal pinhole camera [65]. According to this theoretical model, the transformation mapping a three-dimensional spatial point $P(X,Y,Z)$ to its 2D projection $p(u,v)$ on the image plane is described in homogeneous coordinates by the following equation:

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \cdot [R | t] \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6.1)$$

Where s is an arbitrary scale factor and K denotes the camera intrinsic matrix, formally defined as:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (6.2)$$

Within this matrix, the parameters f_x and f_y represent the focal lengths expressed in pixel units, while c_x and c_y define the coordinates of the optical centre (principal point) on the image plane. The matrix $[R | t]$ contains the extrinsic parameters, which describe the pose of the camera relative to the scene and therefore differ from image to image. Only K and the distortion coefficients are properties of the camera itself, and it is these that the calibration procedure determines.

6.2.2 Calibration Implementation using a Chessboard Pattern

To empirically calculate these intrinsic parameters, Zhang's camera calibration method [47] was employed, which is the standard approach for this purpose and is natively integrated into the OpenCV library. The implementation was executed via the custom script `calibrate_camera.py`.

The calibration pipeline consisted of the following stages:

1. **Calibration Pattern:** A planar chessboard pattern with a known grid of 9×6 inner corners was utilised as the calibration target.
2. **Image Capture and Processing:** Multiple images of the chessboard were captured from various viewpoints and distances; the pattern being presented at differing orientations and positions within the frame so that the parameters are constrained across the whole of the image area. For each captured frame, the `findChessboardCorners` function was executed to detect the grid's inner corners. The `cornerSubPix` function was then applied to refine the detected corner positions to sub-pixel accuracy, using a termination criterion of 100 iterations or a residual below $\epsilon=0.001$, whichever is reached first.
3. **Parameter Computation:** The final mathematical calibration was performed using the `cv2.calibrateCamera` function, which simultaneously computes the intrinsic camera matrix K and the lens distortion coefficients vector D [66].

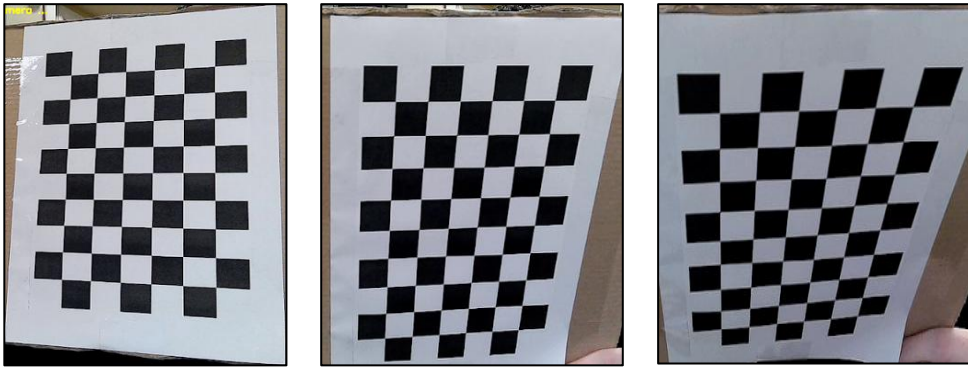


Figure 6.3: Image acquisition of the calibration chessboard pattern from various viewpoints and angles.

6.2.3 Calibration Results

The calibration script was executed for the camera used in this setup, which is enumerated as device 4 by the operating system. The following values were computed and stored in the `camera_params.json` configuration file:

- **Camera Intrinsic Matrix (K):**

$$K = \begin{bmatrix} 1094.33 & 0 & 1009.32 \\ 0 & 1092.53 & 550.24 \\ 0 & 0 & 1 \end{bmatrix}$$

- **Lens Distortion Coefficients (D):**

$$D = [-0.0882, 0.4505, -0.0049, 0.0029, -1.5012]$$

The calibration was performed over 60 images and returned an RMS reprojection error of 0.31 pixels, which is the mean discrepancy between the detected corner positions and the positions predicted by projecting the corresponding model points through the recovered parameters. This figure characterises the quality of the fit. The positional accuracy achieved by the complete localisation pipeline is assessed experimentally in Chapter 9.

The two focal lengths agree to within 0.2 %, as expected for a sensor with square pixels. The principal point is displaced from the geometric centre of the 1920×1080 image by approximately 49 pixels horizontally and 10 pixels vertically, a deviation attributable to the tolerances of the lens assembly relative to the sensor.

The coefficients of D describe the radial and tangential distortion introduced by the lens [66]. The first, second and fifth entries are the radial terms and the third and fourth the tangential ones. Their magnitudes indicate that the distortion is dominated by the radial component, as is typical of a wide-angle lens of this class. This distortion is removed by the software pipeline, each frame being undistorted before the detection of the ArUco markers, so that the geometric relations assumed by the pinhole model of Section 6.2.1 hold to within the accuracy of the calibration.

6.3 Detection of ArUco Markers and Object Identification

The detection of ArUco markers and the identification of objects constitute **the core of the localisation process**, enabling the transition from raw image capture to the extraction of geometric information. The reliability of this process ensures that the robotic arm interacts with the correct laboratory glassware, preventing errors that could lead to experimental failure or physical damage to the equipment.

6.3.1 Detection Algorithm and OpenCV Logic

Marker detection relies on the functionality provided by the OpenCV library, and follows a structured sequence of stages:

- **Adaptive Thresholding:** The raw colour frame is converted to greyscale and then to a binary image, each pixel being classified as black or white according to a threshold computed from the intensities of its local neighbourhood rather than from a single global value [67]. Adapting the threshold locally allows markers to be segmented correctly even when the illumination across the workbench is uneven, which a global threshold would not achieve. The stage nonetheless remains the one most sensitive to image quality: under insufficient or strongly directional lighting, or in the presence of specular reflection from the glassware, the binarisation degrades and detection may fail.
- **Boundary Detection and Analysis:** The contours present in the binary image are extracted by border following [68], and those approximating closed quadrilateral polygons of the kind formed by the thick black border of an ArUco marker are retained. Contours that are too small, non-convex, or of the wrong number of vertices are discarded.
- **Binary Code Extraction:** The region enclosed by each candidate quadrilateral is rectified to a square by perspective transformation and divided into a regular grid of cells. Each cell is classified as black or white and mapped to a binary digit, producing the code carried by the candidate.
- **Dictionary Matching:** The extracted sequence is compared against the entries of the `DICT_4X4_50` dictionary, which contains 50 distinct markers each encoding a 4×4 grid of 16 bits. The codes are chosen to be maximally separated from one another, so that a marker is accepted only if the observed code matches an entry within a small number of bit errors; candidates matching no entry are rejected. Where a match is confirmed, the marker is assigned its corresponding identifier.

Corner refinement is enabled, so that the four corner positions returned by the detector are located to sub-pixel accuracy rather than to the nearest whole pixel. This matters more here than in a typical

detection application, because the procedure of Section 6.4 estimates positions by least squares on those corner coordinates: the accuracy of the recovered geometry is bounded by the accuracy with which the corners are located.

The output of this stage is, for each detected marker in each frame, an identifier and four image points. How those observations are converted into positions is the subject of the next section.

6.3.2 Labware Identification and Spatial Configuration

While the physical placement of the markers is determined by the laboratory setup, the software assigns the detected identifiers to their operational roles:

- **Reference Marker:** Identifier 0 is treated separately from the rest. Its purpose is to relate the manipulator's base coordinate system to the camera's, as described in Section 6.1.2, rather than to identify an object to be manipulated. A frame in which it is not detected is discarded entirely, since such a frame contains no information that can be referred to the robot's coordinate system.
- **Object Markers:** Identifiers 1 to 5 correspond to the five test tubes, one each, and identifier 6 to the mixing vessel into which their contents are poured. This distinction allows the system to determine whether a command issued by the user refers to a valid source and destination before any motion is planned.

Detection is not guaranteed on any given frame, whether because the line of sight is briefly interrupted by the arm, because of reflection from the glassware, or because of noise in the image. The procedure of Section 6.4 accommodates this by aggregating over many frames rather than relying on any one of them, and a marker observed in too few frames to be solved reliably is reported as unsolved and is not published. The consequence is that a marker which cannot be located is absent from the transform tree rather than present at an unreliable position; the planner then places the corresponding object at a nominal position, as described in Section 7.8.1

6.4 Pose Estimation

The transition from detecting a two-dimensional pattern within an image to determining an object's position in three dimensions is the foundation of the localisation process. This section describes how the corner observations of Section 6.3 are converted into positions expressed in the manipulator's own coordinate frame.

The obvious procedure is to solve the Perspective-n-Point problem [69] independently for each marker in each frame, obtaining a full six-degree-of-freedom pose from the four corner correspondences and the known 30 mm side length. That procedure is available and is used here, but only as one ingredient. Applied directly it has two weaknesses that matter at this scale, and the method described below is arranged to avoid both.

6.4.1 Why a Marker-by-Marker Solution is Insufficient

The first weakness is the planar pose ambiguity. A square marker viewed from a distance admits two distinct poses that project to almost the same image, differing by a reflection of the marker's plane. Solvers for planar targets return both, ranked by reprojection error, and the ranking becomes unreliable precisely when the marker is viewed close to head-on — which, for markers lying flat on a bench beneath an elevated camera, is the normal case rather than an exception. A per-marker solution must commit to one of the two on evidence that is weakest exactly where it is most needed.

The second is that the two components of the pose are not equally well determined. The position of a small marker along the camera's viewing direction is estimated far less accurately than its position across the image, since a change in depth alters the apparent size of a 30 mm square only slightly. Most of the error in a per-marker solution lies in that one direction.

Both weaknesses are addressed by exploiting a property of this particular scene: the markers do not have arbitrary independent poses. They are affixed to the same flat working surface and share, to within the accuracy of their mounting, a single orientation.

6.4.2 A Shared Orientation Estimated Jointly

Rather than treating the orientation as seven independent unknowns, the procedure treats it as one three-parameter quantity observed many times over. Every detection of every marker in every frame contributes both of its candidate solutions, and the consensus among them determines the shared rotation.

The estimate is formed iteratively. An initial reference is taken as the medoid of the candidate set, the sample whose total angular distance to all others is least. Each detection then contributes whichever of its two solutions agrees better with that reference; samples deviating from it by more than a fixed threshold are rejected as outliers, and the remaining orientations are averaged. The reference is replaced by that average and the process repeated until it ceases to move.

The advantage is one of evidence. A single detection of a single marker is a poor witness to the orientation, but the estimate is formed from every detection of every marker across the whole scan, and no individual marker viewed from a bad angle can outvote the rest. The ambiguity that a per-marker solution must resolve on weak evidence is resolved here by consensus over the entire observation set.

The dispersion of the accepted samples about the final estimate, and the deviation of each individual marker's own orientation from it, are both recorded. The first indicates how well the orientation is determined, the second tests the assumption on which the whole construction rests, since a marker several degrees away from the consensus is either mounted crookedly or being read from an unfavourable viewpoint, and its position estimate deserves corresponding suspicion.

6.4.3 Positions by Linear Least Squares

Once the rotation is fixed, recovering a marker's position ceases to be a non-linear problem. Writing the projection of a marker corner $\mathbf{p}$, expressed in the marker's frame, as

$$\mathbf{X} = \mathbf{R} \cdot \mathbf{p} + \mathbf{t} \quad (6.3)$$

with normalised image coordinates $(u, v) = (X_0/X_2, X_1/X_2)$, each corner contributes two equations that are linear in the unknown translation $\mathbf{t}$:

$$t_0 - u t_2 = -(\mathbf{R}\mathbf{p})_0 + u (\mathbf{R}\mathbf{p})_2 \quad (6.4)$$

$$t_1 - v t_2 = -(\mathbf{R}\mathbf{p})_1 + v (\mathbf{R}\mathbf{p})_2 \quad (6.5)$$

Four corners therefore give eight equations in three unknowns, solved by least squares. The raw system minimises an algebraic error weighted by depth; one reweighting by the reciprocal of the estimated depth converts this to the reprojection error to first order, and two iterations suffice.

This formulation also disposes of the ambiguity described above. The reflection that makes a planar marker's pose ambiguous is a property of its rotation; with the rotation given, the position that follows is unique.

6.4.4 The Coplanarity Constraint

A further reduction is available where the markers genuinely lie on a common surface with the reference marker. In that case a marker's position may be written as the reference position plus a displacement within the plane,

$$\mathbf{t} = \mathbf{t}_{ref} + a \mathbf{e}_0 + b \mathbf{e}_1 \quad (6.6)$$

where $\mathbf{e}_0$ and $\mathbf{e}_1$ are the first two columns of the shared rotation, that is, the two in-plane directions of the marker frame. Two unknowns replace three, and the direction that is eliminated is precisely the one

identified in Section 6.4.1 as carrying most of the error: the normal to the plane, which for a camera looking down at the bench is close to the viewing direction.

The quantity solved for is also, directly, the quantity the system requires: the in-plane offset of each marker from the reference, in the reference marker's own frame.

The assumption is not taken on trust. For every marker the unconstrained three-parameter solution is computed as well, and its component along the plane normal is reported. Where the markers really are coplanar, that residual is the depth error the constraint has removed; where a marker is genuinely off the plane, the residual reveals it and the constraint can be disabled.

6.4.5 Aggregation across Frames

The steps above yield, for each marker in each frame, an offset from the reference marker. These are combined by taking the median across frames, after one pass of outlier rejection in which offsets lying further than a fixed distance from the initial median are discarded. The median is preferred to the mean because an occasional gross error — a misdetected corner, a partially occluded marker — displaces a mean and does not displace a median.

The frame-to-frame scatter of the retained offsets is recorded for each marker and reported alongside its position. It bounds the random component of the error, and it is worth being explicit about what it does not bound: an error common to every frame, arising from an incorrect marker size, an inaccurate calibration or a mismeasured reference position, is a bias, and no amount of averaging over frames will reveal it. It is, in other words, a measure of repeatability rather than of accuracy, and Chapter 9 assesses the two separately. The scatter is therefore a lower bound on the total error rather than an estimate of it, and the accuracy of the pipeline as a whole is assessed in Chapter 9.

A marker observed in fewer frames than a fixed minimum is not solved. It is reported as such and is not published, for the reason given in Section 6.3.2.

6.5 Expression in the Robot's Coordinate Frame

The procedure of Section 6.4 yields, for each marker, an offset from the reference marker expressed in that marker's own frame. What the manipulator's controller requires is a position in the robot's base frame. This section describes the final step, and the reason it is performed as a difference rather than as a chain.

6.5.1 Why the Camera Pose is not Used

The obvious route is to determine the pose of the camera relative to the robot, and then to carry each marker through it. Since the reference marker is observed in the camera frame and its position in the base frame is known by measurement, the camera pose follows immediately by composing the one with the inverse of the other.

That route is available, and it is not the one taken. The reason is the character of the error it would introduce. Any inaccuracy in the estimated camera pose, arising from the calibration, from the marker size, or from the finite precision with which a 30 mm square can be located, displaces every marker in the scene by very nearly the same amount, because all of them are observed through the same camera in the same frames. Such an error is common-mode, and carrying each marker through the camera pose preserves it in full.

Expressing each marker instead as an offset from the reference marker removes it. Writing $\mathbf{t}_i$ and $\mathbf{t}_0$ for the positions of a marker and of the reference in the camera frame, and $\mathbf{R}$ for the shared rotation of Section 6.4.2, the offset

$$\mathbf{d}_i = \mathbf{R}^T (\mathbf{t}_i - \mathbf{t}_0)$$

is a difference of two quantities affected almost identically, and what remains after the subtraction is the relative geometry of the scene, which is what the system actually needs. This is the purpose the reference marker serves, beyond providing a link between the two coordinate systems.

6.5.2 Mapping to the Robot's Frame

The reference marker is fixed at a measured position in the base frame, at the coordinates $(0, -0.07, 0)$, that is, 70 mm from the base origin along the negative y axis. Its orientation in that frame is likewise known, since it lies flat on the working surface with a known heading, and is expressed as a rotation $\mathbf{R}_0$ taking a vector in the marker's frame to the base frame.

The position of any other marker in the robot's frame follows directly:

$$\mathbf{p}_i = \mathbf{p}_0 + \mathbf{R}_0 \mathbf{d}_i$$

where $\mathbf{p}_0$ is the measured position of the reference marker. Every object on the bench is therefore located by a measured constant and a differential quantity, without the camera's absolute pose entering the result.

The camera pose is nevertheless computed and published, since a visual comparison of where the system believes the camera to be against where it physically is, remains a useful check on the whole procedure. It is used for that purpose and for visualisation, and for nothing else.

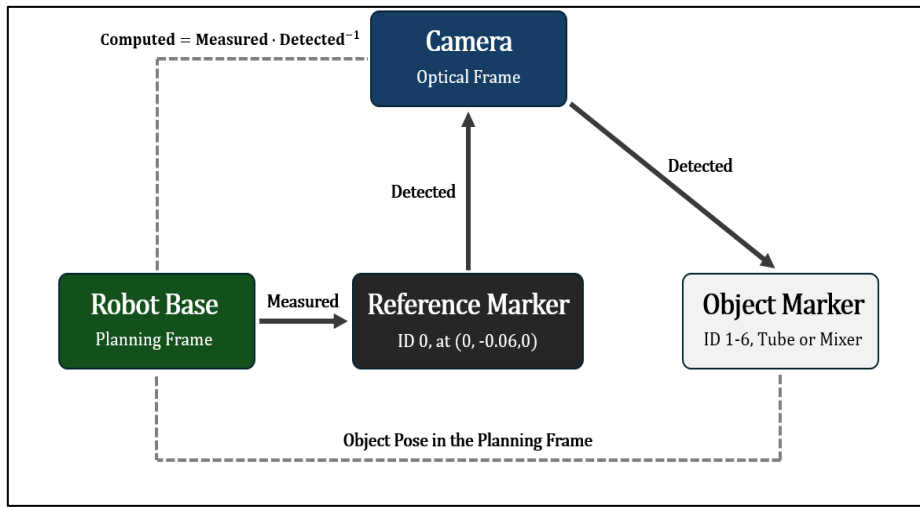


Figure 6.4: The chain of coordinate transformations. The position of the reference marker in the robot's base frame is known by measurement; the positions of the reference marker and of each object marker in the camera frame are recovered by the procedure of Section 6.4. Each object is expressed as an offset from the reference and mapped into the base frame through the reference marker's known pose, so that error common to the camera's observation of both cancels.

6.5.3 Publication and Scope of the Result

The resulting positions are published into the transform tree as static transforms, each marker appearing as a direct child of the robot's base frame rather than of the camera. Nothing downstream is therefore required to route through an estimated camera pose in order to obtain a coordinate, and the control logic of Chapter 7 reads the position of a tube in the frame in which it plans.

Because the transforms are static, the solution holds until the scanning procedure is repeated. A displacement of the camera after the scan is not detected and not corrected: every position remains as it was solved, and the discrepancy is silent. The same applies to a displacement of the reference marker itself, with the difference that an error in the reference propagates identically to every object localised through it.

The accuracy of the result is bounded by two quantities. The first is the precision with which a 30 mm marker's geometry can be recovered from the image, which degrades with distance from the camera and with the obliquity of the viewing angle, and which the aggregation of Section 6.4.5 reduces but does not eliminate. The second is the accuracy with which the position of the reference marker was measured relative to the robot's base, since that measurement is a constant added to every result and no amount of observation will reveal an error in it. The magnitude of the resulting positional error is examined in Chapter 9, and it is the reason the tube support bases of Section 5.4.2 are chamfered.

6.6 Integration with the Control System via ROS 2 and tf2

The procedure described above is implemented in a dedicated ROS 2 node, written in Python, which bridges the OpenCV computer vision library and the ROS 2 ecosystem. The node performs three functions.

- **Scanning:** On starting, and whenever the operator requests a rescan, the node collects frames from the camera until a fixed number containing the reference marker has been accumulated, or until a timeout expires. Frames in which the reference marker is absent are discarded, since they contain nothing that can be referred to the robot's coordinate system, and the node refuses to solve at all if too few usable frames have been gathered. The state of the scan is displayed continuously, including which markers have been seen and how often, so that a marker which is not being detected is apparent before the solution is attempted rather than after.
- **Solution and publication:** When enough frames have been collected the procedures of Sections 6.4 and 6.5 are applied, and the resulting positions are published into the transform system of ROS, tf2 [62], which maintains the relationships between coordinate frames and allows any node to query the pose of one frame with respect to another. Each marker is published as a TransformStamped message naming the robot's base frame as its parent, the shared orientation being converted from a rotation matrix to the quaternion representation tf2 uses internally.

Two properties of this publication are deliberate. The transforms are static: they are published once, following the solve, and are not refreshed thereafter. And every marker is a direct child of the robot's base frame rather than of the camera, so that no consumer of the information is required to route through an estimated camera pose in order to obtain a coordinate. The camera's own frame is published alongside them for visualisation, but nothing depends on it.

- **Reporting:** The node reports, for each marker, the number of frames in which it was observed, the number retained after outlier rejection, its position in the robot's frame, the frame-to-frame scatter of that position, its distance from the reference marker's plane when solved without the coplanarity constraint, and the deviation of its own orientation from the shared one. Markers that were detected but could not be solved are named explicitly as unpublished, and warnings are raised where a marker's orientation departs appreciably from the consensus or where it lies far from the assumed plane. The results are also written to a file, so that a solution can be inspected or compared against a later one without repeating the scan.

The consequence of the static publication is that the solution holds until a rescan is performed. This suits a workbench whose layout changes between procedures rather than during them: the operator rearranges the bench, rescans, and the system plans against the new arrangement. It does not accommodate an object moved while the system is running.

6.7 Summary

This chapter has described the perception subsystem through which the positions of the objects on the workbench are determined.

The camera is modelled as an ideal pinhole device, and the parameters of that model — the intrinsic matrix and the coefficients describing the lens distortion — were obtained by Zhang's calibration procedure [47] from images of a planar chessboard pattern. Every frame is undistorted against those parameters before any detection is attempted.

Objects are identified by ArUco markers from the DICT_4X4_50 dictionary, detected by adaptive thresholding, contour extraction and matching of the recovered code against the dictionary, with the corners of each detection refined to sub-pixel accuracy.

The recovery of position from those corners exploits a property of the scene rather than treating each marker independently. All the markers lie on one surface and share a single orientation, so that orientation is estimated once, jointly, from every detection of every marker across the whole scan, which

resolves by consensus the planar ambiguity that a marker-by-marker solution must resolve on weak evidence. With the orientation fixed, each marker's position follows from a linear least-squares fit to its four corners, reduced under the coplanarity assumption to an in-plane offset of two parameters, which eliminates the direction in which a small marker's position is least well determined.

Each marker is then expressed as an offset from a reference marker whose position in the robot's frame is known by measurement. Since the two are observed through the same camera in the same frames, the error common to both cancels in the difference, and the camera's absolute pose never enters the result. The positions are published into the tf2 transform tree as children of the robot's base frame, from which the planning system of Chapter 7 reads them directly.

The system therefore determines where the objects are rather than assuming it, which is the capability identified in Section 1.2 as absent from conventional laboratory automation. Two qualifications attach to that claim. The accuracy of the result is bounded below by the frame-to-frame scatter of the recovered positions, and bounded in a way no amount of observation will reveal by the hand measurement of the reference marker's position, which is a constant added to every result. And the solution is static: it is established by a scan and holds until the next one, so an object moved during operation is not tracked. Both are examined in Chapter 9.

7. Motion Planning on the Constraint Manifold

This chapter concerns the part of the system that decides how the manipulator moves: the motions from which a liquid transfer is composed, the planners that generate them, and the algorithms that turn a commanded target into an executable trajectory.

One requirement governs everything else. The manipulator carries an open test tube containing liquid, and the tube must remain vertical throughout the motion rather than merely at its endpoints. In motion-planning terms this is a path constraint on the orientation of the end-effector, and it changes the character of the problem entirely: as Section 4.3 establishes, the configurations satisfying it form a surface of measure zero within the space a sampling-based planner explores, so almost every sample drawn is invalid. The constraint is not one difficulty among several; it is the whole of the difficulty.

The approach developed here therefore removes the constraint from the search rather than enforcing it during the search, which the S-R-S structure established in Chapter 3 is what makes possible. Three planning architectures are presented: constrained search in the full seven-dimensional joint space, serving as both baseline and fallback; and two reductions built on that structure: a conservative four-dimensional reduction, in which everything distal of the wrist centre is replaced by a bounding body enclosing every pose it may assume; and an exact five-dimensional reduction that parameterises the freedom the constraint leaves, on which the constraint holds identically by construction. The remaining sections address what is required to make the last of these work in practice.

7.1 Motion Architecture and Anatomy of the Transfers

A liquid transfer is not a single motion and is not planned as one. It is assembled from a small number of motion primitives, each delegated to the planner best suited to it: the constraint that makes transit motions expensive is irrelevant to a gripper actuation, and the geometric precision required of a descent into a tube rack is neither available from nor needed by a sampling-based planner.

7.1.1 Motion primitives

Four kinds of motion appear in the system, distinguished by what must be controlled during the motion rather than by which part of the arm executes them.

- **Free-space transit:** A collision-free motion between two configurations with no requirement on the path taken between them, planned by RRTConnect [13] through the OMPL pipeline [12]. Because the tool orientation is unconstrained, planning is fast. This primitive is used only where no claim about the tool orientation is required, which in practice means an empty gripper, and even then it is a last resort rather than a default.
- **Upright transit:** The same problem with the orientation path constraint imposed: the tool axis must remain vertical for the whole motion. This primitive carries the tube between stations and is the subject of the sections that follow. It is planned in the reduced five-dimensional space where possible and by constrained search in the full joint space otherwise, the latter being a graded sequence of attempts ending by planning without the constraint and admitting the result only if the tube is measured to stay within a hard tilt limit over the whole trajectory.
- **Vertical Cartesian segments:** The descent into the tube rack before a grasp and the lift immediately after it: straight-line motions along the world vertical, 0.10 m for the descent and 0.12 m for the lift, computed by Cartesian interpolation rather than by a sampling-based planner. The geometry is prescribed, since the fingers must descend around the tube along its axis and any deviation risks contact with the rim; and the distance travelled must be deterministic, since it fixes the height at which the tube is grasped. These segments are short and unobstructed, so the flexibility of a sampling-based planner buys nothing that would offset the loss of geometric control.
- **Gripper Actuation and in-place Reorientation:** Opening and closing the gripper are motions of a separate joint group requiring no obstacle avoidance, executed by direct joint interpolation. The

pour is likewise a rotation performed in place about the tool axis, without translating the wrist centre, as is the reconfiguration of the wrist after an approach that arrives in an unfavourable posture, introduced later in this chapter.

The distinction between the first two primitives is therefore one of guarantee rather than of planner. A free-space transit makes no claim about the tool orientation; an upright transit makes one, and that claim is verified on the resulting trajectory regardless of which mechanism produced it — by construction in the reduced space, by measurement in the full one.

Velocity and acceleration are scaled according to what the gripper holds rather than according to the primitive: 30 % of the joint limits with an empty gripper, 15 % while carrying liquid, in accordance with Section 4.5.

7.1.2 Backward planning of the transfer

The segments of a transfer are planned in the reverse of the order in which they are executed, and this inversion is the organising principle of the approach algorithm.

The grasp configuration is the most constrained point of the whole motion and the only one that is not negotiable. The tube stands at a known position with a known axis, the gripper must arrive at a specific pose relative to it, and if no collision-free inverse kinematic solution exists there, no amount of skill in planning the preceding transit will help. The algorithm therefore solves for the grasp configuration first, then for a standoff configuration above the tube, and only then for a transit reaching the standoff.

Failing in this order is also more informative than failing in execution order. A system that plans forwards discovers the infeasibility of a grasp after having planned and possibly executed everything leading up to it; a system that plans backwards reports it before the arm has moved. The diagnosis is correspondingly sharper: by re-solving the inverse kinematics with collision checking disabled, the algorithm distinguishes a target that is out of reach from one that is reachable but obstructed — two failures indistinguishable from the outside that call for entirely different responses.

The dependency runs in the same direction. The standoff is generated from the grasp rather than chosen independently of it, seeded by the grasp solution and accepted only after the descent from it has been verified as executable. Its height is selected from a descending sequence of candidates, since a nominal standoff added to an already high target can place the intermediate pose outside the workspace even where the target is comfortably inside it.

A single transfer, beginning at the home configuration with the gripper empty and open, proceeds as follows.

1. **Transit to the standoff above the target tube:** Free-space or upright transit, depending on configuration, ending at the standoff configuration.
2. **Descend:** Vertical Cartesian segment of 0.10 m, terminating at the grasp pose.
3. **Close the gripper and attach the tube:** Joint interpolation, together with the modification of the collision environment.
4. **Lift:** Vertical Cartesian segment of 0.12 m, clearing the rack before the arm reorients.
5. **Upright transit to the mixer:** The tube is now carried and the orientation constraint is active for the whole of this motion.
6. **Pour:** In-place rotation about the tool axis above the mixer, followed by the reverse rotation.
7. **Upright transit back, descend, release, retreat:** The reverse of steps 1-4, with the constraint still active until the tube has been released.

Steps 2 and 4 are geometrically prescribed; steps 1, 5 and 7 are the ones whose cost the rest of this chapter is concerned with reducing. Two implementations of the sequence exist: the imperative pipeline described throughout this chapter, in which each stage is planned and executed in turn and its outcome verified before the next begins, and a declarative one expressing the same transfer as a MoveIt Task

Constructor pipeline [64], in which the stages are assembled into a single task and solved as a whole before any of it is executed.

7.2 Baseline: Constrained Planning in the Full Joint Space

The conventional way to keep the tube upright is to declare the requirement to the planner and let the planner enforce it. This section describes that approach as implemented in the system. It serves both as the baseline against which the reduction of Section 7.5 is to be understood and as the fallback used when that reduction does not apply.

7.2.1 Expressing the constraint

The requirement is imposed as an orientation path constraint on the link at the tool centre point, referred to the planning frame and expressed relative to the upright reference orientation. Unlike a goal constraint, which restricts only the final state, a path constraint must hold at every state along the trajectory.

The constraint is deliberately anisotropic. Rotation of the tube about its own long axis spills nothing, so that degree of freedom is left free with a tolerance of π , while the two orthogonal directions in which the tube would tip are held to 0.30 rad. The axis convention matters and is easy to invert by accident: the free axis is the tool axis itself, and an earlier version of the code had the opposite convention, permitting a 14° tip while forbidding harmless roll.

Leaving one tolerance at π has a consequence for the parameterisation. MoveIt [11] decomposes an orientation deviation into per-axis components, and the default XYZ Euler decomposition becomes singular precisely when one component is allowed to span a half-turn, which is the case here by construction. The constraint is therefore declared with the rotation-vector parameterisation, which represents the deviation as a single rotation about an arbitrary axis and remains well behaved when one direction is unrestricted.

7.2.2 Enforcement and the three-rung ladder

How a sampling-based planner enforces a path constraint determines what the constraint costs. Two mechanisms are available. Under rejection sampling, the planner samples the joint space as usual and discards the states that violate the constraint; since the constraint-satisfying set is of measure zero within that space, virtually every sample is discarded and planning becomes slow or fails outright at tight tolerances. Under a constrained state space, the planner instead projects each sample numerically onto the constraint surface, which is more expensive per sample but yields states that satisfy the constraint. The system enables the projection-based mechanism for the arm's planning group. The setting is namespaced by planning pipeline and a mismatch there disables it silently, so its effect was confirmed by instrumenting MoveIt's state-space selector rather than assumed from the configuration file.

Even so, constrained planning fails often enough that a single attempt is not a usable policy. The implementation is therefore a ladder of three rungs, tried in order:

1. **Tight Constraint:** The constraint as defined above, at a tolerance of 0.30 rad.
2. **Loose Constraint:** The same constraint relaxed to 0.50 rad, on the reasoning that a plan which tips the tube by up to 28.6° remains well inside the hard limit of Section 4.6.1 and is easier to find than one held to 17.2°.
3. **Verification after freeing the constraint:** The constraint is removed entirely, the query is planned as an ordinary free-space motion, and the resulting trajectory is measured. It is accepted only if the tube never tips beyond the hard limit $\beta_{max} = 0.70$ rad of Section 4.6.1, and rejected otherwise.

The third rung is what makes the ladder work, and it deserves comment because it appears at first to defeat the purpose. Constrained sampling is expensive because the planner discards most of what it generates; planning freely and checking the answer afterwards costs almost nothing, and an unconstrained plan is frequently upright anyway, since the shortest path between two upright configurations rarely has reason to tip the tool. The constraint is not abandoned but relocated: from a

condition imposed during the search to one verified after it. Without this rung the system could not reliably carry a tube back from the mixer.

The verification is applied to every rung, not only the third. A constraint the planner believes it has satisfied is still worth checking.

7.2.3 Measuring the Tilt of a Trajectory

The acceptance test measures the quantity that matters: the angle between the tube's axis and the vertical, at every waypoint of the trajectory. The tool axis is transformed into the world frame at each waypoint, its angle to the reference direction recovered from the inner product, and the worst value over the trajectory reported. This is the tilt function β of equation (4.5).

The measure is chosen over the planner's own per-axis tolerances for two reasons. It is roll-invariant, spin about the tube axis contributing nothing to it, whereas an Euler-angle test goes singular exactly where that freedom is unrestricted. And it is independent of the planner, so the same number can be computed for any method however it enforces the constraint, which is what makes the methods of this chapter comparable at all.

Two properties of the approach should be recorded before the alternatives are described, because both persist regardless of how well the planner performs.

The constraint is satisfied approximately rather than exactly, and by two independent mechanisms. The rotation-vector parameterisation distributes a given deviation across the per-axis tolerances differently from a direct measurement of the tool axis, so a trajectory may lie within specification as MoveIt defines it while tipping the tube by a visibly larger angle. And the planner enforces the constraint at the states it samples, whereas the tilt is measured at the waypoints of the resulting trajectory: a state satisfying the constraint and its successor may be joined by a segment that does not, and such an excursion is invisible to the mechanism intended to prevent it. A trajectory can therefore report zero tilt at every waypoint and still be unacceptable when executed — a limitation that becomes important later.

Satisfaction must accordingly be established by measurement rather than assumed. A tolerance is a specification the planner is asked to respect, not a property the trajectory possesses, which is why the verification is applied to every rung of the ladder and not only to the third.

The remainder of this chapter is therefore concerned not merely with planning faster, but with changing the character of the guarantee: from a constraint enforced during search and verified afterwards, to a constraint the search space cannot violate.

7.3 Implementation and Validation of the Closed-Form Solver

Everything that follows rests on a single software component: an implementation of the closed-form S-R-S inverse kinematics derived in Chapter 3. Both reduction methods use it, the reduced-space planner calls it at every sample it evaluates, and if it were even slightly wrong the trajectories produced would be plausible and incorrect. Sections 3.3.4 and 3.5.3 establish that a closed-form solution exists; they do not establish that the implementation reproduces it, nor that the kinematic model shipped with the robot has the structure the derivation assumes. This section reports the checks that close that gap, and they are the reason the rest of the chapter can treat the mapping from reduced coordinates to joint values as exact rather than approximate.

- **Structure of the model:** The S-R-S property was verified against the kinematic model rather than inferred from the manufacturer's drawing, by the test identified in Section 3.3.4: not whether the link offsets vanish, but whether the mutual distance between the joint axis lines vanishes. Over 2000 randomly sampled wrist configurations the maximum distance from the computed wrist centre to any of the three wrist axes was 3×10^{-17} m, which is machine epsilon; the shoulder triple gives the same result. The axes are concurrent by design, not approximately so. This also confirms that the trap described in Section 3.3.4 is real: the last joint carries an offset of 66 mm expressed as a translation in its parent frame, which inspected naively appears to separate the axis from the wrist

centre, but the offset lies along that joint's own axis. A check written as "are the offsets zero?" would have rejected the S-R-S classification and, with it, the entire method.

- **Agreement with the reference kinematics:** The analytic forward kinematics were compared against those computed by MoveIt from the same model.

Table 7.1: Validation of the closed-form solver against the reference kinematics

Check	Result
Analytic vs Reference Forward Kinematics (Wrist Centre)	Max Error 1.3 μm
Position Inverse Kinematics (5000 random states)	5000/5000 solved, residual 0 m
Wrist ZYZ Inverse Kinematics (3000 random states)	3000/3000 solved, residual 10^{-5}

The round-trip tests take a random valid configuration, compute its forward kinematics, invert them in closed form, and compare the recovered configuration against the original; all 8000 trials recovered the correct solution.

The 1.3 μm disagreement in the first row deserves explanation, since a micrometre-level residual in a closed-form solution invites the suspicion that the derivation contains an approximation. It does not: the residual originates in the robot description, which writes every right angle as the rounded decimal 1.5708 rad, 3.67×10^{-6} rad short of $\pi/2$. Replaying the identical comparison with exact right angles brings the disagreement to 10^{-16} m. Two observations confirm this diagnosis rather than merely being consistent with it: the same rounding propagated over the 66 mm lever arm of the last joint predicts a displacement of 2.424×10^{-7} m, and the measured axis-intersection error with the URDF as written is 2.424×10^{-7} m, agreeing to four significant figures; and the effect scales with the lever arm exactly as a small angular error should. The closed form is exact and the model is rounded, by an amount three orders of magnitude below the manipulator's repeatability.

1. **The self-test as a guard:** These checks are not run once and recorded: the kinematics class executes them in its constructor at every startup and throws if they fail, aborting the node rather than allowing it to run. The tolerances sit well above the rounding artefact and well below any discrepancy a genuine change of frame convention could produce. The solver encodes assumptions about the model — which axes are concurrent, which Euler set each triple forms, which direction each offset points — that hold for the current robot description and would not survive its being regenerated with different conventions. A solver whose assumptions have quietly become false does not fail visibly: it returns configurations wrong by a few millimetres or a few degrees, which the planner then treats as exact, and the error surfaces as an unexplained collision or a missed grasp far from its cause. Failing at startup converts that class of silent error into an immediate one, which is why the exactness claimed in Section 7.5 can be relied upon in operation rather than only in principle.
2. **Locating the redundant degree of freedom:** Chapter 3 established that positioning the wrist centre with four joints leaves one degree of freedom redundant, parameterised by the arm angle. For the implementation a more specific question matters: which joint can be held fixed, and at what cost, if a further reduction in dimension is wanted. The reachable set of wrist-centre positions was sampled over 3×10^6 random configurations and voxelised at 15 mm resolution, once with all four proximal joints free and once with the third locked at zero.

Table 7.2: Reachable wrist-centre volume, with and without the third joint locked

	Reachable Voxels
q_1, q_2, q_3, q_4 free	16312
q_3 locked at zero	15655

Locking the third joint costs 657 voxels, or 4 % of the reachable volume, and the loss lies entirely at the workspace boundary — precisely the region the task does not use, since the stations are well inside the reachable shell. The remaining three joints therefore constitute a genuine positioner for the wrist centre, which makes the three-dimensional variant of Section 7.4 and the four-dimensional variant of Section 7.5 real options rather than theoretical ones.

7.4 Reduction I: A Conservative Swept-Volume Surrogate

The kinematic decomposition of Chapter 3 suggests an immediate way to reduce the dimension of the search. If the wrist centre is determined by the first four joints alone, then a planner searching only those four joints can position it, and the last three contribute nothing to the position being planned.

The difficulty is not in the positioning but in the collision checking. A planner that does not know the wrist joint values cannot know where the gripper and the carried tube are, and those are precisely the parts most likely to strike something. This section describes the standard remedy and the reason it does not work here. The result is negative, and it is reported because the cause is a specific, quantifiable property of the robot rather than a defect in the construction.

7.4.1 The Surrogate Body: A Conservative Swept-Volume Surrogate

The remedy is to replace everything distal of the wrist centre — the wrist links, the mounting face, the gripper and the carried tube — with a single rigid body enclosing every pose those parts can assume, and to check that body instead of the real geometry. The guarantee is one-sided but sound: since the body contains the real end-effector in every wrist configuration, a path collision-free against the body is collision-free for the real arm whatever the wrist is doing. The converse does not hold, so a conservative surrogate can reject feasible paths but never accept an infeasible one, which is the correct direction for the error to run. The whole question is how large that body has to be.

Enclosing all wrist orientations produces a body of revolution about the wrist centre whose radius is the distance from that centre to the furthest point of the carried tube: on this manipulator a sphere of radius approximately 0.24 m, on an arm whose entire reach is 0.2433 m. Such a body intersects the table almost everywhere.

The task constraint is what makes the idea tractable. While the tube is carried its axis must remain vertical, which removes two of the three orientational degrees of freedom and leaves rotation about the vertical itself. Sweeping only that freedom collapses the body from a ball into a flat annulus, the ring traced by the gripper and tube as they rotate about the vertical. Both bodies were generated by sweeping the robot's own collision geometry about the wrist centre, and are therefore properties of the manipulator rather than of any particular workspace.

Table 7.3: Dimensions of the swept surrogate body, with and without the orientation constraint

	Constraint-Aware (tilt 0.20 rad)	Unconstrained, full SO(3)
Max Radius from Wrist Centre	236.6 mm	238.2 mm
Height Span	250 mm (-153.7...+96.4)	407 mm (-176.1...+231.3)
Volume	26.4 L	49.3 L

The constraint buys a factor of 1.9 in volume, and the reduction is almost entirely vertical. The radius is essentially unchanged, and this is not an accident: the width of the ring is set by how far the gripper and tube extend from the wrist centre, and no orientation constraint can shrink that distance. This is the first appearance of the quantity that ultimately decides the outcome. Tightening the tolerance shrinks the body, but with diminishing returns and in the dimension already smallest.

Table 7.4: Dimensions of the swept body as a function of the tilt tolerance

Tilt Tolerance	Max Radius	Volume
0.05 rad	225.5 mm	16.3 L
0.10 rad	226.4 mm	20.0 L
0.15 rad	232.6 mm	23.0 L
0.20 rad	236.6 mm	26.4 L

7.4.2 Attaching and Constructing the Body

One structural detail is easy to get wrong and would invalidate the construction. The swept body is a solid of revolution about the world vertical, not about any axis fixed in the forearm: the constraint that generates it is expressed in the world frame, so as the arm moves the body must retain its orientation rather than rotate with the link it is attached to. Rigidly welding it to the last link of the reduced model would force a return to the useless spherical bound. It is instead attached through two continuous align joints, outside the planning group and never sampled, which the state validity checker sets at every state so that the body's axis remains on the world vertical; two joints suffice, the third rotation being about the body's own symmetry axis. Over 2000 random arm configurations the alignment held to within 10^{-16} rad.

The reduced model is generated programmatically from the full robot description rather than written by hand, so that the first four joints remain bit-identical to those of the real robot; the generated wrist-centre frame matched the full model's to within 0.01 m over 2000 configurations. Two planning groups are emitted, one over joints 1–4 and one over joints 1, 2 and 4, corresponding to the four- and three-dimensional variants.

Because the guarantee depends on the body genuinely containing the end-effector, errors in its construction matter in both directions: too small breaks the guarantee, too large destroys the method's usefulness. Three findings are worth recording, all of which had inflated the body. The six joints of the gripper are mimic joints driven by a single actuator, and sweeping them independently allowed the fingers to reach configurations the mechanism cannot produce. The carried tube was initially swept as though its long axis were the tool frame's z axis, when it is in fact the x axis, adding roughly 35 mm of phantom radius. And the body was emitted as a stack of solid cylinders rather than a resolved mesh, which fills the ring's central hole; preserving that hole matters, since a filled ring rejects every configuration in which a slender obstacle passes through its centre — which, above a rack of tubes, is most of them.

7.4.3 The Planner

The reduced planner drives RRTConnect over a three- or four-dimensional real-vector state space, at the same motion-checking resolution as the full pipeline. One implementation detail is worth noting because it is a natural mistake: the collision request is deliberately left un-scoped to the planning group, since the swept body is not a member of that group and a group-scoped request would skip precisely the object the model exists to represent.

Goal generation exploits the closed-form solver. A single Cartesian target maps not to one joint configuration but to a family — two elbow branches, two shoulder branches, and a continuum of arm angles — and the whole family can be handed to the planner as a goal set rather than a single goal state, so that the search may terminate at whichever member it reaches first. The three-dimensional variant admits a single goal per target instead, since locking q_3 removes the continuum of arm angles from which the others were drawn.

7.4.4 The Result on the Task, and why it Fails

The method does not work on this manipulator. Applied to the transfers the task requires, at the robot's own stations and working height, it admitted almost no candidate configuration and produced no usable plan. Two distinct effects are involved, and only the second is a failure of the method.

The body extends 154 mm below the wrist centre, so it clears the table only when the wrist is above roughly 144 mm. At the height at which a tube is grasped the body therefore always collides with the table — correctly, since it contains the swept tube and the real tube is at table level. This is a scoping result rather than a failure: the reduced planner was never intended to cover the approach and grasp, which are handled by the Cartesian segments of Section 7.1.1.

The failure proper is at transit height. The stations sit at a wrist height of roughly 0.24 m, where the ring is wide enough that positioning it over the tube rack encloses a neighbouring tube, and every candidate configuration is rejected in consequence. The obstacles the arm must avoid are separated by distances much smaller than the ring itself, so there is nowhere above the rack for the surrogate to be.

The cause is not a flaw in the construction but a ratio. The gripper and carried tube extend 201 mm from the wrist centre, on a manipulator whose total reach from shoulder to wrist centre is 243 mm, and the swept ring is 236.6 mm in radius: the surrogate is nearly as wide as the robot itself. Any conservative bound built around such an end-effector occupies most of the workspace the arm can reach, and where the obstacles are much smaller than the bound — a rack of closely spaced tubes — rejection is not a matter of chance but of geometry.

This generalises cleanly, which is what makes the result useful rather than merely disappointing. The conservative bound scales with the reach of the end-effector, while the free space available to move it scales with the reach of the arm. The technique degrades exactly where the payload is long relative to the manipulator and improves where it is short: on an industrial arm with a metre of reach carrying a compact tool the same surrogate would occupy a small fraction of the workspace and be entirely serviceable; on a 300 mm arm carrying a 100 mm test tube it is not.

The next section addresses the cause directly: the surrogate is conservative because the planner does not represent the wrist's remaining freedom, and the remedy is to represent it.

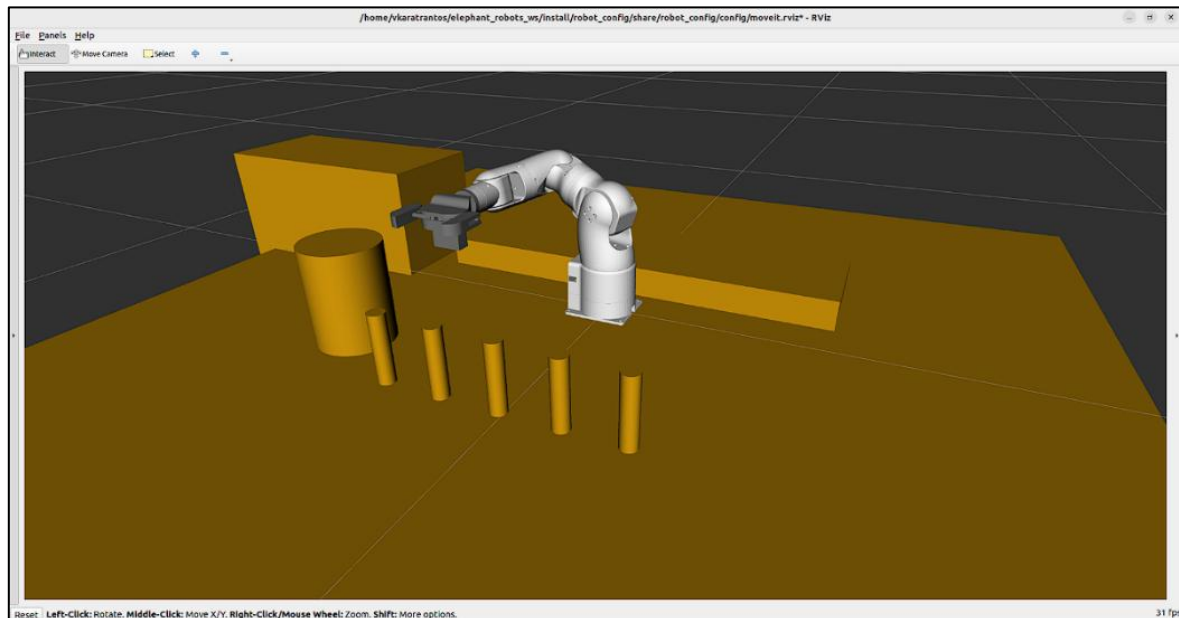


Figure 7.1: RViz preview showing the real seven-joint arm with the carried tube inside the translucent swept ring, animated along a path produced by the reduced planner. The containment of the gripper and the tube within the ring is the assumption on which the whole approach rests

7.5 Reduction II: Exact Planning on the Constraint Manifold

The previous section reduced the dimension of the search but paid for it with conservatism, and the conservatism proved fatal. This section reduces the dimension by the same amount, pays nothing, and is the principal contribution of the work.

The failure of Section 7.4 has a precise cause, and stating it exactly is what produces the remedy. The surrogate exists because the four-joint planner does not know the wrist joint values and must therefore be safe against every value they might take, paying at every configuration it evaluates for the entire range of a quantity it cannot see. But that quantity is not genuinely unknown. The constraint has already fixed two of the three orientational degrees of freedom; only one remains — the rotation of the tool about the vertical — and its value is unknown only to a planner that does not represent it. The remedy follows immediately: rather than bounding that freedom, search it, promoting it from an uncertainty the collision checker must absorb into a coordinate the planner decides. This is not an incremental improvement on the previous approach; it removes the reason the previous approach was conservative.

7.5.1 The chart

Let φ denote the rotation of the tool about the world vertical, the single orientational freedom the upright constraint leaves. The planner searches the five-dimensional space:

$$(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \varphi)$$

and every point of it determines the entire manipulator, in three steps.

Table 7.5: The chart map from the five search coordinates to a complete joint configuration

Coordinate	Determines	By
$q_1 \dots q_4$	The Wrist Centre	Forward Kinematics of the Proximal Chain
φ	The Tool Orientation	The Upright Reference Orientation rotated by φ
-	q_5, q_6, q_7	Closed-Form ZYZ Inversion of that Orientation

The first four coordinates place the wrist centre; φ fixes the orientation the tool must have; and the three wrist joints follow in closed form from the orientation the wrist is required to produce, given what the proximal joints have already contributed. No iteration is involved at any stage, and no configuration is ever rejected for violating the constraint, because none is ever generated that could. Every point of the five-dimensional space therefore maps to a complete seven-joint configuration of the real robot. The map is the chart, and the set it parameterises is the constraint manifold: the surface of configurations at which the tube is exactly vertical.

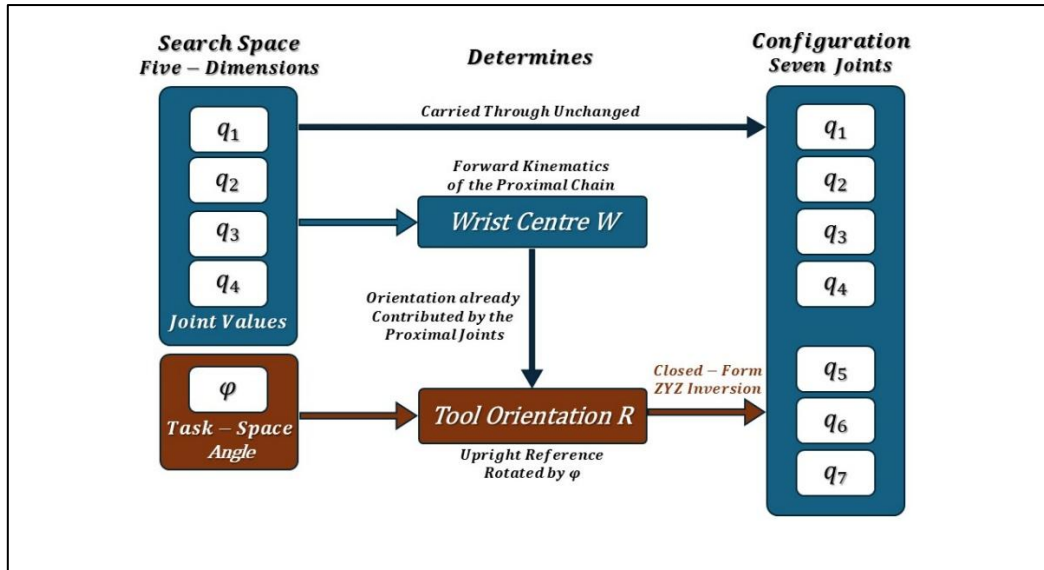


Figure 7.2: The analytic chart. The five search coordinates determine the complete seven-joint configuration in closed form: q_1 – q_4 are carried through unchanged and place the wrist centre, ϕ fixes the tool orientation, and the wrist joints q_5 – q_7 follow from the ZYZ inversion of that orientation given the contribution of the proximal joints. Because every point of the search maps to a configuration at which the tube is exactly vertical, the constraint cannot be violated by construction. The map is defined on a restricted domain and for one wrist branch at a time; these restrictions are set out in Section 7.5.5.

7.5.2 Properties of the Construction

Three consequences follow, and together they are the reason this is the right construction rather than merely one that happens to work.

1. **The constraint is satisfied identically:** Every sample is drawn from a space in which the constraint already holds. There is no rejection on account of the constraint, no projection and no tolerance: the tube is exactly vertical by construction rather than vertical to within some bound. This is a change in the kind of guarantee, not in its magnitude — the distinction set out in Section 4.3.4, where exactness is also defined as holding with respect to the nominal kinematic model. Here it holds, in addition, only within the domain of the chart set out in Section 7.5.5.
2. **The collision check becomes exact:** Because the chart yields the complete configuration, the checker tests the real wrist links, the real gripper and the real tube in their real poses. The conservatism that defeated Section 7.4 does not have to be reduced or bounded more carefully; it disappears, there being no surrogate left to be conservative about. The 201 mm extent of the end-effector is no longer a problem to be worked around — it is simply where the gripper is.
3. **The dimension is forced rather than chosen:** Counting from the constraint, the manipulator has seven degrees of freedom and the constraint fixes two of the three orientational ones, leaving $7 - 2 = 5$. Counting from the geometry, three coordinates locate the wrist centre, one more is the redundant arm angle of Section 3.5.1, and one is ϕ . That two unrelated arguments agree is evidence that the parameterisation is the natural one for this problem rather than an arbitrary choice that happens to work. Locking the arm angle yields a genuine four-dimensional search, at the 4 % workspace cost established in Section 7.3.

Because the claim is easily misread, it is worth stating precisely what is and is not being reduced. The robot has seven degrees of freedom throughout, and all seven joints move. What is five-dimensional is the space the planner searches, not the robot and not the trajectory it produces: four of the five coordinates are joint values, the fifth is a task-space quantity rather than a joint, and the three remaining joints are recovered from it rather than searched. The reduction is therefore not an approximation, a simplification of the mechanism, or a restriction of its capability. It is a change of coordinates — the

planner explores the constraint manifold directly, in coordinates intrinsic to it, instead of exploring the ambient joint space and repeatedly discovering that most of it is inadmissible.

7.5.3 Relation to General Constrained Planning

General-purpose constrained planners construct a chart of the constraint manifold numerically: given an arbitrary constraint function, they project sampled states onto the manifold by iterative root-finding, and that projection is what makes such planners both expensive and approximate[56]. The baseline of Section 7.2 is exactly this.

The construction here is cheaper and exact for a specific reason: the kinematic structure of the arm supplies the chart in closed form. The generality is lower — the technique requires a manipulator satisfying Pieper's condition and a constraint leaving a parameterisable residual freedom — but where those conditions hold, nothing needs to be projected, because nothing is ever off the manifold to begin with [57].

7.5.4 Goal Generation

A significant part of the method's speed comes not from the search but from how the goal is specified, and this too is a consequence of the closed-form solver.

A target is given as a Cartesian pose. Under a numerical solver that pose maps to a single joint configuration, whichever one the solver converges to from its seed. Under the closed-form solver it maps to a family — two elbow branches, two shoulder branches, and a continuum of arm angles, sampled to yield a discrete set — all of which are handed to the planner as a goal set rather than a single goal state, so the search may terminate at whichever member it reaches first. The planner is not asked to reach one configuration satisfying the task; it is asked to reach any of them [60].

Two refinements were necessary, and both are instructive.

1. **The Goal Hint must Rank, not Filter:** To bias the planner toward a configuration near a preferred posture, an initial implementation eliminated every candidate outside a radius of 0.12 rad. On the live task this discarded every candidate, so the planner reported that no goal configuration existed and the caller fell back to the constrained seven-dimensional pipeline — which plans to the pose with no preference at all and lands wherever it likes. A preference strict enough to eliminate the planner that enforces it simply hands the decision to something with no preference. The hint was changed to a ranking: candidates inside the radius are preferred, and if none are, the closest few are retained rather than the query being failed.
2. **Joint-stop rejection is not a preference:** Goal configurations pressed against a joint limit are rejected outright, and this is not subject to the fallback above. A configuration at a joint stop may be reachable, but there is nowhere to go from it, and Section 7.7 shows that arriving at such a configuration is precisely what caused the descent to fail.

7.5.5. Domain, Branches and Singularities on the Chart

The chart of Section 7.5.1 is not defined on the whole of its coordinate space, and a single chart does not cover the whole of the constraint manifold. This section states where the map is defined, which part of M it covers, and which restrictions it inherits from the wrist inversion and from the joint limits.

The Map

For a branch index $\sigma \in \{+1, -1\}$, the chart is the map:

$$\Phi_\sigma: (q_1, q_2, q_3, q_4, \varphi) \mapsto (q_1, q_2, q_3, q_4, q_5, q_6, q_7)$$

in which the first four joints pass through unchanged. The wrist joints are the ZYZ angles of Section 3.4.3 recovered from the rotation the wrist must produce,

$$R_w(q_1, q_2, q_3, q_4, \varphi) = (R_0^4)^T \cdot Rot(\hat{z}_w, \varphi) \cdot R_{up}$$

where R_0^4 is the orientation contributed by the proximal chain, R_{up} is the upright reference orientation, and the constant rotation of equation (3.28) is removed before the inversion. One element of R_w fixes the middle angle q_6 up to sign. The branch index σ selects that sign, and q_5 and q_7 then follow from two-argument arctangents. The coordinates q_1 to q_4 range over their joint limits; ϕ is an angle, defined modulo 2π .

Domain

The map is defined wherever $\sin q_6 \neq 0$, and its domain D_σ is the set of coordinates for which q_1 to q_4 lie within their limits, the recovered q_5 , q_6 and q_7 lie within theirs, and $q_6 \neq 0$. The domain is not a box. The bounds on the first four coordinates are simple intervals. The conditions on the wrist are implicit, depend on all five coordinates, and can be checked only once the map has been evaluated. A sample falling outside D_σ is therefore rejected. This is a rejection on account of the joint limits, not of the orientation constraint. The wrist-limit conditions define a set of full dimension within the five-dimensional domain, so samples fall within it at a positive rate, and every sample that is retained lies on M .

Branches

Φ_+ covers the configurations of M with $q_6 > 0$, and Φ_- those with $q_6 < 0$. Their images are disjoint, and together they cover the whole of M except the set on which $q_6 = 0$:

$$\Phi_+(D_+) \cup \Phi_-(D_-) = M \setminus S, S = \{q \in M: q_6=0\}$$

A plan uses a single chart, the branch being fixed from the start configuration (Section 7.6.1). A motion between configurations on opposite branches must pass through S , cannot be represented within either chart, and is declined and delegated to the constrained planner of Section 7.2 (Section 7.11.1).

Singularity

The chart has one singularity, at $q_6 = 0$, the wrist-lock condition of Section 3.6.2. There the axes of the fifth and seventh joints are collinear, the rotation R_w determines only the sum $q_5 + q_7$, and the map is undefined. Near it the recovered q_5 and q_7 vary rapidly with the chart coordinates, so that a small step in $(q_1, \dots, q_4, \phi)$ can produce a large motion of the wrist: the chart is poorly conditioned close to the boundary between the two branches. The shoulder and elbow singularities of the arm do not affect Φ_σ , because the proximal joints are coordinates of the chart rather than quantities recovered by it; they enter only goal generation (Section 7.5.4), where a target pose is inverted in full. The value $q_6 = \pm\pi$, at which $\sin q_6$ also vanishes, lies outside the range of the sixth joint (Table 5.2).

Joint limits and wrapping

The range of the fifth joint is less than a full turn, so its recovered value has a unique representative within the limits. The range of the seventh joint spans exactly one turn. Its value is recovered in $(-\pi, \pi]$ and unwrapped against the preceding waypoint, and a motion that would carry it across $\pm\pi$ is excluded by the edge validator of Section 7.6.1. Near that seam, the joint values assigned to a point of the chart therefore depend on the path by which it is reached, not on the point alone.

Exactness

Every point of D_σ is mapped to a configuration at which the tube axis, computed through the nominal kinematic model, is vertical to floating-point precision. The statement is relative to that model in the sense of Section 4.3.4, and holds only within D_σ and on the branch σ .

7.6 Continuity of the Analytic Chart

A chart is only useful if the mapping from chart coordinates to joint values is continuous, and here it is not automatically so. Four defects were found, and although their immediate causes differ they share a single character: the orientation varies smoothly while the joint trajectory does not.

This makes them unusually difficult to diagnose, because every check the planner performs returns a correct answer. The constraint is satisfied at every sample, the collision checker finds nothing, and the reported tilt is zero. The defect lies strictly between the states that were checked, or in the correspondence between a chart coordinate and the configuration it is supposed to denote — neither of which any per-state test can see. The first two defects below are generic to planning on an analytic chart of a constraint manifold; the remaining two are specific to this implementation, but of a kind any such chart is liable to reproduce.

7.6.1 Four Discontinuities

1. **The 2π wrist wrap:** The wrist angles are recovered through a two-argument arctangent and therefore lie in the half-open interval $(-\pi, \pi]$. A path crossing the branch cut of that function produces a sequence of orientations that is perfectly continuous — a rotation and the same rotation less a full turn are identical — while the joint value jumps by 2π between one waypoint and the next. Executed literally, such a trajectory spins the wrist through a complete revolution, sweeping the tube through every orientation on the way, including inverted. The symptom was correspondingly confusing: the planner reported a tilt of essentially zero while the arm visibly tumbled. The planner was not wrong; it was correct about every waypoint it generated and had no way to observe the motion between two of them, which is the concrete consequence of the limitation noted in Section 7.2.3. The remedy is to unwrap each recovered wrist angle against the corresponding value at the preceding waypoint, choosing the representative nearest to it rather than the one the arctangent happens to return, subject to the joint remaining within its limits.
2. **The joint-7 seam:** Unwrapping removes the discontinuity but exposes a subtler problem. The limit of the final wrist joint spans exactly one full turn, $\pm\pi$, so a path whose unwrapped values would carry that joint across the seam is asking it to rotate through the single point it cannot reach, and such a path is not executable at all.

The question is where the test belongs. Detecting it after planning, when the trajectory is extracted, merely establishes that a finished plan must be discarded: the planner has already spent its effort, and the caller falls back to a slower method for a query that was solvable by a slightly different route. The correct place is the edge between samples, because traversability across the seam is a property of the connection between two states rather than of either state individually — both endpoints may be perfectly valid while the motion joining them is not. A custom motion validator therefore walks each candidate edge, unwraps the wrist against the preceding state, and rejects the edge if doing so would leave the joint limits. The seam thereby becomes an obstacle within the search, and the sampling planner routes around it exactly as it routes around a physical obstacle, which is what such planners are for. Queries that previously produced an unusable plan and fell back now produce a usable one directly. Validating each edge costs more than validating its endpoints, but the alternative is not a faster plan; it is no plan.

3. **A quarter-turn convention error:** The direction from the wrist centre to the mounting face, as a function of φ , is $(\sin \varphi, -\cos \varphi, 0)$ and not $(\cos \varphi, \sin \varphi, 0)$, because the reference upright orientation maps the tool's own axis onto the negative world y direction. Getting this wrong is entirely silent as long as φ is swept over a full circle, since the set of wrist centres generated is the same either way. It bites only when a specific value of φ must correspond to a specific wrist centre — which is exactly what a planner carrying φ as a state dimension requires. The convention is now expressed in a single helper function so that it cannot drift between the several places depending on it.

4. **An unstable branch index:** The ZYZ inversion returns two branches, and the solver discards those falling outside the joint limits, so the first returned solution could be the positive branch at one state and the negative branch at a neighbouring one. With ϕ carried as a state dimension, this makes the map from chart coordinates to configurations discontinuous in a way no individual state reveals: each state maps to a valid configuration, but adjacent states may map to configurations on opposite branches. The solver was given a stable branch index, and the planner fixes the branch once per plan from the start state.

7.6.2 Bridging onto the manifold

A final continuity problem arises not within a plan but at its beginning, and it is worth recording because it remained invisible for a considerable time.

The manifold planner begins its path at the projection of the current state onto the constraint manifold: the start configuration is packed into $(q_1 \dots q_4, \phi)$ and the wrist rebuilt in closed form, so any component of the real configuration lying off the manifold is discarded. If the arm is already upright the projection changes nothing; if it is not, the first point of the planned trajectory differs from where the arm actually is.

The execution layer refuses any trajectory whose first point deviates from the measured state by more than a small tolerance. The plan was therefore produced successfully, reported as successful, and then rejected at execution, at which point the caller silently fell back to the constrained seven-dimensional pipeline. Because the success was logged by the planner and the rejection occurred elsewhere, the logs showed a working manifold planner throughout. The consequence was significant: the manifold planner was never actually used for empty-gripper transits. While carrying a tube the arm is already on the manifold from the preceding upright motion and the projection is a no-op, which is why the defect surfaced only when the constraint was extended to empty-gripper transits as well.

The remedy is to bridge explicitly from the real configuration to the planned start, checking every intermediate state, and to refuse the bridge outright when any joint would have to move more than 0.25 rad, a threshold that excludes a wrist flip by a wide margin. A discontinuity at the start of a trajectory is no more acceptable than one in the middle of it.

Every one of these defects passed all per-state validation, and the lesson generalises to any planner operating on an analytic chart of a constraint manifold. Satisfying a constraint at every sampled state does not establish that the motion between the states satisfies it, and where the chart involves inverse trigonometric functions, branch structure or a projection, that gap is not a theoretical concern but the ordinary case. The tests that catch such defects are edge tests, not state tests, and the natural place for them is inside the planner rather than after it.

7.7 Integration, Dispatch and Descent Feasibility

The preceding sections established a planner that is fast, exact and correct on the transits it is given. This section concerns what is required to make it usable as part of a working system: how it is combined with the methods it replaces, and what happens when a transit succeeds by every measure available to the planner and the arm is nevertheless unable to continue.

The second occupied considerably more effort than the first. A transit is not an end in itself; it exists to place the arm somewhere the next motion can proceed from. A planner evaluated in isolation cannot see that requirement, and the failure it produces appears at the following stage, with a cause no amount of investigation at that stage will reveal.

7.7.1 Dispatch

Upright transits are dispatched through a single entry point which tries three methods in order.

1. **The Manifold Planner:** Attempted first for every transit
2. **Constrained Search in the Full Joint Space:** Attempted if the manifold planner declines the query – most commonly because the start and goal lie on opposite wrist branches.

3. Unconstrained Planning, Verified: Reached only through the third rung of the ladder of Section 7.2.2, and only when the measured tilt of the resulting trajectory is within the hard limit.

The architecture of this ladder is deliberate and its cost is asymmetric in the right direction. When the manifold planner declines, it does so during goal generation, after which the system falls back to exactly the behaviour it had before the manifold planner existed. The reduction can therefore only improve the system's behaviour, never degrade it — a property worth having in a component whose failure modes were difficult to observe.

Two runtime switches control the policy. One selects whether the upright constraint is applied to empty-gripper transits as well as carrying ones; the default is that it is, so that the tool remains upright throughout the task and not only while a tube is held. The other allows the manifold planner to be disabled entirely, so that the alternative can be exercised against an identical scene without rebuilding.

7.7.2 Diagnosing the descent failures

With the manifold planner in place, upright transits succeeded and reported zero tilt. The descent that followed them frequently did not: the Cartesian segment into the tube rack stalled partway, having completed only a fraction of its commanded distance. The pattern was initially reported as intermittent, stalling anywhere between roughly a tenth and four-fifths of the way down, which suggested a marginal geometric interference — the fingers clipping the rim of a neighbouring tube, or the tube itself catching. Every fix attempted on that hypothesis failed. The conclusion governs the shape of this whole section: nothing was ever in collision.

The reason the hypothesis survived so long is that the Cartesian solver signals a failed waypoint with a single Boolean, and there are three distinct ways of reaching it:

- no inverse kinematic solution exists at that waypoint at all, through reach or singularity;
- solutions exist, but every one of them is in collision;
- solutions exist and are collision-free, but lie further from the previous waypoint than the continuity limit permits — a jump across the null space rather than an obstruction.

These require opposite remedies, and the third is invisible to any amount of collision debugging. Two diagnostics were therefore added. The first re-runs the collision check with contact reporting enabled and returns the touching body pairs together with their penetration depths, since the ordinary check answers only yes or no. The second re-solves the inverse kinematics at the blocked waypoint with collision checking disabled and grades every solution separately against the three cases, so that the reason for the failure is reported rather than inferred. The result was unambiguous: across many runs no blocked waypoint was ever found to be in collision, and almost every one had collision-free solutions available that lay beyond the continuity limit.

Printing the joint values at the stall against their limits then identified the cause immediately. The fifth joint arrived at the standoff at 2.962 rad, against a limit of 2.967 rad — roughly 0.005 rad of travel remaining. The descent requires that joint to continue moving in the same direction; it cannot, so the inverse kinematics jumps to the mirrored wrist branch some 2.80 rad away, and the continuity check correctly rejects the jump. The decisive observation is that the intended standoff configuration itself held the favourable branch, with the fifth joint at 0.070 rad, yet the upright transit went to 2.962 rad regardless — a gap of 2.89 rad from the configuration it was supposed to reach. The transit was not arriving imprecisely at the right configuration; it was arriving precisely at a different one, on the other wrist branch.

This also explained the apparent intermittency. On one attempt the arm arrived within 0.054 rad of the intended standoff and stalled anyway, because by then the standoff configuration had inherited the saturated branch from the start state of a previous failed attempt. Arrival distance was never the variable; the branch was.

7.7.3 Choosing and verifying a standoff

The remedy is to stop treating the standoff as any collision-free configuration at the standoff pose, and to require instead that it be a configuration from which the descent is known to work. Four mechanisms implement this, applied in the order given so that the expensive tests are reached only by candidates surviving the cheap ones.

1. **A reachability gate:** One cheap kinematic probe establishes whether the standoff pose is reachable at all, before any of the expensive machinery runs. This matters because a nominal standoff added to an already distant target can fall outside the workspace even where the target itself is comfortably inside it. Without the gate, an unreachable standoff was proved unreachable six separate times by six different seed strategies, each exhausting its full inverse-kinematics timeout.
2. **A joint-limit headroom screen:** Candidates pressed against a joint stop are rejected, on the grounds established in Section 7.5.4. The threshold is 0.05 rad, and the choice is empirical: every configuration observed to stall a descent had less than 0.01 rad of headroom, and every configuration that worked had at least 0.18 rad, so the threshold sits in the empty gap between the two populations. An earlier value of 0.15 rad was too aggressive, discarding candidates with 0.13 rad of headroom without testing them and forcing the search down to a lower standoff for no reason. Marginal cases are better sent to the authoritative test below than screened out by a proxy for it.
3. **Mirror-wrist seeds:** This is the load-bearing mechanism. Inverse kinematics seeded from a configuration pinned against a stop returns configurations pinned against that stop: all four of the original seed strategies returned solutions within 0.09 rad of the limit, while a perfectly usable configuration existed on the other branch with 0.23 rad to spare. The remedy seeds the solver from the same wrist orientation expressed on the opposite ZYZ branch, using the branch structure of Section 3.4.3 constructively. This is not equivalent to retrying more often: re-seeding on the arm's own branch produces the same pinned solutions however many times it is attempted.
4. **A dry run of the descent:** The authoritative test. Before the arm commits to a standoff, the descent from it is planned — from a configuration the arm has not yet moved to — and the candidate is accepted only if that plan succeeds. This required the Cartesian planner to accept an explicit start state rather than always beginning from the current one, and it settles what the screens above can only estimate.

A duplicate-skip check was also added; four of the six generators having been found to re-test the same rejected configuration repeatedly.

These measures choose a good standoff but do not guarantee the arm reaches it, since the transit may land on a different branch. After each transit the arm's actual configuration is therefore compared against the intended standoff, and if it differs by more than 0.05 rad and the descent from where the arm actually is would fail, the arm reconfigures in place before descending. The reconfiguration is a planned joint-target motion rather than naive interpolation — interpolation was observed sweeping the wrist through the rack — and carries an explicit tilt limit when a tube is held. This mechanism fired at both ends of the originally reported band: what had looked like an intermittent geometric interference was a branch mismatch appearing at whatever point along the descent the pinned joint happened to run out of travel. Cost, and where it is actually spent. The measures are not free, and the distribution of their cost is counter-intuitive. For the most expensive target, almost the whole of the delay before the arm moved at all lay in inverse kinematics for a standoff pose that was not reachable, proved so six times over by six generators in succession; the manifold planning itself accounted for a fraction of one per cent of that delay. The general point is worth carrying beyond this system. Having reduced planning time substantially, the planner ceased to be the bottleneck, and the cost moved somewhere the earlier measurements had no reason to look. Optimising a component that represents a fraction of one per cent of the total is not an improvement; identifying which component that is, is.

One failure mode remains and is fixed by none of the above, because it is a property of the chart rather than of its use: a transit whose start and goal lie on opposite wrist branches cannot be represented within a single chart and correctly falls back to the constrained pipeline of Section 7.2. This is taken up in Section 7.11.

7.8 The Collision Environment

The planner's behaviour depends entirely on its having an accurate description of the world it is planning in. This section describes how that description is built and kept current. Its placement after the planners is deliberate: the scene is not an inert backdrop but an active component, and two of the defects described in earlier sections were caused by the scene being subtly different from what the planner assumed.

Four categories of object are published once at startup, expressed in the planning frame.

Table 7.6: Collision objects published to the planning scene

Object	Primitive	Dimensions
Table	Box	$1.5 \times 1.5 \times 0.04$ m, top surface at $z = -0.01$
Static Obstacle 1	Box	$0.7 \times 0.2 \times 0.05$ m
Static Obstacle 2	Box	$0.4 \times 0.2 \times 0.3$ m
Test Tubes ($\times 5$)	Cylinder	height 0.1 m, radius 0.008 m
Mixer	Cylinder	height 0.17 m, radius 0.04 m

The table is the most consequential of these. Because it extends well beyond the manipulator's reach in both horizontal directions, it bounds the workspace from below at every configuration, and it is what the swept body of Section 7.4 collides with at every wrist height under 0.144 m. The two static obstacles represent laboratory fixtures adjacent to the working area and give the planner something non-trivial to route around.

7.8.1 Dynamic tracking, and behaviour when perception is absent

The tubes and the mixer are not at fixed positions. Each carries a fiducial marker, and their positions are obtained by querying the transform tree for the marker frame directly in the planning frame, a background timer refreshing them every 500 ms. Two details of this arrangement are worth recording, because both were originally wrong.

- **The transform must be requested into the planning frame, not merely stamped with it.** An earlier implementation looked up the transform between the marker base and the individual marker, then labelled the resulting object with the planning frame, which is correct only if the two frames coincide. Asking the transform tree directly for the planning frame is always correct and fails visibly when the frames are not connected — the desired behaviour, since a silently mislocated obstacle is far worse than an evident failure.
- **The scene must not change shape when perception arrives.** The tube height and the mixer radius were originally defined with different values in the startup code and in the refresh timer, so the scene had one geometry at startup and a different one 500 ms later, without any indication that it had changed. All such quantities are now defined once and used everywhere.

When a marker is not visible — at startup, before the camera has published, or if a marker is occluded — the object falls back to a nominal position rather than being omitted. Omission would be the more dangerous choice: a planner that cannot see an obstacle will plan through it, whereas an obstacle in approximately the right place still forces the plan around the region it occupies. The refresh timer is suspended for the duration of a planned motion, since a scene update arriving between planning and execution can invalidate the trajectory that was just verified.

7.8.2 Grasping: attachment, permissions, and where the tube is

When a tube is grasped it must cease to be an obstacle and become part of the robot. The two parts of this are separate and both are necessary.

- **Attachment** transfers the tube from the world to the gripper link, so that it moves with the arm and is checked against the environment in its carried pose. This is what makes the collision check of Section 7.5.2 exact: the planner tests the real tube in the real place it is being held.
- **Modifying the allowed-collision matrix** is the part that is easy to omit. The tube's collision cylinder sits exactly where the gripper closes around it — which is correct and is the whole point of modelling it — so every grasp pose reads as a collision until the fingers are explicitly permitted to touch that one tube, and attachment alone does not create these permissions. Without them the Cartesian descent returns a partial fraction, and the grasp fails at the first waypoint, indistinguishable at first sight from the failure diagnosed in Section 7.7.2.

The permission is granted before the descent begins, and attachment follows before the gripper closes, so that the matrix is already correct while the fingers move into the tube's geometry. On release the sequence is reversed and the tube returned to the world at its current pose.

One geometric quantity governs the entire grasp and deserves to be stated explicitly, because it is the source of a systematic error that is easy to make and hard to see. The tool centre point frame is at the mounting face, whereas the fingers close 135 mm beyond it. The offset between the marker and the tube body, and the offset between the marker and the commanded tool pose, therefore, differ by exactly that distance — not because of any modelling adjustment, but because they refer to two different points on the robot. Measured from the wrist centre rather than the mounting face, the fingers close 201 mm out, the offset d_7 of Section 3.3.3 accounting for the difference.

This is the same distinction identified in Section 3.5.3 in connection with the wrist-centre retraction, and it recurs wherever a pose is specified: a target expressed at the fingertips and the same target expressed at the mounting face differ by a distance larger than the tube's own radius. Getting it wrong does not produce an obvious failure. It produces a grasp that closes 135 mm short of the tube, which looks like a perception error, a calibration error, or a planning error, depending on where one happens to look first.

7.8.3 Verifying the scene

The collision scene is the planner's only knowledge of the world, and an error in it is invisible to every test the planner can perform: the planner will confidently and correctly report that it has avoided an obstacle that is not where it thinks it is. Verification has to come from outside — from visual comparison of the published scene against the physical workspace, and from the contact reporting introduced in Section 7.7.2, which names the bodies actually touching rather than merely reporting that something is. One such discrepancy is known and unresolved. The vertical position of the tube cylinder is currently set to a value tuned against the physical system rather than derived from the table height and the tube dimensions, and the derived value differs from it by 25 mm. A collision cylinder positioned above where the real tube stands would allow the gripper to clip the real tube without the planner registering anything, so this should be settled by measurement rather than left to the tuned value. It is also the reason no quantitative comparison of the planning methods is offered in this chapter, as Section 7.11 sets out.

7.9 Time Parameterisation and Execution

A planner produces a path: an ordered sequence of configurations with no timing attached. What a controller requires is a trajectory, in which every waypoint carries a position, a velocity, an acceleration and a time. This section describes that conversion, and the arrangements needed to make the resulting trajectory be what the arm actually executes.

Paths are converted to trajectories by time-optimal trajectory generation, on the principles set out in Section 4.5.2 and scaled by the factors given there. The same parameterisation is applied to every path the system produces, whether it came from the manifold planner, the constrained pipeline, or Cartesian interpolation. This is deliberate: paths from different planners are otherwise timed by different mechanisms, and motions that are geometrically similar then execute at visibly different speeds. Applying one parameterisation uniformly also matches the request adapter used inside the planning pipeline, so that a trajectory produced by the pipeline and one produced directly by this system are timed identically [70].

7.9.1 Two defects between the plan and the motion

Two faults, in different components, meant that for a period the executed motion bore little relation to the planned one.

1. **The state relay decimated everything to 10 Hz:** The joint-state publisher's default rate was never overridden, so every update reaching the visualisation and the transform tree was decimated before anything was drawn, giving a maximum step of 0.45 rad — nearly 26° — between consecutive updates.
2. **The driver ignored the planned timing entirely:** The trajectory execution layer advanced through the waypoints at a fixed 20 ms per point, discarding the velocity profile that had just been computed. This is more serious than an appearance problem: every trajectory executed at a flat 20 ms per waypoint regardless of its planned timing, so the velocity and acceleration scaling factors had no effect on the running system at all. The motion that resulted was effectively at the arm's full dynamic limits — it appeared fast because it was unphysical.

Both were corrected. The relay now runs at 100 Hz, reducing the maximum step per update to 0.0104 rad, and the driver walks the trajectory against the wall clock, interpolating between waypoints at 50 Hz. Motions now run at the speed they were planned to run at, which is the precondition for the scaling factors of Section 4.5.2 having any effect at all.

7.9.2 Joint Limits, and which Scaling Factor matters

Correcting the playback exposed a configuration fault beneath it. The joint limits are declared with a maximum velocity and acceleration of 3.0 for all seven joints, but the parameter carrying them was never passed to the planning node. The time parameterisation running inside that node — which times every manifold transit and every Cartesian segment — therefore fell back to a default acceleration of 1 rad/s² and then scaled it. Trajectories produced by the planning pipeline were unaffected, because that pipeline loads the limits itself, so the discrepancy was visible only as some motions being unaccountably slower than others.

With the limits correctly supplied, the behaviour is that predicted by Section 4.5.3: at these limits and over these distances no joint reaches its velocity bound, so the motion is entirely acceleration-limited, its duration varies with the inverse square root of the acceleration scale, and the velocity scale has essentially no effect. Of the two factors, it is the acceleration scale that determines how long a motion takes.

There is a general lesson in both of these faults. A quantity never measured end to end will be assumed to hold, and a scaling factor that silently does nothing is indistinguishable from one correctly applied, until something downstream is measured directly.

7.10 Command Handling, Dispensing and Status Reporting

The preceding sections describe how individual motions are planned and executed. This section describes the layer above them: how commands arrive, how a complete transfer is sequenced, how the liquid is dispensed, and how the outcome of each command is reported back to the operator interface.

7.10.1 The Command Loop and Outcome Reporting

Commands arrive asynchronously on a topic published by the graphical interface and are placed on a queue under a mutex; the main loop takes one at a time and runs it to completion before considering the next. The arrangement is deliberately serial: a robot arm has one physical state, and a system accepting a second command while the first was still executing would be planning from a configuration the arm is no longer in. The command set is small — a complete automated transfer for a given tube, a manual move to a marker or to raw coordinates, gripper open or close, the pour, return to home, a runtime toggle for the empty-gripper constraint policy of Section 7.7.1, and quit.

One ordering detail is worth recording, because the failure it caused was silent. The branch handling the complete-transfer command must be tested before the coordinate parser, not after. In an earlier version it sat afterwards, so a transfer command fell into the parser, failed to read three numbers from it, and reached the error path — leaving the entire automated-transfer branch as unreachable code that could never execute whatever the operator sent.

The interface also needs to know whether each command succeeded, and the difficulty is that the command loop has roughly a dozen early exits: bad arguments, an already-held tube, an unreadable robot state, a failed plan at any of several stages. Publishing a result explicitly at each is the kind of requirement met correctly when the code is written and violated the first time a new exit is added. The result is therefore published by an object whose destructor does it: a reporter is constructed at the top of each iteration, initialised to failure, and set to success only where success is actually established, so that whichever path the iteration takes — normal completion, early exit, or an exception — the object goes out of scope and the outcome is published exactly once.

Two properties follow, and both matter for a system with a human operator. The default is failure, so a path that forgets to record its outcome reports a failure rather than a spurious success — the safe direction for the error to run is the one leaving the operator informed that something did not work. And correctness does not depend on remembering, since a new early exit added later inherits the behaviour automatically.

7.10.2 Dispensing

The pour is a rotation of the final wrist joint through a right angle, held for one second, and reversed. Three aspects are worth noting.

It is executed in place: only the last joint moves, so the wrist centre does not translate and the tube tilts about its own held position. This is the one motion in the system where tilting the tube is the objective rather than the failure, and the upright constraint is necessarily inactive during it.

The upright value is read from the robot rather than assumed — the current joint value is measured first, the rotation applied relative to it, and the same measured value restored afterwards. Assuming a nominal upright angle would accumulate error over repeated pours and would be wrong outright if the transit had arrived on the opposite wrist branch, precisely the situation Section 7.7.2 establishes to be common. The return is checked separately. The rotation back to upright is a distinct motion with its own outcome, and its failure is reported distinctly: a pour that tilts successfully but fails to return has left the arm holding an open tube at ninety degrees, a materially different situation from one in which the pour never started. The command is reported successful only when the tube has been returned upright.

The tilt geometry is arranged so that the tube hangs beside the mixer and above its rim rather than over its centre, so that rotating it directs the contents into the mixer without the tube entering the vessel. A

diagnostic prints the distance from the mixer axis at which the tube will land, with a warning if that distance would place the tube inside the mixer or so far from it that the pour would miss.

7.10.3 Bookkeeping, and two implementations of the transfer

Whether a tube is currently held is not a detail of one command but a fact several parts of the system must agree on, and disagreement produces failures that are hard to localise. Three components need it: the command loop, to decide whether an upright transit is required and which velocity scaling applies; the scene updater of Section 7.8.1, to avoid republishing a world copy of a tube currently attached to the gripper, which would place a duplicate obstacle exactly where the gripper is; and the release command, to know what to detach.

The state is therefore updated in one place on each transition, and the automated transfer sets it on completion so that a subsequent manual command behaves correctly. After every motion the loop waits for the reported robot state to settle before the next command plans from it, since planning from a stale state produces a trajectory whose first point does not match where the arm is — the same class of defect as the bridging problem of Section 7.6.2.

The automated transfer itself exists in two forms, which differ in more than style. The imperative pipeline plans and executes each stage in turn, verifying the outcome of each before beginning the next; its advantage is precisely that structure, since the standoff verification and arrival check of Section 7.7.3 and the diagnostics of Section 7.7.2 all depend on being able to inspect the result of one stage and act on it before committing to the next. The declarative pipeline expresses the same transfer as a task constructor sequence, in which the stages are assembled and solved as a whole before any of them is executed. This is the more elegant formulation and has a real advantage: a failure at a late stage causes an alternative solution to be found for an earlier one, rather than the arm having already moved. The imperative pipeline is nevertheless the one used throughout this work, for the reason given — the diagnostic and verification mechanisms this chapter describes depend on stage-by-stage inspection, which the declarative formulation does not afford.

7.11 Scope and Evaluation

The three methods described in this chapter differ in kind rather than in degree, and the distinction is worth stating before any measurement is considered. The swept-volume surrogate fails not because it is slow but because it lacks the information to accept anything: not knowing the wrist joint values, it must be safe against all of them, and the resulting body is too large for the workspace it must move through. The manifold chart succeeds not because it searches better but because it searches a space in which violation of the constraint is impossible. The constrained baseline lies between the two, enforcing the constraint by sampling from a set of measure zero and paying for it in the manner Section 4.3 predicts.

No quantitative comparison of the three is offered here. As Section 7.8.3 records, the collision scene against which such a comparison would have to be conducted is known to differ from the physical workspace, and success rates and planning times measured against a scene of uncertain fidelity cannot support the conclusions that would be drawn from them. The comparison of the chart with the constrained baseline is therefore made on the physical arm and reported in Chapter 9. A planning-only benchmark of the chart against the baseline on common queries, conducted against the scene of the physical trials, is reported in Section 9.4.3.

Two properties are nevertheless established without measurement, since they follow from the constructions themselves. The manifold planner satisfies the constraint exactly with respect to the nominal kinematic model — not a result that could be improved upon by a better implementation or degraded by a worse one, since every point of the search space maps to a configuration at which the tube is vertical and no other configuration can be produced. What remains open is whether the motion between waypoints preserves that property, which is the subject of Section 7.6; the physical tilt of the

arm was not measured (Section 9.1.4). And the reduction cannot degrade the system: the dispatch of Section 7.7.1 attempts the manifold planner first and falls back to the constrained pipeline when it declines, which it does during goal generation rather than after exhausting a search, so the behaviour of the system with the reduction disabled is exactly its behaviour before the reduction existed.

The method is exact within its domain, and the domain has boundaries. The remainder of this section states them.

7.11.1 The Chart Commits to a Single Wrist Branch

The most significant limitation is a direct consequence of what makes the chart usable at all. As Section 7.6.1 established, the ZYZ inversion admits two branches, and the map from chart coordinates to configurations is continuous only if the branch is held fixed for the duration of a plan. A transit whose start and goal lie on opposite branches therefore cannot be represented within a single chart. Such a query is correctly declined and falls back to the constrained pipeline of Section 7.2: the system remains correct, but at the baseline's cost rather than the reduced method's.

This is not a rare corner case. At one tube's standoff pose, on the branch the arm was occupying, of fourteen candidate arm solutions eight were rejected by wrist joint limits and the remaining six were pressed against a joint stop: no usable goal configuration existed on that branch at all. The geometry of the tube rack effectively forces the opposite branch, which is the same mechanism that produced the descent failures of Section 7.7.2.

The consequence for a system designer is worth stating generally. A reduced-space planner built on an exact analytic chart commits the manipulator to a single wrist branch for the duration of each plan, and the workcell layout, home configuration and station poses must be arranged compatibly with that commitment. This is a real constraint on cell design, and it does not arise with a general constrained planner.

The natural remedy is an atlas. The two branches meet at the wrist singularity $q_6 = 0$, identified in Section 3.6.2 as the point where the ZYZ recovery becomes indeterminate, and two charts joined there would cover both branches and remove the restriction entirely. The construction is standard in the constrained-planning literature and the geometry of this manipulator supplies the join point explicitly; implementing it is the most valuable single extension to this work.

7.11.2 Conditions on the manipulator, the constraint and the task

- 1. Exact decoupling is assumed:** The chart recovers the wrist joints from an orientation on the assumption that the last three axes meet at a point. Chapter 3 established that this holds for the myArm 300 Pi by construction, and Section 7.3 confirmed it numerically to machine precision. A wrist whose axes only approximately intersect — through manufacturing tolerance, wear, or a design that never intended concurrency — would introduce a position error in the recovered configuration growing with the offset. The error remains bounded by the geometry rather than unbounded, but the method ceases to be exact, and with exactness goes the principal argument for preferring it.
- 2. The payload-to-reach ratio governs the swept-volume method:** As Section 7.4.4 established, the conservative bound scales with the reach of the end-effector while the free space available scales with the reach of the arm. The failure reported here is a statement about a 201 mm end-effector on a 243 mm arm, exercised with one payload in one workcell, and should not be read as a general verdict on the technique.
- 3. The constraint must leave a parameterisable residual freedom:** The upright requirement is close to the ideal case: it fixes two of three orientational degrees of freedom and leaves exactly one rotation, which becomes a single additional search coordinate. A constraint leaving a two-dimensional residual set — a tool axis required to lie within a cone, say — would enlarge the chart to six dimensions, retaining exactness but reducing the benefit relative to the seven-dimensional

baseline. A constraint whose satisfying set is irregular, not a smooth manifold or not parameterisable in closed form, does not admit this treatment at all, and the general projection-based methods remain the appropriate tool. The method is therefore not a replacement for general constrained planning but a specialisation: substantially better where its conditions hold, and inapplicable where they do not.

4. **The chapter concerns transit planning:** A complete transfer also involves grasp inverse kinematics, standoff selection and Cartesian segments, and Section 7.7.3 shows that after this reduction the transit accounts for a very small fraction of the time elapsed before the arm moves. An improvement in transit planning does not improve the whole task in proportion, and it would be misleading to imply otherwise.

7.11.3 Unresolved matters

Four items are recorded as open rather than closed.

1. **An inverse-kinematics timeout worth re-testing:** Reducing the solver timeout from 0.05 s to 0.02 s produced a substantial improvement in inverse-kinematics time. It was reverted because two tubes then failed outright, but that attribution is not established: the tube positions had been changed at approximately the same time, and a later run reproduced the same failure at the restored timeout, which suggests the position change was the cause. The two affected tubes sit 286 mm from the base against 251 mm and 230 mm for the others and are the marginal ones by a clear margin. The speedup is probably available, and the experiment should be repeated once those positions are confirmed reachable.
2. **A discrepancy in the modelled tube height:** As noted in Section 7.8.3, the vertical position of the tube collision cylinder is tuned against the physical system rather than derived from the table height and tube dimensions, and the two differ by 25 mm. Because a collision object placed above the real tube would permit contact the planner does not register, this should be settled by measurement.
3. **Empirical tolerance thresholds:** The joint-limit headroom and arrival tolerances of Section 7.7.3 were chosen to fall in the observed gap between configurations that stall a descent and those that do not. Both are defensible for this manipulator and this task, and neither derives from anything more general. A principled threshold would follow from the descent distance and the joint's remaining travel and would transfer to manipulators where these values do not.
4. **The declarative implementation has not been brought to parity:** As Section 7.10.3 describes, the transfer exists in a second, all-at-once formulation, which is shorter, easier to modify, and able to revise an early stage in response to a late failure rather than after the arm has moved. It nevertheless uses the constrained seven-dimensional planner throughout, performs neither standoff nor arrival verification, and requires the perception transform to be available where the imperative pipeline falls back to nominal positions; it should be understood as a working alternative formulation rather than a second evaluated method. Bringing the two to parity would answer an interesting question: whether the verification machinery of Section 7.7 can be expressed as stage-level alternatives, in which case the declarative form would be strictly better, or whether it genuinely requires the sequential inspection the imperative form provides.

7.12 Summary

The difficulty of this planning problem lay almost entirely in a single requirement: that an open tube of liquid remain upright throughout its motion, and not merely at the endpoints. Section 4.3 establishes why that requirement is expensive — the configurations satisfying it form a set of measure zero within the space a sampling-based planner explores — and Section 7.2 describes the conventional response, in which the constraint is enforced during search and verified afterwards.

Three architectures were built, the two reductions on the S-R-S structure of Chapter 3. Constrained search in the full joint space is the conventional approach and serves here as both baseline and fallback.

A conservative four-dimensional reduction, replacing everything distal of the wrist centre with a bounding body, is sound in construction and does not work on this manipulator; the cause is a ratio, a 201 mm end-effector on a 243 mm arm, and the analysis identifies the class of manipulators for which the same technique would serve. The third parameterises the single orientational freedom the constraint leaves and searches a five-dimensional space in which every point maps, in closed form, to a complete seven-joint configuration. On that space the constraint is satisfied identically rather than approximately, and the collision check is exact rather than conservative.

Making that construction usable required more than the planner. The chart is continuous only if the recovered wrist angles are unwrapped and the branch held fixed, and the traversability test belongs on the edge between samples rather than after planning — a distinction that turns a discarded plan into a routed one. A transit that succeeded by every measure available to the planner could still leave the arm at a joint stop from which the next motion was impossible, which is a reminder that a motion is correct only in relation to what follows it. And once planning ceased to be the bottleneck, the cost moved to where no earlier measurement had reason to look. The quantitative comparison of the chart with the constrained baseline is made on the physical arm and reported in Chapter 9.

8. Graphical User Interface

The development of a robotic system involving complex kinematic algorithms, motion planning and computer vision introduces, by design, a significant proficiency requirement. For the system to be practically viable in a synthetic chemistry laboratory, it must be accessible to researchers and chemists who may not possess advanced programming or robotic expertise. To address this, a custom Graphical User Interface (GUI) was developed.

Rather than acting as a simple terminal wrapper, the interface is implemented as a native ROS 2 node using the `rclpy` and `tkinter` libraries [71]. It serves as the communication bridge between the operator and the robotic laboratory assistant, translating high-level commands into ROS 2 messages published on the `/gui_commands` topic, which the control program of Chapter 7 subscribes to.

This chapter describes the design, software architecture and operation of the application. It covers the measures taken to protect the hardware from malformed or repeated input – thread management and input debouncing – and the implementation of the two operating modes: Auto Synthesis, in which a named chemical product is selected and the corresponding sequence of transfers is executed in full, and Manual Control, in which the operator either assembles a sequence of tubes or commands individual motions directly.

8.1 Design Philosophy and Interface Architecture

To allow the system to be operated without recourse to a terminal, a custom interface was developed using Python's `tkinter` and `ttk` libraries. The interface adopts a high-contrast dark theme, with a near-black background and a saturated blue for interactive and highlighted elements. This reduces visual fatigue during prolonged laboratory sessions and keeps the active controls and status text readily distinguishable.

As shown in Figure 8.1, the application window is divided into two principal sections. The header occupies the top of the window and serves as the visual identity of the system, displaying "CHEMICAL SYNTHESIS" with the subtitle "AUTOMATED FLUID HANDLING SYSTEM", separated from the panel below by a horizontal rule. It is structurally isolated from the interactive elements and does not change in response to the operator's actions, which gives the application a consistent frame regardless of the mode selected.

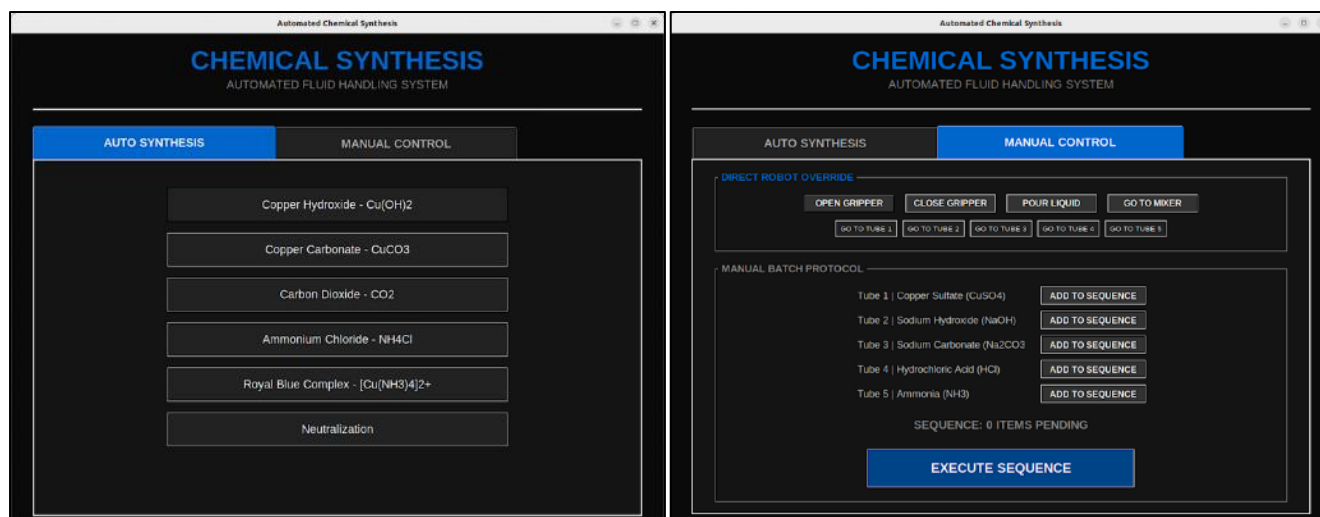


Figure 8.1: Overall view of the GUI, displaying the Auto Synthesis and the Manual Control operating modes.

Directly below lies the primary interaction zone, a `ttk.Notebook` widget forming a tabbed container that divides the system into its two operating paradigms. The tabs are prominently labelled and use colour to indicate which is currently active, and the content of the central panel is rebuilt for each tab, so that the frames, buttons and layout the operator sees correspond to the mode selected. The purpose of the

division is to keep the two kinds of interaction visually distinct: in one, the operator selects a chemical product and the system determines the sequence of operations required to produce it; in the other, the operator specifies the operations themselves.

The separation is one of presentation rather than of mechanism. Both tabs publish to the same command topic, and the tab in view does not restrict what may be sent; the measures that prevent conflicting commands from being issued are described later in this chapter.

8.2 ROS 2 Network Integration and Node Architecture

A design requirement for the user interface was direct, low-latency communication with the robotic control architecture. Rather than executing terminal scripts or relying on an intermediary socket server, the GUI is implemented as a native participant in the ROS 2 ecosystem. Using the `rc1py` client library, the application initialises its own node, designated `gui_commander_node`. The interface runs on the same personal computer as the remainder of the ROS 2 stack described in Section 5.5 and communicates with the controller over the local ROS 2 network rather than across the Ethernet link to the arm. The complete source of the interface is given in the Appendix (Chapter 12).

8.2.1 The Publisher Pipeline

Communication runs in both directions. On initialisation, the GUI establishes a publisher on the `/gui_commands` topic, using the standard `std_msgs/msg/String` message type, and a subscription to `/gui_status`, on which the controller of Chapter 7 reports the outcome of every command as OK or FAIL (Section 7.10.1) [51].

Sequences use that channel to pace themselves. Each command of a sequence is sent through a `send_and_wait` method, which publishes it and blocks until the controller reports its outcome or a timeout expires – 180 s for a move, 90 s for a gripper action or a pour. A failure or a timeout stops the sequence at that step, so that a failed grasp is not followed by a transfer to the mixer. The outcome is recorded in the interface's terminal log but not displayed in the window, and commands issued individually from the override panel are sent without waiting for it.

When an operator triggers an action – whether through an automated recipe or a manual button press – the request is routed through a single `send_command` method. This function encapsulates the raw command string, such as `m 5`, `o` or `p` into a `String` message and publishes it. The centralised routing is advantageous for maintainability, since any subsequent change to the messaging protocol need be made in only one place.

Table 8.1: The command vocabulary published on `/gui_commands`

Command	Meaning	Issued from
o	Open the gripper	Recipe Expansion, Manual Sequence, Override Panel
c	Close the gripper on the tube	Recipe Expansion, Manual Sequence, Override Panel
p	Execute the pour above the mixing vessel	Recipe Expansion, Manual Sequence, Override Panel
m N	Move to the object identified by the marker N (where N is 1-5 for the test tubes and 6 for the mixing vessel)	Recipe Expansion, Manual Sequence, Override Panel

The `send_command` method also incorporates a debouncing mechanism for manual input. A physical double-click on a button would otherwise place two identical commands in the controller's queue, causing the same motion to be attempted twice in succession. By recording the last manual command and the time at which it was issued, the function discards a command identical to its immediate predecessor if the two occur within 0.3 s of one another.

Two limitations of this mechanism should be noted. It compares only against the immediately preceding command, so two different commands issued in rapid succession are both transmitted; and it is applied only to manual input, the commands generated by an automatic recipe being sent unconditionally. It is therefore a guard against accidental repetition of a single button press, and not a general rate limit on the command channel.

8.2.2 Lifecycle Management and Clean Shutdown

Because the GUI operates as an active ROS 2 node, terminating the application window without shutting the node down would leave it registered on the network, which can produce spurious traffic or conflicts when the interface is next started. The interface therefore binds to the window manager's WM_DELETE_WINDOW protocol, intercepting the close event and invoking an `on_close` function which destroys `gui_commander_node` and shuts down the `rc1py` context before the window itself is destroyed and the process exits.

Together with the command pipeline, this allows the interface to operate as a stable participant in the distributed network rather than an external script driving it. Figure 8.2 shows the resulting architecture, mapping the flow of commands from the operator interface, through ROS 2 middleware, to the C++ execution layer and the hardware beneath it.

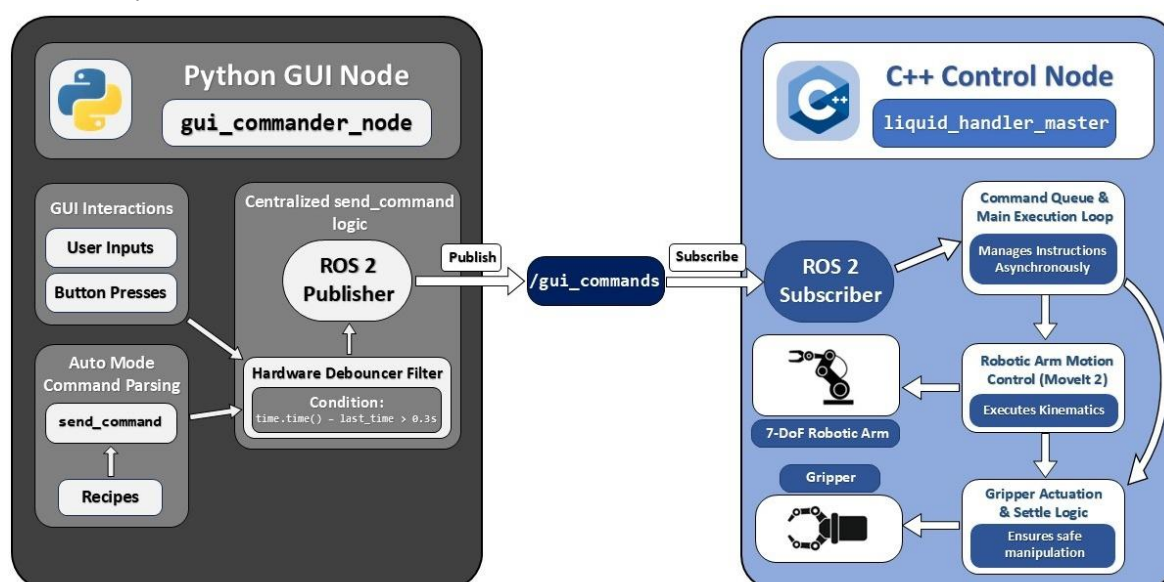


Figure 8.2: Node communication architecture and internal data flow, showing the ROS 2 integration between the graphical user interface layer and the hardware execution layer.

8.3 Auto Synthesis Mode

The Auto Synthesis mode automates standard and repeatable chemical protocols. By removing the need for the operator to select tubes individually and in the correct order, it reduces the opportunity for error and shortens the time required to execute a procedure.

8.3.1 Data Structure and Predefined Recipes

At the core of the Auto Synthesis logic are two dictionaries. The first maps each test tube to the reagent it contains; the second maps the name of a chemical product to the ordered list of tubes whose contents are to be transferred to the mixing vessel.

Table 8.2: Assignment of reagents to test tubes

Tube	Reagent
1	Copper Sulfate (CuSO_4)
2	Sodium Hydroxide (NaOH)
3	Sodium Carbonate (Na_2CO_3)
4	Hydrochloric Acid (HCl)
5	Ammonia (NH_3)

Table 8.3: Predefined recipes and the tubes each requires

Recipe	Tubes
Copper Hydroxide, ($\text{Cu}(\text{OH})_2$)	1, 2
Copper Carbonate, (CuCO_3)	1, 3
Carbon Dioxide, (CO_2)	3, 4
Ammonium Chloride, (NH_4Cl)	4, 5
Tetraamminecopper(II) Complex, ($[\text{Cu}(\text{NH}_3)_4]^{2+}$)	1, 5
Neutralisation	4, 2

This structure keeps the protocols separate from the control logic: a new synthesis is added by inserting an entry in the recipe dictionary, without modification to the code that executes it. The generality is bounded by what the format can express, namely which tubes are used and in what order. A protocol requiring a different sequence of operations, or a specified quantity of each reagent, would require the execution routine itself to be extended.

8.3.2 Macro Expansion and Thread Management

When the operator selects a recipe, the system acts as a macro expander. The `run_recipe` routine retrieves the corresponding list of tubes and passes each in turn to the `_run_tube` method, which generates the sequence of commands the C++ controller expects.

For one tube that sequence is: move to the tube (m followed by its identifier), close the gripper on it (c), move to the mixing vessel (m6), execute the pour (p), return to the tube's own position, and open the gripper to release it. Six commands per tube, and one such group for each tube named in the recipe.

Each command is sent through the `send_and_wait` method of Section 8.2.1, so that the next is issued only once the controller has reported the previous one as successful. A sequence therefore advances at the pace of the arm, and stops at the first step that fails or times out.

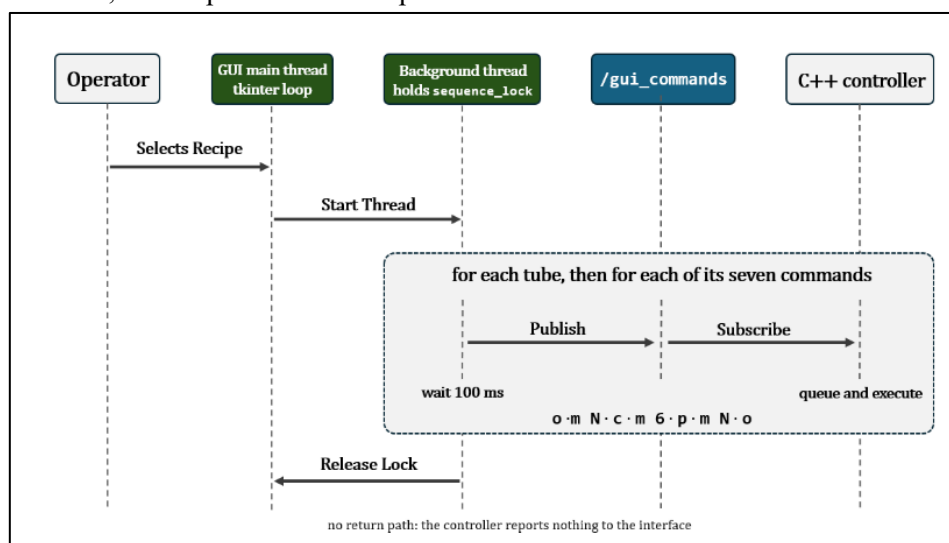


Figure 8.3: The Sequence Diagram.

Two aspects of the execution deserve comment. The sequence is executed on a background thread, so that the tkinter main loop continues to service the interface while a procedure that may take several minutes is in progress. Without this the window would cease to redraw and would appear to the operating system to have stopped responding.

The thread body attempts to acquire a mutex, `sequence_lock`, before starting. If another sequence already holds it, the new request is ignored and logged rather than queued, so the commands of two sequences are never interleaved. The operator has no means of cancelling a procedure once it has been started, other than closing the interface. Nor does the lock govern the manual controls of Section 8.4, which publish directly and may therefore be issued while an automatic sequence is in progress.

8.4 Manual Control and Batch Processing

The Manual Control mode supports procedures that are not among the predefined recipes. Rather than selecting a chemical product and allowing the system to determine the transfers required, the operator specifies the transfers directly, either by assembling a sequence of tubes to be processed in turn or by commanding individual motions of the arm.

8.4.1 Construction of the Sequence

The lower panel of the Manual Control tab lists the five tubes together with the reagents assigned to them, each with a control that appends it to a pending sequence. A tube may be added more than once, and the order in which tubes are added is the order in which they will be processed.

Each addition updates a counter displayed above the execution control, which reports the number of items currently pending. This gives the operator a running indication of the sequence being assembled and serves as a check before the procedure is committed.

The sequence is executed by the "EXECUTE SEQUENCE" control. If the pending list is empty, a warning dialogue is displayed and no command is issued. Otherwise, the list is copied, and the pending sequence is cleared immediately, before execution begins. Clearing at this point rather than on completion has a specific purpose: the procedure runs on a background thread and may take several minutes, during which the interface remains responsive, and the operator is free to begin assembling the next sequence. Were the list cleared afterwards, additions made during execution would be discarded along with the items that had already been processed. A sequence executed while another is still running is, however, ignored rather than queued (Section 8.3.2), and since the pending list has already been cleared, its contents are lost and must be added again.

The copied list is then traversed in order; each tube being expanded into the same six-command sequence described in Section 8.3.2 and dispatched through the publisher pipeline of Section 8.2. As in the automatic mode, the traversal is performed on a background thread holding `sequence_lock`, so that a manual sequence and automatic recipe cannot be interleaved with one another.

8.4.2 Direct Override

The upper panel of the tab provides direct control over individual operations: opening and closing the gripper, executing a pour, and moving the arm to any of the five tubes or to the mixing vessel. Each control publishes a single command, and these are the commands to which the debouncing of Section 8.2.1 applies.

These controls are not governed by `sequence_lock`. They publish directly and may therefore be issued while a recipe or a manual sequence is in progress, in which case the command enters the controller's queue alongside those of the running procedure. The mode is intended for testing individual motions and for recovering the arm to a known state, and it places the responsibility for issuing commands in a sensible order with the operator.

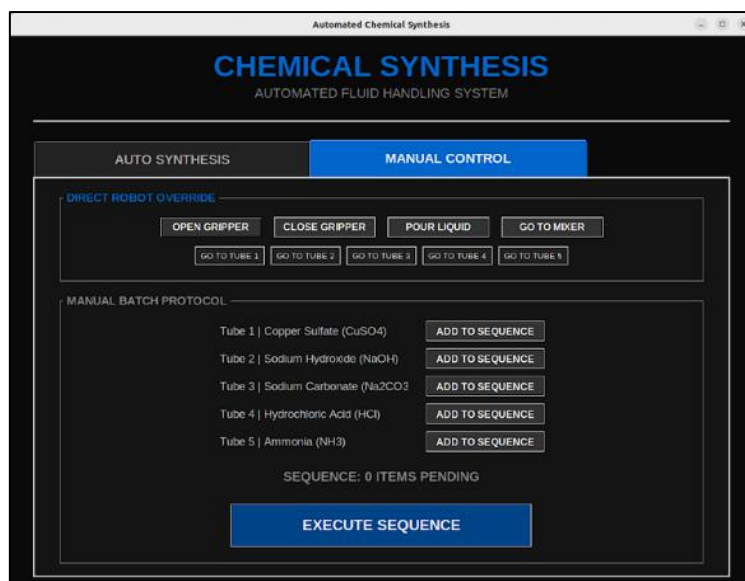


Figure 8.4: the Manual Control close-up.

8.5 Safety Mechanisms and their Limits

The interface commands a physical manipulator carrying open vessels of liquid, and a command issued in error is executed as readily as one issued deliberately. Four mechanisms in the interface guard against the more likely forms of erroneous input: the input debouncing of Section 8.2.1, the gating of each step of a sequence on the outcome of the previous one, also described there, and the mutual exclusion of sequences under `sequence_lock` and their execution on background threads, both described in Section 8.3.2. Three details of their design should be added here.

The debounce threshold of 0.3 s is chosen to be longer than the interval between the two events of an accidental double-click and shorter than any interval at which an operator would deliberately repeat a command. The mutual exclusion guards against a specific failure: were two procedures dispatched concurrently, their commands would arrive interleaved, and the resulting sequence — a gripper opened between a grasp and a transfer, or a move to the mixer issued while the gripper is empty — would be executed as given. And the background threads are created as daemon threads, so that they do not prevent the process from terminating when the window is closed.

What matters more than the mechanisms, however, is the boundary they do not reach.

8.5.1 What is not Guarded

The mechanisms above address input that is repeated or concurrent. Four classes of risk remain, and they are recorded here because they bound what the interface can be relied upon to prevent.

1. **There is no means of stopping a procedure once it has begun:** The execution routines provide no control by which a running sequence may be cancelled. Closing the interface stops the sequence before its next command, but does not interrupt the motion in progress; otherwise the operator must intervene at the controller or at the arm's power supply.
2. **The direct override controls are not serialised:** They publish immediately and do not acquire the mutex, so a manual command issued during a running procedure enters the controller's queue alongside the commands of that procedure. This is deliberate, in that the override panel exists to command individual motions independently of any sequence, but it means the separation between the two modes is one of presentation rather than of mechanism.
3. **Commands are not validated beyond a check that the pending sequence is non-empty:** The interface does not verify that a requested motion is kinematically feasible, that the gripper is in an appropriate state for the command issued, or that the target is where the system believes it to be. Such verification is performed by the controller of Chapter 7, which plans against the current scene

and declines a motion it cannot execute; the interface does not anticipate those refusals, and registers them only as a failed step.

4. **Outcomes are used but not shown:** The controller reports the outcome of every command, and sequences stop at the first failure, but the result is written only to the interface's terminal log and not displayed in the window. Commands issued from the override panel do not wait for their outcome at all, so the operator's only indication of their progress is the motion of the arm itself.

The first of these is the most consequential. A procedure that is proceeding wrongly can be stopped only by closing the interface or at the arm itself. An abort control, together with a display of the stage reached and the outcome of each command, would build on the status channel that already exists, and is discussed further in Chapter 10.

8.6 Summary

The interface is implemented as a ROS 2 node in Python publishing single-string commands on a topic to which the C++ controller of Chapter 7 subscribes, and receiving the outcome of each on a second topic. It is not a wrapper around a terminal session but a participant in the same network as the rest of the system, which is what allows a command issued by a button press to reach the controller by the same path as one issued by any other node.

Two modes of operation are provided. In Auto Synthesis the operator selects a chemical product, and the recipe associated with it — an ordered list of tubes — is expanded into the sequence of gripper and motion commands required to carry each tube to the mixing vessel, pour its contents and return it. In Manual Control the operator either assembles such a sequence directly, choosing the tubes and their order, or commands individual motions through the override panel. The distinction between the two is what determines the sequence: in the first the system derives it from the product requested, and in the second the operator supplies it.

Of the risks the interface does not address, the absence of any means of cancelling a procedure once it has started is the most significant.

9. System Integration and Experimental Results

The preceding chapters describe a system and the reasoning behind its construction. Chapter 7 develops three planning architectures and establishes what each guarantees, but defers the comparison between them, for the reason given in Section 7.8.3: the collision scene against which such a comparison would have to be conducted is known to differ from the physical workspace, and figures obtained against it would not support the conclusions drawn from them.

This chapter reports the evaluation of the assembled system on the physical arm. It sets out the protocol, the quantities measured and the criteria by which each trial was judged. It then reports the results in three groups, the localisation of the objects, the outcome of the transfers, and the behaviour of the planning methods, and closes with an account of the failures observed and their causes.

Claims established analytically in earlier chapters are not re-examined here. The exactness of the closed-form solver was settled by the checks of Section 7.3, which compare one computation against another and are independent of the workspace. The dimension of the constraint manifold is a property of the constraint. What is at issue here is whether the system, perceiving the bench through the camera and executing on the arm, performs the task it was built for.

9.1 Experimental Protocol

The purpose of this section is to fix, in advance of the results, what was measured and what counted as a success, so that the criteria cannot be read as having been chosen to suit the outcome.

9.1.1 Conditions

All experiments were conducted on the setup described in Chapter 5, in its normal configuration: five tubes in their support bases, the mixing vessel with its funnel, the camera on its tripod, and the arm mounted to the bench. The camera was calibrated once, according to the procedure of Section 6.2, and the calibration was not repeated between trials.

The scanning procedure of Section 6.6 was performed before the trials began, and the resulting positions held throughout. The reference marker was placed at its measured position and left undisturbed.

The tubes were charged with coloured water rather than with the reagents named in the interface, since the object of the evaluation is the mechanical and geometric behaviour of the system rather than the chemistry, and a coloured but inert liquid makes spillage visible without introducing hazard. Each tube was filled to 9 ml.

Velocity and acceleration scaling were those given in Section 4.5.2 throughout: 30% with an empty gripper and 15% while carrying. All trials were conducted after the execution-layer corrections of Section 7.9.1, without which the scaling factors have no effect on the executed motion.

9.1.2 The Unit of Trial

A transfer is the unit of measurement. It comprises the complete sequence of Section 7.1.2 for a single tube: transit to the standoff, descent, grasp, lift, transit to the mixing vessel, pour, return transit, descent, release and retreat.

A transfer begins with the gripper empty and open, and ends with the tube returned to its base and the arm withdrawn to the standoff above it. The arm does not return to a home configuration between transfers, so each begins from wherever the preceding one left it. The transits in the results that follow are consequently station-to-station motions from a start configuration that varies between trials, which is how the system operates in normal use.

Sixteen transfers were executed in total, distributed across the five tubes: four at tube 1 and three at each of tubes 2 to 5. Each was commanded individually, through the manual mode of Section 8.4.

9.1.3 Outcomes

Each transfer was assigned exactly one of the following outcomes, judged by observation and by the system's own logs.

1. **Complete.** The tube was grasped, carried, emptied into the mixing vessel and returned to its base, with no liquid deposited outside the vessel and no contact between the arm and any object other than the tube it was handling.
2. **Complete with spillage.** The sequence completed as above, but liquid was deposited outside the mixing vessel at any stage.
3. **Grasp failure.** The gripper closed without securing the tube or secured it in a position from which the subsequent motion was not executable.
4. **Placement failure.** The tube was not returned to its base, whether because it was released outside the base, because it jammed against the rim, or because it was displaced from the base on release.
5. **Planning failure.** The system declined to execute a stage, reporting that no plan was found. The stage at which the failure occurred, and the reason reported, were recorded.
6. **Aborted.** The trial was stopped by the operator to prevent damage.

The outcomes are ordered by severity, and where more than one would apply the later one is recorded.

9.1.4 Measured Quantities

1. **Localisation accuracy.** For each marker, the position reported by the vision system was compared against a position measured by hand relative to the robot's base. The per-marker diagnostics produced by the scanning procedure – frame-to-frame scatter, out-of-plane residual, and deviation from the shared orientation — are reported alongside.
2. **Grasp and placement reliability.** The outcome of each transfer, in the categories above. Placement is considered separately from grasping because the chamfered support bases of Section 5.4.2 were introduced specifically to absorb positioning error on insertion, and the placement result is what establishes whether they do.
3. **Planning behaviour.** For each transit, which method produced the plan and how long it took: the reduced-space chart of Section 7.5, the constrained baseline of Section 7.2 at either of its tolerances, or the verified unconstrained rung.
4. **Planned tilt.** The maximum deviation of the tube's axis from the vertical over each trajectory, computed by the roll-invariant test of Section 7.2.3 from the joint values of the planned trajectory at its waypoints. It is a property of the commanded motion, not a measurement of the arm. No independent measurement of the physical tilt was made, and none could be derived from the system, since the joint state it publishes is the commanded position rather than one read from the encoders (Section 5.5.2). The physical tilt differs from the planned one by the backlash in the joints (Section 9.5.3), whose magnitude was not measured. Only its consequence, that no liquid was spilled in any trial, was observed. For transits planned by the chart, the figure covers the path on the constraint manifold only. Where the arm was not already on the manifold, a short joint-space bridge of at most 0.25 rad per joint precedes that path (Section 7.6.2). The bridge is checked for collision but not for tilt, and is not included in the figure. While a tube is held the arm is already on the manifold from the preceding upright motion, so a bridge can occur only on transits with an empty gripper.
5. **Cycle time.** The elapsed duration of a complete transfer, together with the decomposition reported by the system into its constituent stages.

9.1.5 Design of the Comparison

A comparison in which one method is the author's own contribution, and the other is the alternative it replaces is the situation in which a flattering result is easiest to produce by accident. Four conditions bear on the fairness of the comparison. The first and third were satisfied, the second only in part, and the fourth not at all.

1. **Every query is feasible.** All the transits reported are those the task itself generates; between stations the arm reaches in ordinary operation. A query set containing unreachable targets would manufacture an arbitrarily large speedup, since the reduced planner rejects such a target during goal generation while a sampling-based planner exhausts its time budget before concluding the same thing — a difference that says nothing about planning.
2. **Both methods see the same scene, but not the same queries.** The two were not exercised in separate passes but as the dispatch ladder of Section 7.7.1 selected them, so that each faced the same bench, the same scan and the same sessions. The queries themselves, however, were not shared. The chart handled every transit it could represent, and the baseline received only those it declined: the transits to the two nearest tubes, where every goal configuration lies close to a joint stop. The planning times of Section 9.4.1 are therefore descriptive; a controlled comparison on common queries is reported in Section 9.4.3.
3. **The baseline is configured as deployed.** The goal tolerances and the number of planning attempts are those the system uses in operation, not library defaults, which would represent a materially harder problem than the baseline actually solves in normal use.
4. **No control was run.** A comparison with the orientation constraint removed altogether would separate the hypothesis that the constraint is the source of the difficulty from the alternative that the task or the configuration is at fault. No such trial was conducted, and that claim therefore rests on the analysis of Section 4.3 rather than on measurement reported here.

9.1.6 What is not Evaluated

The evaluation is conducted with a single manipulator, a single camera and a single arrangement of the workbench. Nothing here establishes how the system would behave on different hardware, and the negative result of Section 7.4 is precisely a case in which the outcome depends on the proportions of the particular arm.

Each transfer was commanded individually. A recipe was executed successfully in automatic mode but was not instrumented, so the behaviour of a multi-tube sequence, in particular whether the positional error identified in Section 9.5.3 accumulates across consecutive transfers, was not measured.

No chemistry is performed. The liquids are inert, the quantities are not metered, and no property of the resulting mixture is measured.

The trials are conducted under supervision, with an operator present who could intervene. The system has not been evaluated in unattended operation, and Section 8.5.1 records why it is not presently suited to it.

In particular, the proportion of transfers completed is reported as the outcome of this proof-of-concept evaluation and not as a measure of the system's reliability.

9.2 Localisation Accuracy

This section assesses the localisation pipeline of Chapter 6 in three steps. The calibration on which it rests is recorded first. The diagnostics reported by the scan itself are then examined, which characterise the repeatability of the procedure and test the assumptions of the method. Finally, the reported positions are compared against positions measured by hand, which bears on absolute accuracy, subject to the uncertainty of that measurement. The distinction between repeatability and accuracy is maintained throughout, since the two are affected by different sources of error.

9.2.1 Calibration

The calibration described in Section 6.2 was performed once, over 60 images of the chessboard pattern, and was not repeated between trials (Section 9.1.1). It returned an RMS reprojection error of 0.31 pixels, focal lengths agreeing to within 0.2 %, and a principal point displaced from the centre of the image by approximately 49 pixels horizontally and 10 vertically, as reported in Section 6.2.3.

At the focal length of approximately 1094 pixels, a reprojection error of this size corresponds to well under a millimetre at the working distance of the camera, an order of magnitude smaller than the discrepancies reported in Section 9.2.3. The fit of the calibration is therefore not a limiting factor for the localisation. The reprojection error measures only how well the model reproduces the calibration images, however, and says nothing directly about the accuracy of positions recovered from other images. That is assessed by the remainder of this section.

9.2.2 Scan Diagnostics

The scanning procedure reports, for each marker, the number of frames in which it was observed and retained, the frame-to-frame scatter of its recovered position, its out-of-plane residual when solved without the coplanarity constraint, and the deviation of its own orientation from the shared one.

Table 9.1: Per-marker diagnostics from the scan used throughout the trials.

Marker	Frames Seen	Frames Used	Scatter (mm)	Out-of-plane (mm)	Orientation Deviation (°)
0	89	89	0.00	-	4.94°
1	89	89	0.48	-37.05	2.45°
2	89	89	0.52	-35.11	3.96°
3	89	89	0.56	-35.47	1.49°
4	89	89	0.51	-24.78	2.56°
5	89	89	0.46	-2.65	7.34°
6	89	89	0.35	470.37	87.62°

The scan completed on 89 frames, the procedure having reached its time limit before the nominal frame count was accumulated. All 89 were retained for every marker, so no observation was rejected as an outlier.

The frame-to-frame scatter lies between 0.35 and 0.56 mm and is essentially uniform across the markers. This bounds the random component of the localisation error, which is accordingly well below the tolerances the task requires. It does not bound any error common to every frame, and the comparison of Section 9.2.3 addresses that separately.

The scatter is a measure of repeatability within a single scan: how closely the procedure agrees with itself from one frame to the next. It is not a measure of accuracy, and it is not the repeatability of the procedure as a whole, which would be established by repeating the scan on an undisturbed bench. Since the trials used a single scan, that was not measured.

The out-of-plane residual is the distance of each marker from the plane of the reference marker, measured perpendicular to it, when the marker's position is solved without the coplanarity constraint. Since all five tube markers lie flat on the same working surface as the reference, the true value is zero in every case, and the figures of 25 to 37 mm for markers 1 to 4, and 2.7 mm for marker 5, are therefore error rather than displacement. That they are of consistent sign and similar magnitude is what would be expected of an error in the direction of the camera's line of sight, which Section 6.4.1 identifies as the direction in which a small marker's position is least well determined. This is precisely the quantity that the coplanarity constraint removes from the answer, and its magnitude is the measure of what that constraint is worth.

Two markers depart from the assumptions of the method. Marker 5 deviates from the shared orientation by 7.3°, which leaves its position usable but indicates that it is either not mounted exactly parallel to the others or is observed from an unfavourable angle. Marker 6, which identifies the mixing vessel, deviates by 87.6° and is placed 470 mm from the reference plane by the unconstrained solve, although it was laid flat alongside the others with its axes aligned to theirs. The two figures are not independent:

the unconstrained solve applies the shared rotation, so a marker whose recovered orientation departs from the consensus by close to a right angle is being solved with a rotation that does not describe it, and the position returned is correspondingly unreliable. The cause of the orientation discrepancy was not established, and marker 6's position should be regarded as the least trustworthy of the seven.

9.2.3 Comparison against Measured Positions

The scatter above bounds the random component of the error only: it characterises repeatability, not absolute accuracy. An error common to every frame, such as an incorrect marker size, an inaccurate calibration or a mismeasured reference position, is a bias, and no amount of averaging reveals it. The positions reported by the system were therefore compared against positions measured by hand relative to the robot's base.

Table 9.2: Reported positions against hand measurement.

Marker	Measured (m)	Reported (m)	Δx (mm)	Δy (mm)	Error (mm)	Distance from reference (mm)
0	(0,-0.07,0)	(0,-0.07,0)	-	-	-	0
1	(-0.0965,-0.342,0)	(-0.0985,-0.3245,0)	-2.0	+17.5	17.6	289
2	(-0.0334,-0.345,0)	(-0.0323,-0.3524,0)	+1.1	-7.4	7.5	277
3	(0.042, -0.375,0)	(0.0343,-0.3753,0)	-7.7	-0.3	7.7	308
4	(0.097,-0.352,0)	(0.1010,-0.3479,0)	+4.0	+4.1	5.7	298
5	(0.179,-0.32,0)	(0.1753,-0.3172,0)	-3.7	+2.8	4.6	307
6	(-0.199, -0.221,0)	(-0.1919,-0.2353,0)	+7.1	-14.3	16.0	250

The discrepancies range from 4.6 to 17.6 mm, with a median of 7.6 mm. Every one of them is an order of magnitude larger than the frame-to-frame scatter of Table 9.1, which lies between 0.35 and 0.56 mm. The hand measurements are not exact, and the discrepancies are therefore the combined error of the vision system and of the ground truth, which this comparison cannot separate. Each position was measured once, with a ruler, along the two horizontal axes of the base frame. The uncertainty of that measurement has four main components:

- reading the ruler, about ± 1 mm;
- locating the origin of the base frame, which lies on the axis of the first joint and is not physically marked, about ± 2 mm;
- locating the centre of each marker, about ± 1 mm;
- keeping the ruler aligned with the axes of the base frame, where an angular error of 1° displaces a point 300 mm away by about 5 mm.

Combined in quadrature, these give an uncertainty of roughly ± 6 mm in each coordinate. Since each position was measured only once, this is an estimate rather than a measured spread. Discrepancies of that order cannot be attributed to the vision system with any confidence; only the two largest, at markers 1 and 6, clearly exceed it.

The discrepancies also have no common component. Their mean over the six object markers is $(-0.2, +0.4)$ mm. They follow neither the pattern of a rotation about the reference marker nor that of a scale error, both of which would displace markers lying at similar distances from the reference in a consistent direction. The error is therefore specific to each marker: constant from frame to frame, but different from one marker to the next. An error in the measured position of the reference marker would appear as an offset common to every marker, and the mean above bounds it to a few millimetres; the standard error of that mean is about 2 mm in x and 4 mm in y. The marker-specific sources that remain are the ground truth itself, the mounting of each marker, and, for marker 6 and to a lesser extent marker 5, the departure from the shared orientation identified in Section 9.2.2.

Marker 6 disagrees by 16.0 mm, among the largest in the set despite being the closest to the reference. This is consistent with the diagnostic of Section 9.2.2, which identified it as the one marker whose recovered orientation departs substantially from the shared estimate. Its position should accordingly be treated as less reliable than those of the tube markers, which bear directly on the grasping accuracy examined in Section 9.3.

The absolute accuracy of the localisation is therefore established only to within the uncertainty of the ground truth. The reported positions agree with the hand measurements to within 4.6 to 17.6 mm, of which several millimetres may belong to the measurement rather than to the system. Separating the two would require a better reference, such as a calibrated fixture, or the arm itself touching each marker, subject to the arm's own accuracy.

9.3 Results of the Transfers

Table 9.3: Outcome and planning behaviour of each of the sixteen transfers. Planning times are given as a range where several transits were handled by the same method; a dash indicates that the method was not invoked.

#	Tube	Outcome	Transits by chart	Planning, chart (ms)	Planning, baseline (ms)	Max planned tilt (°)	Reconfig	Time
1	1	Complete	1/3	29.1	884, 3209	16.6	-	1:19
2	1	Placement Failure	1/3	33.0	457, 1424	12.5	-	1:14
3	1	Complete	1/3	28.0	667, 3887	10.6	-	1:15
4	1	Complete	1/3	25.7	943, 3381	16.1	-	1:07
5	2	Placement Failure	1/3	36.0	1063, 7055	17.7	-	1:19
6	2	Complete	1/3	22.6	1809, 6378	16.8	-	1:31
7	2	Complete	1/3	24.1	6215, 6950	16.2	-	1:27
8	3	Complete	3/3	25.9-47.4	-	0.00033	-	1:25
9	3	Complete	3/3	40.0-47.2	-	0.00033	-	1:14
10	3	Complete	3/3	15.9-29.1	-	0.00033	-	0:55
11	4	Complete	3/3	25.5-61.9	-	0.00033	-	1:32
12	4	Complete	3/3	15.7-51.6	-	0.00033	-	1:35
13	4	Complete	3/3	19.0-80.5	-	0.00033	-	1:33
14	5	Complete	3/3	7.7-34.0	-	0.00028	-	1:15
15	5	Complete	3/3	7.8-64.7	-	0.00035	Yes	1:26
16	5	Complete	2/3	18.3-45.3	589 (loose)	33.0	-	1:17

Fourteen of the sixteen transfers completed. The two exceptions were placement failures, in which the tube was released a few millimetres outside its support base but remained upright. No tube was dropped, no grasp failed, no planning attempt was abandoned, and no trial required intervention.

These figures describe the sixteen trials reported here, which were conducted on one bench, from a single scan and under supervision. They demonstrate that the task can be performed by this system; they do not estimate how reliably it is performed. Even if the trials were independent, sixteen of them would bound the success rate only loosely: the exact 95 % confidence interval for fourteen successes in sixteen extends from about 62 % to 98 %. The trials are not independent, moreover, since they share a bench, a scan and a session, and the interval understates the uncertainty accordingly.

The most striking feature of the table is the column recording how many of each transfer's three transits were handled by the reduced-space chart. It takes only two values, and they divide the tubes cleanly. For tubes 1 and 2 the chart handled one transit of three on every occasion, the transit to the mixing

vessel, and declined both transits to the tube itself. For tubes 3, 4 and 5 it handled all three, with a single exception discussed below. The division is by position rather than by trial. The behaviour was identical across every repetition at a given tube.

The reason is recorded in the logs and is the limitation set out in Section 7.11.1. At tubes 1 and 2 the chart's goal generation returned no usable configuration: of the arm inverse-kinematic solutions available at the standoff pose, every one lay against a joint stop on the wrist branch the arm occupied, and configurations at a joint stop are rejected because there is no travel left for the descent that follows. At tubes 3, 4 and 5 the same procedure returned eight goal states with substantial joint headroom. The geometry of the two nearer tube positions leaves the manipulator no comfortable posture from which to reach them.

The consequences follow the same division. Planning by the chart took between 7.7 and 80.5 ms and produced a maximum planned tool tilt never exceeding 0.00035° . Planning by the constrained baseline, for the transits the chart declined, took between 457 and 7055 ms, and produced tilts between 10.6° and 17.7° — in one case exceeding the 0.30 rad tolerance the planner had been asked to respect. Both methods were exercised within the same session, against the same scan of the same bench, minutes apart. The queries differed, however, a qualification taken up in Sections 9.4.1 and 9.4.3.

Both placement failures occurred at tubes 1 and 2, and both on the return transit, which at those tubes is planned by the baseline. The nine transfers at tubes 3, 4 and 5 completed without exception. The sample is too small to express this as a rate, but the association is consistent with the analysis of Section 9.5: the return transit planned by the baseline arrives in a configuration chosen without preference, and the accumulated positional error at release is correspondingly less controlled.

Three individual trials departed from the pattern and are treated separately in Section 9.5: trial 7, in which the carried tube touched the funnel above the mixing vessel, trial 15, in which the in-place reconfiguration of Section 7.7.3 was triggered on arrival at the standoff and trial 16, in which a wrist-branch mismatch forced the dispatch ladder down to its loose tolerance and the carried tube tilted 33° . A recipe was also executed in automatic mode, in which the sequence of transfers associated with a named product is issued without further intervention. The procedure completed correctly. No measurements were taken during this run and it is not included in Table 9.3, so it is reported here only as confirmation that the recipe expansion of Section 8.3.2 operates as described and that consecutive transfers execute without the operator commanding each stage.

9.4 Planning Behaviour

9.4.1 The Two Methods on the Physical Task

The sixteen transfers comprise forty-eight transits, each of which was dispatched through the ladder of Section 7.7.1: the reduced-space chart attempted first, the constrained search in the full joint space reached only where the chart declined. Because the division fell out of the geometry rather than being imposed, both methods were exercised on the same bench, against the same scan, within the same session. For the same reason, they were not exercised on the same queries.

Table 9.4: The two planning methods as exercised on the physical task, each on the subset of transits it was dispatched.

	Reduced-Space Chart	Constrained baseline
Transits Handled	33 of 48	15 of 48
Planning Time	7.7 – 80.5 ms	457 – 7055 ms
Maximum Planned Tilt	0.00035°	33.0°

The separation is of two orders of magnitude in planning time and of five in tilt, and the two distributions do not overlap in either quantity. On a common set of queries the factor is about twenty (Section 9.4.3). The slowest plan the chart produced was faster than the fastest the baseline produced, by a factor of six.

These figures must be read with the division in mind: the baseline's fifteen transits are those the chart declined, and the table alone does not establish how the baseline would perform on the transits the chart handled. The common benchmark of Section 9.4.3 answers that question.

The difference in tilt is not a difference of degree. The chart's figure is not an achievement of the planner but a property of the space it searches: every point of that space maps to a configuration at which the tube is vertical, so no other configuration can be produced. The values recorded here are the residue of the numerical conversion between chart coordinates and joint values, and they are four orders of magnitude below the tolerance the task requires. They are equally far below anything the arm could physically realise, since the backlash in its joints alone exceeds them by several orders of magnitude. The figure should be read as establishing that the planned trajectory lies on the constraint manifold at every waypoint, not as a statement of the tube's physical orientation. The baseline's figures, by contrast, are the outcome of a tolerance the planner was asked to respect and of a verification performed afterwards, and in two instances — 17.7° at the tight tolerance, 33.0° at the loose one — the result exceeded what that tolerance was understood to permit.

The variability differs as sharply as the magnitude. The chart's planning time varied by a factor of ten across thirty-three transits and its tilt by less than a factor of two; the baseline's planning time varied by a factor of fifteen on transits to the same two tubes. That is characteristic of the two mechanisms: enumeration of a finite solution set against a randomised search that terminates when it happens to succeed.

9.4.2 Where the Time is Spent

The reduction in planning time does not reduce the duration of a transfer in proportion, and the decomposition shows why.

Table 9.5: Decomposition of the time taken by a transfer.

Stage	Duration
Scan of the workspace, once per session	approximately 20 s
Grasp inverse kinematics, per transit	755 ms
Standoff selection, per transit	0.83 – 12.2 s
Transit planning, chart	7.7 – 80.5 ms
Transit planning, baseline	0.46 – 7.1 s
Arrival check, per transit	0.06 – 13.3 s
Complete transfer	55 – 95 s

Where the chart handles a transit, planning accounts for a few tens of milliseconds of a transfer lasting a minute and a half. The costs that dominate lie elsewhere: the grasp inverse kinematics, which took 755 ms on every transit without exception, the selection of a descendable standoff, which ranged from 0.83 s where the first candidate was accepted to 12.2 s at tube 2, where all six generators were exhausted at three successive heights before a standoff was found, and the arrival check, which reached 13.3 s in the one trial where it triggered a reconfiguration. The remainder is execution, at the 15 % velocity and acceleration scaling of Section 4.5.2.

This is the outcome Section 7.7.3 anticipates. A component that has been reduced by more than an order of magnitude ceases to be the bottleneck, and the cost moves to wherever no earlier measurement had reason to look. At tube 2, the standoff search alone took roughly five hundred times as long as the transit planning it preceded.

It follows that the improvement demonstrated in Section 9.4.1 is not an improvement in throughput of the same magnitude. What it changes is the character of the guarantee and the cost of the planning step

itself; the duration of a transfer is set by the arm's motion and by the inverse-kinematics calls that precede it.

9.4.3 A Common Benchmark

The comparison of Section 9.4.1 was made on different queries, since each method received only the transits the dispatch ladder assigned to it. To remove that confound, both planners were run on a common set of queries with repetitions. The command sequence of the sixteen trials was replayed with simulated controllers, against the scene of the trials, with the objects at the positions reported in Table 9.2. At every transit, before it was executed, each planner was invoked ten times in planning-only mode from the same start configuration to the same target, with the settings used in operation. This gave 48 queries and 480 attempts per method. The queries are therefore of the same kind as those of the physical trials but not identical to them, since the start configurations arise from the simulated execution. The planning time of the chart includes goal generation but not the bridge onto the manifold of Section 7.6.2. The benchmark was run on a faster computer than the trials, so its absolute times are not comparable with those of Table 9.4; only the relation between the two methods is.

Table 9.6: The two planning methods on a common set of queries, ten repetitions per method and query.

Query Subset	Method	Success	Median [IQR] (ms)	Range (ms)	Max Planned Tilt
Chart Represents (33)	Chart	330/330	44.0 [49.7]	9.2 – 254	0.00057°
	Baseline	330/330	923 [1219]	48.7 – 8479	21.4°
Chart Declines (15)	Chart	0/150	-	-	-
	Baseline	150/150	616 [692]	80 - 4127	21.0°

On the 33 queries the chart could represent, both methods succeeded in every attempt. The chart planned in a median of 44 ms and the baseline in a median of 923 ms, a factor of about twenty. The distributions of planning time overlap at their extremes: the fastest plans of the baseline took 49 ms, and the slowest plans of the chart 254 ms. The planned tilt did not overlap. The chart never exceeded 0.00057°, and the baseline never fell below 0.01°, reaching 21.4° at worst. Every plan of the baseline was found on the first rung of the ladder, and in 20 of the 480 attempts it exceeded the tolerance of 0.30 rad that rung imposes, for the reason given in Section 7.2.3.

On the remaining 15 queries the chart declined. Fourteen of them are the transits to the two nearest tubes, declined during goal generation in about 1.5 ms. The fifteenth is the final return to tube 5, where the search ran for about 4 s without a result. On these queries the baseline planned in a median of 616 ms, faster than on the queries the chart could represent. The hypothesis of Section 9.4.1 is therefore not supported: the queries the chart declines are not harder for the baseline, and the difference in planning time observed on the physical arm is attributable to the method rather than to the division of queries.

9.5 Failures, Anomalies and Sources of Error

Fourteen of the sixteen transfers completed. This section examines the two that did not, three individual trials in which something unexpected occurred without the transfer failing, and the physical characteristics of the setup that underlie them.

9.5.1 The Placement Failures

Both failures took the same form. The tube was carried to the mixing vessel, its contents delivered correctly, and it was returned towards its support base and released a few millimetres outside it — in the case observed directly, five to six millimetres.

Both occurred at tubes 1 and 2, and both on the return transit, which at those tubes is planned by the constrained baseline. The nine transfers at tubes 3, 4 and 5, where the chart handled every transit, completed without exception. The sample is too small to express this as a rate, and no single cause accounts for it; the contributing factors are set out in Section 9.5.3.

9.5.2 Three Anomalies

Three trials departed from the ordinary pattern without failing, and each illustrates a mechanism described in Chapter 7.

1. **Contact with an unmodelled object:** In trial 7 the carried tube touched the funnel seated in the mixing vessel. Nothing was spilled or displaced, and the transfer completed. The planner had reported the path collision-free and was correct to do so: the collision scene represents the mixing vessel as a cylinder of 40 mm radius with its rim at a height of 0.1 m, and the tube was commanded to a position 80 mm above that rim. The funnel occupies part of that clearance and is not present in the model.

This is the situation Section 7.8.3 describes in which the planner confidently and correctly reports that it has avoided an obstacle that is not where the model places it. The consequence here was benign, since the object struck was a light funnel and the striking body a glass tube. The same omission would permit a contact between the arm and the funnel, which would not be.

2. **In-place reconfiguration:** In trial 15 the arm arrived at the standoff above tube 5 in a configuration 2.15 rad from the one intended. The arrival check of Section 7.7.3 planned the descent from where the arm actually was, found that it would reach only 98.7 % of the commanded distance, and reconfigured in place to the verified standoff configuration before descending. The descent then completed in full.

The mechanism converted what would otherwise have been a stalled descent into a successful transfer, at a cost of 13.3 s. It is also the only instance in the sixteen trials, which bears on the threshold it uses: arrival deviations of 2.48, 2.84 and 3.14 rad were recorded in other trials without any reconfiguration being required, because the descent from those configurations was found to be executable. The deviation alone does not predict a stall, and the decision to plan the descent rather than to threshold the deviation, as Section 7.7.3 argues, is vindicated by that.

3. **A wrist-branch mismatch and the loose tolerance:** In trial 16 the transit to the mixing vessel began from a configuration lying on the opposite wrist branch from the planned start, 3.14 rad away, with the fifth and seventh joints each a full half-turn from their planned values. The chart correctly declined the query, this being the limitation of Section 7.11.1. The constrained baseline then failed at the tight tolerance, exhausting its ten-second budget without finding a plan, and the dispatch ladder descended to the loose rung, which produced a plan in 589 ms with a maximum planned tool tilt of 33.0°.

The tube was carried to the mixing vessel at that inclination, visibly tipped, and did not spill. It was the only occasion on which the constrained planner descended below its first rung.

This warrants attention beyond the individual trial. Section 7.2.2 admits the loose tolerance of 0.50 rad on the reasoning that a plan tipping the tube by up to 28.6° remains well inside the hard limit of Section 4.6.1. The trial shows that the rung can be reached in ordinary operation. It also shows that the tolerance the rung enforces bounds the planner's own measure rather than the tilt of the tube, which can exceed it for the reason given in Section 7.2.3. The planned tilt of 33.0° nonetheless lies within the hard limit of 40.1° and some 47° short of the static spill angle, consistent with the absence of spillage. The rung is a last resort that prevents a failure, and it is the hard limit, not the rung's tolerance, that ensures the tube is never tipped further than its contents can bear.

9.5.3 Physical Sources of Error

Three characteristics of the physical setup contributed to the placement failures, and none is visible to the planning system.

1. **The mechanical condition of the manipulator.** The arm is an educational unit that has seen extended use, and several of its joints exhibit backlash. The gripper consequently does not hang exactly vertical even when the commanded configuration is exactly upright, and the deviation is

not repeatable in a way that could be calibrated out. Every figure reported by the planner is a property of the commanded configuration; the relation between that configuration and the arm's physical pose is bounded by the condition of the joints, and the system has no means of observing the difference, since as Section 5.5.2 records the joint state published to the planning layer is the commanded position rather than a measured one.

- 2. Displacement of the support base during grasping.** The gripper occasionally disturbed a tube's support base while closing on it, so that the base was no longer where it had stood when the workspace was scanned. Because the scan of Section 6.6 is performed once per session and its results published as static transforms, the system continued to plan against the original position for the remainder of that session and returned the tube to where the base had been rather than to where it now was.

This is the most consequential of the three, because it accumulates. Each grasp that disturbs a base increases the discrepancy between the scene the planner holds and the bench in front of it, and nothing in the system detects or corrects the drift. It is also the point at which a static scan, entirely adequate for a bench that is arranged and then left alone, becomes inadequate for a bench the robot is itself modifying.

- 3. Residual localisation error.** The comparison of Section 9.2.3 established discrepancies of 4.6 to 17.6 mm between the positions reported by the vision system and those measured by hand, dominated by bias rather than by noise, although part of that discrepancy may belong to the hand measurement rather than to the system (Section 9.2.3).

The three combine rather than acting independently. A localisation error of a few millimetres, a base displaced by a few more, and a gripper hanging slightly off vertical are each within the clearance the chamfered support bases of Section 5.4.2 afford; together they are not. That is why the placement failures occurred where they did, and why they were not reproducible from one trial to the next.

9.5.4 What was not Established

Two observations are recorded without explanation.

The orientation of the marker identifying the mixing vessel was recovered as departing from the shared estimate by 87.6° , although the marker was laid flat alongside the others with its axes aligned to theirs. The cause was not determined, and the marker's reported position should be treated as the least reliable of the seven.

In one trial the arm paused for one to two seconds during a transit and then resumed, moving more quickly than before, without consequence. The behaviour is consistent with the rate-limited hardware channel described in Section 5.5.2, in which a message is forwarded only when the joint values have changed appreciably and the driver subsequently advances against the wall clock, but this was not confirmed.

9.6 Summary

The system performs the task it was built for, under the conditions in which it was tested. Fourteen of the sixteen transfers of the evaluation completed, the two exceptions being placements a few millimetres outside the support base rather than failures of grasping, carrying or pouring; no tube was dropped during the execution of the experiments, no liquid was spilled, no planning attempt was abandoned, and no trial required intervention.

The localisation repeats to better than 0.6 mm from frame to frame within a scan. Its absolute accuracy could be assessed only against positions measured by hand, with which the reported positions agree to within 4.6 to 17.6 mm. The discrepancies are specific to each marker rather than common to all and are of the same order as the uncertainty of the hand measurement itself, estimated at roughly ± 6 mm per coordinate, so that part of them belongs to the ground truth rather than to the system.

The comparison between the planning methods, which Chapter 7 was unable to make against a scene of uncertain fidelity, is made here by the dispatch ladder itself. Across forty-eight transits the reduced-space chart handled thirty-three, planning each in 7.7 - 80.5 ms with a maximum planned tool tilt of 0.00035° ; the constrained baseline handled the remaining fifteen, on the same bench in the same sessions, in 457 - 7055 ms with tilts between 10.6° and 33.0° . The two distributions do not overlap in either quantity. The methods were not applied to the same queries in the trials. On a common benchmark of the same kind of queries, repeated ten times per method, the chart planned about twenty times faster than the baseline (median 44 ms against 923 ms), and the distributions of planned tilt again did not overlap (Section 9.4.3).

The difference in tilt is not a difference of degree but of kind: the chart cannot produce a configuration that violates the constraint, whereas the baseline respects a tolerance and is verified afterwards — and in two instances exceeded what that tolerance was understood to permit.

Which method handles a given transit is determined by the geometry of the target. The chart declined every transit to the two nearest tubes and no transit to the three furthest, on every repetition, because at the nearer positions every available goal configuration lies against a joint stop. That is the limitation stated on structural grounds in Section 7.11.1, observed here as a clean division across five tube positions on one bench.

Two things the evaluation does not establish should be stated alongside. The improvement in planning time does not improve the transfer in proportion: at 15% velocity scaling a transfer takes between 55 and 95 seconds, of which planning by the chart accounts for a few tens of milliseconds, and the dominant costs are the grasp inverse kinematics, the standoff search and the arm's own motion. And the sample is sixteen transfers on one manipulator in one workcell, which supports the descriptions given above but not rates, and not any claim about behaviour on other hardware.

10. Conclusions and Future Work

10.1 Conclusions

This thesis set out to automate a laboratory transfer task with a manipulator that locates its targets rather than replaying fixed coordinates and found that the interesting difficulty lay not in the perception or the mechanism but in a single requirement: that an open tube of liquid remain upright throughout its motion, and not merely at the endpoints.

That requirement confines the admissible configurations to a surface of measure zero within the space a sampling-based planner explores, which is why general-purpose planners address it by rejecting samples or by projecting them numerically, and why both remedies are expensive and approximate. The contribution of this work is to avoid the question. Exploiting the spherical-shoulder, revolute-elbow, spherical-wrist structure of the manipulator, an explicit analytic chart of the constraint surface is constructed, so that the planner searches a five-dimensional parameter domain each point of which maps in closed form to a complete seven-joint configuration. No sample violating the constraint in the nominal kinematic model can be produced, because none exists in the space being searched.

Three architectures were built and compared. Constrained search in the full joint space is the conventional approach and serves as both baseline and fallback. A conservative four-dimensional reduction, in which everything distal of the wrist centre is replaced by a bounding body, is sound in construction and does not work on this manipulator. The cause is a ratio between the extent of the end-effector and the reach of the arm, and the analysis identifies the class of manipulators for which the same technique would serve. The third is the chart.

The evaluation on the physical arm supports the comparison, with one qualification. Across forty-eight transits the chart planned the thirty-three it handled in 7.7 - 80.5 ms with a maximum planned tool tilt of 0.00035° , and the constrained baseline, on the transits the chart declined, in 457 - 7055 ms with tilts between 10.6° and 33.0° . The distributions do not overlap in either quantity. A common benchmark on the same kind of queries, with repetitions, confirmed the difference: about twenty times faster in planning, and no overlap in planned tilt. Fourteen of the sixteen complete transfers of the evaluation succeeded, the two exceptions being placements a few millimetres outside the support base. The figure shows that the task can be performed with this system; it does not establish how reliably.

Three findings are worth separating from the headline comparison, because each was established rather than assumed.

1. **The difference between the methods is one of kind, not degree.** The chart's tilt figure is not an achievement of the planner but a property of the space it searches, and it cannot be improved by a better implementation or degraded by a worse one. The baseline's figure is the outcome of a tolerance that must be verified afterwards, and on two occasions the verification found more tilt than the tolerance was understood to permit.
2. **The reduction's benefit is bounded by where the time actually goes.** Having reduced transit planning by more than an order of magnitude, the cost moved to the grasp inverse kinematics, the standoff search and the arm's own motion, so that a transfer still takes between 55 and 95 seconds. An improvement in one component of a system improves the system only in the proportion that component contributed.
3. **And the method's applicability depends on the layout of the cell.** The chart declined every transit to the two nearest tubes and no transit to the three furthest, on every repetition, because at the nearer positions every available goal configuration on the wrist branch the arm occupied lies against a joint stop. This was predicted on structural grounds and observed as a clean division across five positions on one bench. A planner built on an exact analytic chart commits the manipulator to a single wrist branch, and the workcell must be arranged compatibly with that commitment.

The system as a whole demonstrates that a laboratory assistant of this kind can be assembled from a low-cost educational arm, a commodity webcam, printed fixtures and open-source software, and operated through an interface requiring no knowledge of any of the algorithms beneath it. That is the accessibility argument of Section 1.4, made concrete.

10.2 Limitations

The system described in this thesis is a proof of concept, and several of its properties follow from that status rather than from any difficulty of principle. They are set out here in six groups, since the remedies differ in kind.

10.2.1 The Planning Method

The reduction developed in Chapter 7 rests on three assumptions, each of which bounds where the method applies.

The chart holds the wrist branch fixed for the duration of a plan. This is what makes the mapping from reduced coordinates to joint values continuous, and it is also the method's principal restriction: a motion whose start and goal lie on opposite branches of the ZYZ inversion cannot be represented within a single chart, and must be delegated to a general constrained planner. The two branches meet at the wrist singularity identified in Section 3.6.2, so the restriction is geometric rather than incidental.

This restriction was observed directly. Across the sixteen transfers of Chapter 9 the chart declined every transit to the two nearest tube positions and no transit to the three furthest, on every repetition, because at the nearer positions every available goal configuration on the wrist branch the arm occupied lay against a joint stop. A planner built on an exact analytic chart therefore constrains the layout of the workcell, and the arrangement of the stations must be chosen with that in mind.

The method assumes that the kinematic decoupling holds exactly. It does so on this manipulator by construction, as Section 7.3 establishes to machine precision, but a wrist whose axes only approximately intersect would introduce a position error in the recovered configuration growing with the offset. The technique is therefore specific to mechanisms satisfying Pieper's condition exactly, and not merely to a good approximation.

The reduction applies to constraints leaving a parameterisable residual freedom. An upright axis leaves a single rotation and is consequently close to the most favourable case; a constraint leaving a two-dimensional residual set would enlarge the chart correspondingly, and one leaving an irregular set would not admit this treatment at all.

A fourth limitation concerns the alternative approach of Section 7.4. Its failure is not a property of the construction but of the proportions of this particular manipulator, and the result should not be read as a general verdict on conservative surrogates. On an arm whose reach is large relative to its end-effector, the same technique would be serviceable.

10.2.2 Perception

Perception is used for localisation only: the camera establishes the positions of the objects before the transfers begin and is not consulted during motion, so the system does not perform visual servoing. The accuracy with which the system locates the objects it manipulates was measured in Section 9.2 at between 4.6 and 17.6 mm, against a frame-to-frame scatter below 0.6 mm. The error is therefore systematic rather than random. It is not, however, common to all markers, and it is of the same order as the uncertainty of the hand measurement against which it was assessed, roughly ± 6 mm per coordinate (Section 9.2.3), so the absolute accuracy of the system is known only to within that uncertainty. Repeatability across scans was not measured. The registration uses a single reference marker rather than a multi-pose hand-eye calibration, which is the simpler procedure but forgoes the averaging that would reduce the influence of any one observation.

The scene is established once. The scanning procedure of Section 6.6 solves the positions of all the markers and publishes them as static transforms, which hold until the next scan. This suits a bench that

is arranged and then left alone, and it is adequate for the task as performed: the objects do not move of their own accord.

They do, however, move. Section 9.5.3 records that the gripper occasionally displaced a tube's support base while closing on it, after which the system continued to plan against the position established at the scan. The limitation is therefore sharper than it first appears. It is not that the scene may drift, but that the robot itself modifies the scene it is planning against, and that the resulting discrepancy accumulates across a session with nothing to detect or correct it.

A marker that cannot be solved during the scan is reported as such and is not published, and the planner then places the corresponding object at a nominal position (Section 7.8.1) rather than at an unreliable one.

10.2.3 Control and Feedback

The most consequential limitation of the system is the absence of feedback at two distinct levels.

Between the planning layer and the arm, the joint state published to MoveIt is the commanded position rather than a value read from the encoders, as described in Section 5.5.2. The joint servos close their own position loops internally, so the arm does reach the configurations it is commanded to; but the planning layer receives no measurement of what was executed, and cannot therefore detect a deviation, a stall, or an obstruction encountered during motion.

The consequence is visible in the results. Section 9.5.3 records that the manipulator's joints exhibit backlash, so the gripper does not hang exactly vertical even when the commanded configuration is exactly upright. Every tilt figure reported in this thesis is a property of the commanded configuration. The relation between that and the arm's physical pose is bounded by the mechanical condition of the joints, and the planning layer has no means of observing the difference.

Between the controller and the operator, the controller reports the outcome of every command, and a sequence stops at the first failure (Section 8.2.1). The outcome is written only to the interface's terminal log, however, and not displayed in the window, and commands issued from the override panel do not wait for it. There is no means of cancelling a procedure once it has begun, other than closing the interface.

Neither limitation prevents the system from performing its task under supervision. Both prevent it from being operated unattended, since an unattended system must be able to detect that something has gone wrong and to stop.

10.2.4 The Loose Tolerance of the Fallback Ladder

Where the reduced planner declines a transit and the constrained baseline fails at the tight tolerance, the dispatch ladder of Section 7.2.2 relaxes the constraint to 0.50 rad. The rung was justified on the reasoning that a plan tipping the tube by up to 28.6° remains well inside the hard limit of Section 4.6.1, and that a plan of that kind is preferable to no plan at all.

Section 9.5.2 records the one occasion on which the rung was reached. The resulting trajectory tilted the carried tube by a planned 33.0° , exceeding even the relaxed tolerance, and was visibly alarming to the operator although nothing was spilled. Three things follow. The rung is reachable in ordinary operation rather than only in contrived circumstances. The tolerance it enforces is a bound on the planner's own measure and not on the tilt of the tube. Safety therefore rests on the hard limit rather than on the rung.

That limit held: the planned tilt lay 7° inside it and some 47° short of the static spill angle, so the observation does not show the rung to be unsafe. What it shows is different. A motion can satisfy the criterion of Section 4.6 and still be unacceptable to an operator watching an open tube tip visibly. The limit and the margin behind it also rest on the model of the liquid in Section 4.6.2 rather than on measurement. The occasion itself arose from a wrist-branch mismatch, which the atlas of Section 10.3.2 would remove, and with it the need to reach the rung at all.

10.2.5 Scope of the Task

The system transfers the contents of a tube to a mixing vessel by tipping the tube. It does not meter the quantity delivered, and the recipes of Section 8.3.1 specify which tubes are used and in what order but not how much of each. The chemistry is correspondingly limited to procedures in which whole volumes are combined.

No property of the resulting mixture is sensed. The system does not observe the reaction, verify that it occurred, or detect that a tube was empty before it was poured.

10.2.6 Scope of the Evaluation

The results of Chapter 9 were obtained from sixteen individually commanded transfers on one manipulator, in one workcell, with one arrangement of the stations. A recipe was executed successfully in automatic mode but was not instrumented, so the behaviour of a multi-tube sequence, in particular whether the positional error identified in Section 9.5.3 accumulates across consecutive transfers, was not measured.

On the physical arm the planning methods were exercised as the dispatch ladder selected them, so that each faced different queries. A controlled comparison on common queries was conducted afterwards (Section 9.4.3), but in simulation and on a different computer, so it establishes the relation between the two methods rather than their planning times on the arm. No trial was conducted with the orientation constraint removed altogether, so the claim that the constraint is the source of the difficulty rests on the analysis of Section 4.3 rather than on an experimental control.

10.3 Future Work

Each of the limitations above suggests an extension, and they are listed here in the order in which they would most improve the system rather than in the order of the limitations they address.

10.3.1 Closing the Feedback Loops

The extension of greatest value, and the least difficult, builds on the status channel that already exists. The controller reports the outcome of every command; displaying it in the window, together with the stage currently in progress, would let the operator see progress and failures as they occur, and would allow the interface to refuse a command inconsistent with the state the arm is actually in. An abort control that halts the motion in progress, rather than only the next command, follows naturally from the same channel.

Reading the joint positions back from the encoders and publishing those, rather than the commanded values, would close the second loop. This would allow deviations between the planned and the executed trajectory to be detected and would make possible a class of safety response – halting on an unexpected discrepancy – that the present architecture cannot support.

10.3.2 An Atlas of Charts

The restriction that a plan must remain on one wrist branch would be removed by an atlas of two charts joined at the wrist singularity where the branches meet. Motions requiring the passage from one branch to the other could then be planned within the reduced formulation rather than delegated to a general planner, and the method would cover the full workspace of the manipulator. This is the natural continuation of the construction of Chapter 7, and the structure required for it is already established there.

10.3.3 Improved Registration and Perception

Replacing the single reference marker with a proper hand-eye calibration over multiple poses would reduce the influence of any individual observation on the camera-to-base transformation. Distributing several reference markers across the bench and averaging over them would achieve much of the same benefit at lower cost, and would additionally allow a displacement of the camera to be detected, since the markers would cease to agree with one another.

A depth camera or a second viewpoint would remove the dependence on planar fiducials altogether and would permit objects to be located without markers being affixed to them, at a cost in equipment that the affordability objective of Section 1.4 would have to be weighed against.

10.3.4 Maintaining the Scene during Operation

The scanning procedure establishes the positions of the objects once and holds them for the session, which is adequate for a bench that is arranged and then left alone. Section 9.5.3 records that this assumption fails in a specific way: the gripper occasionally displaces a support base while closing on a tube, so that the object the system will later return the tube to is no longer where the scan found it.

The straightforward remedy is to rescan between transfers, at a cost of some seconds per transfer and with the arm withdrawn so that it does not occlude the markers. A less costly alternative is to verify rather than re-solve comparing the observed position of a single marker against its stored value would establish whether the scene has moved without repeating the whole procedure and would allow a full rescan to be triggered only when it has.

Either would convert a discrepancy that at present accumulates silently into one the system detects. This is the extension with the clearest evidence behind it, since the failure it addresses was observed rather than anticipated. A further step would be to close the loop during motion itself, observing the gripper and the tube as the arm approaches a support base and correcting the final placement visually, which would address the placement failures of Section 9.5.1 directly.

10.3.5 Volumetric Control

The addition of a syringe pump, or of a balance beneath the mixing vessel providing gravimetric feedback, would allow the quantity delivered to be specified and verified. This would extend the recipe format of Section 8.3.1 from an ordered list of tubes to a specification of quantities and would bring the system within reach of protocols requiring stoichiometric proportions rather than whole volumes.

10.3.6 Generalisation of the Method

The reduction developed here applies to a constraint fixing the direction of a single tool axis on a manipulator of S-R-S structure. Two directions of generalisation present themselves. The first is to other constraints of the same character, a tool required to remain within a cone, or to maintain contact with a surface, each of which leaves a residual freedom of a different dimension and would require a chart of corresponding size. The second is to other mechanisms satisfying Pieper's condition, of which the S-R-S arm is one family among several, and for which analogous closed-form charts should be constructible by the same reasoning.

10.4 Closing

The question this thesis began with was whether laboratory automation of a useful kind can be built from equipment a research group already has or can readily obtain. The answer developed over the preceding chapters is that it can, but that doing so exposes a difficulty which more expensive equipment conceals rather than solves.

That difficulty is the constraint. A tube of liquid must be kept upright throughout its motion, and this apparently modest requirement changes the character of the planning problem entirely, because the configurations satisfying it form a vanishingly small part of the space a planner searches. The response developed here was not to search harder but to search elsewhere: to construct, from the kinematic

structure of the particular arm, a space in which the requirement cannot be violated because no configuration violating it exists. Within that space, and within the model of the arm on which it is built, the constraint is not enforced, verified or approximated. It is simply true.

The result is a method that is exact where the general technique is approximate, that planned about twenty times faster on common queries, and that is applicable only where a manipulator has the structure it depends upon and a workcell is arranged compatibly with the commitment it makes. Both halves of that sentence were established experimentally.

What the system demonstrates is narrower than a laboratory assistant and broader than a planning method. It is that a task of this kind can be performed, by equipment of this cost, with guarantees of this character, and that the guarantees are what matter, since a transfer that is almost upright is a transfer that has spilled.

11. References

- [1] R. Bogue, “Robots in the laboratory: a review of applications,” *Industrial Robot: An International Journal*, vol. 39, no. 2, pp. 113–119, Mar. 2012, doi: 10.1108/01439911211203382.
- [2] N. J. Knight, S. Kanza, D. Cruickshank, W. S. Brocklesby, and J. G. Frey, “Talk2Lab: The Smart Lab of the Future,” *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8631–8640, Sep. 2020, doi: 10.1109/JIOT.2020.2995323.
- [3] T. Habich and S. Beutel, “Digitalization concepts in academic bioprocess development,” *Eng. Life Sci.*, vol. 24, no. 4, Apr. 2024, doi: 10.1002/elsc.202300238.
- [4] H. Lasi, P. Fettke, H. G. Kemper, T. Feld, and M. Hoffmann, “Industry 4.0,” *Business and Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, Aug. 2014, doi: 10.1007/s12599-014-0334-4.
- [5] G. Tom *et al.*, “Self-Driving Laboratories for Chemistry and Materials Science,” *Chem. Rev.*, vol. 124, no. 16, pp. 9633–9732, Aug. 2024, doi: 10.1021/acs.chemrev.4c00055.
- [6] B. Burger *et al.*, “A mobile robotic chemist,” *Nature*, vol. 583, no. 7815, pp. 237–241, Jul. 2020, doi: 10.1038/s41586-020-2442-2.
- [7] J. Bai, L. Cao, S. Mosbach, J. Akroyd, A. A. Lapkin, and M. Kraft, “From Platform to Knowledge Graph: Evolution of Laboratory Automation,” *JACS Au*, vol. 2, no. 2, pp. 292–309, Feb. 2022, doi: 10.1021/jacsau.1c00438.
- [8] I. Holland and J. A. Davies, “Automation in the Life Science Research Laboratory,” *Front. Bioeng. Biotechnol.*, vol. 8, Nov. 2020, doi: 10.3389/fbioe.2020.571777.
- [9] S. Hutchinson, G. D. Hager, and P. I. Corke, “A tutorial on visual servo control,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996, doi: 10.1109/70.538972.
- [10] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild,” *Sci. Robot.*, vol. 7, no. 66, May 2022, doi: 10.1126/scirobotics.abm6074.
- [11] D. Coleman, I. Sucan, S. Chitta, and N. Correll, “Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study,” Apr. 2014, [Online]. Available: <http://arxiv.org/abs/1404.3785>
- [12] I. A. Sucan, M. Moll, and L. E. Kavraki, “The Open Motion Planning Library,” *IEEE Robot. Autom. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012, doi: 10.1109/MRA.2012.2205651.
- [13] J. J. Kuffner and S. M. LaValle, “RRT-connect: An efficient approach to single-query path planning,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE*

International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), IEEE, pp. 995–1001. doi: 10.1109/ROBOT.2000.844730.

- [14] J. Pan, S. Chitta, and D. Manocha, “FCL: A general purpose library for collision and proximity queries,” in *2012 IEEE International Conference on Robotics and Automation*, IEEE, May 2012, pp. 3859–3866. doi: 10.1109/ICRA.2012.6225337.
- [15] H. R. Kam, S. H. Lee, T. Park, and C. H. Kim, “RViz: a toolkit for real domain data visualization,” *Telecommun. Syst.*, vol. 60, no. 2, pp. 337–345, Oct. 2015, doi: 10.1007/s11235-015-0034-5.
- [16] A. Kaehler and G. Bradski, *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. Sebastopol, CA: O’Reilly Media, 2016.
- [17] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, Jun. 2014, doi: 10.1016/j.patcog.2014.01.005.
- [18] Elephant Robotics, “myArm 300 Pi - 7 DOF Robot Arm,” Elephant Robotics Official Site. Accessed: Dec. 15, 2025. [Online]. Available: <https://docs.elephantrobotics.com/docs/myarm-pi-300-en/>
- [19] Ángel L.M., “Thor: Open Source 3D Printable 6DOF Robotic Arm.” Accessed: Dec. 15, 2025. [Online]. Available: <http://thor.angel-lm.com/>
- [20] Creative, “Creative Live! Cam Sync 1080p V2.” Accessed: Aug. 16, 2026. [Online]. Available: <https://gr.creative.com/p/webcams/creative-live-cam-sync-1080p-v2#feature-section-1>
- [21] C. Liu, Y. Chen, and F. Mo, “Transforming organic chemistry research paradigms: moving from manual efforts to the intersection of automation and artificial intelligence,” Nov. 2023.
- [22] C. Crandell, “Continuous-flow analysis: the Auto-Analyzer,” *J. Anal. Methods Chem.*, vol. 7, no. 3, pp. 145–148, Jan. 1985, doi: 10.1155/S146392468500030X.
- [23] J. N. Little, “The Zymate Laboratory Automation Systems,” *J. Liq. Chromatogr.*, vol. 9, no. 14, pp. 3197–3201, Oct. 1986, doi: 10.1080/01483918608074177.
- [24] D. A. Rudge, “The automation of solution phase synthetic chemistry using XP Zymate™ laboratory robotic systems,” *Laboratory Automation & Information Management*, vol. 33, no. 2, pp. 81–86, Dec. 1997, doi: 10.1016/S1381-141X(97)80003-2.
- [25] R. D. King, “‘Robot scientists: From Adam to Eve to Genesis,’ in *Artificial Intelligence in Science: Challenges, Opportunities and the Future of Research*, Paris, France: OECD Publishing, 2023.” Paris, France.
- [26] D. A. Pereira and J. A. Williams, “Origin and evolution of high throughput screening,” *Br. J. Pharmacol.*, vol. 152, no. 1, pp. 53–61, Sep. 2007, doi: 10.1038/sj.bjp.0707373.

- [27] G. Gauglitz, “Lab 4.0: SiLA or OPC UA,” *Anal. Bioanal. Chem.*, vol. 410, no. 21, pp. 5093–5094, Aug. 2018, doi: 10.1007/s00216-018-1192-6.
- [28] D. Adam, “The automated lab of tomorrow,” *Proceedings of the National Academy of Sciences*, vol. 121, no. 17, Apr. 2024, doi: 10.1073/pnas.2406320121.
- [29] P. Bose, “How Cloud Labs and Remote Research Shape Science Remote research via cloud labs makes it possible for scientists to conduct complex experiments from a distance.”
- [30] M. T. Wayland and M. Landgraf, “A Cartesian Coordinate Robot for Dispensing Fruit Fly Food,” *Journal of Open Hardware*, vol. 2, no. 1, p. 3, Jul. 2018, doi: 10.5334/joh.9.
- [31] R. MacArron *et al.*, “Impact of high-throughput screening in biomedical research,” Mar. 2011. doi: 10.1038/nrd3368.
- [32] H. Makino, “Development of the SCARA,” 2014, *Fuji Technology Press*. doi: 10.20965/jrm.2014.p0005.
- [33] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-32552-1.
- [34] S. Patil, V. Vasu, and K. V. S. Srinadh, “Advances and perspectives in collaborative robotics: a review of key technologies and emerging trends,” Dec. 01, 2023, *Springer Nature*. doi: 10.1007/s44245-023-00021-8.
- [35] J. J. Craig, *Introduction to robotics: mechanics and control*. Pearson, 2018.
- [36] Hamilton Company, “Microlab STAR Automated Liquid Handler,” Hamilton Company Official Website. Accessed: Mar. 08, 2026. [Online]. Available: <https://www.hamiltoncompany.com/microlab-star?part-number=Microlab+STAR>
- [37] EPSON Company, “Epson SCARA GX1-C 175mm Series.” Accessed: Mar. 08, 2026. [Online]. Available: https://www.epson.gr/el_GR/%CE%A0%CF%81%CE%BF%CF%8A%CF%8C%CE%BD%CF%84%CE%B1/%CE%A1%CE%BF%CE%BC%CF%80%CF%8C%CF%84/%CE%A1%CE%BF%CE%BC%CF%80%CE%BF%CF%84%CE%B9%CE%BA%CE%BF%CE%AF-%CE%BC%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CF%83%CE%BC%CE%BF%CE%AF-scara/epson-scara-gx1-c-175mm-series/p/54828
- [38] KUKA Company, “KUKA KR AGILUS,” KUKA Robotics. Accessed: Mar. 08, 2026. [Online]. Available: <https://www.kuka.com/en-de/products/robot-systems/industrial-robots/kr-agilus>
- [39] M. Christensen *et al.*, “Automation isn’t automatic,” *Chem. Sci.*, vol. 12, no. 47, pp. 15473–15490, 2021, doi: 10.1039/D1SC04588A.
- [40] A. Angelopoulos, M. Verber, C. McKinney, J. Cahoon, and R. Alterovitz, “High-Accuracy Injection Using a Mobile Manipulation Robot for Chemistry Lab

- Automation,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2023, pp. 10102–10109. doi: 10.1109/IROS55552.2023.10341743.
- [41] H. Fakhruideen, G. Pizzuto, J. Glowacki, and A. I. Cooper, “ARChemist: Autonomous Robotic Chemistry System Architecture,” in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, May 2022, pp. 6013–6019. doi: 10.1109/ICRA46639.2022.9811996.
 - [42] F. Romano, S. Milani, and C. M. Joppolo, “Airborne particle and microbiological human emission rate investigation for cleanroom clothing combinations,” *Build. Environ.*, vol. 180, p. 106967, Aug. 2020, doi: 10.1016/j.buildenv.2020.106967.
 - [43] J. Ding, K. W. Tan, and X. Chen, “Do-it-yourself instrument integration into an existing mammalian cell line development automation platform,” *SLAS Technol.*, vol. 31, p. 100246, Apr. 2025, doi: 10.1016/j.slast.2025.100246.
 - [44] Z. J. Baum, X. Yu, P. Y. Ayala, Y. Zhao, S. P. Watkins, and Q. Zhou, “Artificial Intelligence in Chemistry: Current Trends and Future Directions,” *J. Chem. Inf. Model.*, vol. 61, no. 7, pp. 3197–3212, Jul. 2021, doi: 10.1021/acs.jcim.1c00619.
 - [45] R. Szeliski, *Computer vision: algorithms and applications*. Springer, 2022.
 - [46] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, “Fiducial Markers for Pose Estimation,” *J. Intell. Robot. Syst.*, vol. 101, no. 4, p. 71, Apr. 2021, doi: 10.1007/s10846-020-01307-9.
 - [47] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000, doi: 10.1109/34.888718.
 - [48] M. Quigley *et al.*, “ROS: an open-source Robot Operating System.” [Online]. Available: <http://stair.stanford.edu>
 - [49] J. M. O’Kane, *A gentle introduction to ROS*. Jason M. O’Kane, 2014.
 - [50] A. Koubaa, Ed., *Robot Operating System (ROS)*, vol. 625. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-26054-9.
 - [51] M. Quigley, B. Gerkey, and W. D. Smart, *Programming robots with ROS*. O’Reilly Media, 2015.
 - [52] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, IEEE, pp. 2149–2154. doi: 10.1109/IROS.2004.1389727.
 - [53] Z. Kingston, M. Moll, and L. E. Kavraki, “Sampling-Based Methods for Motion Planning with Constraints,” *Annu. Rev. Control Robot. Auton. Syst.*, vol. 1, no. 1, pp. 159–185, May 2018, doi: 10.1146/annurev-control-060117-105226.

- [54] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *2009 IEEE International Conference on Robotics and Automation*, IEEE, May 2009, pp. 625–632. doi: 10.1109/ROBOT.2009.5152399.
- [55] M. Stilman, "Global Manipulation Planning in Robot Joint Space With Task Constraints," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 576–584, Jun. 2010, doi: 10.1109/TRO.2010.2044949.
- [56] L. Jaillet and J. M. Porta, "Path Planning Under Kinematic Constraints by Rapidly Exploring Manifolds," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 105–117, Feb. 2013, doi: 10.1109/TRO.2012.2222272.
- [57] Z. Kingston, M. Moll, and L. E. Kavraki, "Exploring implicit spaces for constrained sampling-based planning," *Int. J. Rob. Res.*, vol. 38, no. 10–11, pp. 1151–1178, Sep. 2019, doi: 10.1177/0278364919868530.
- [58] D. L. Pieper, "The Kinematics of Manipulators Under Computer Control," Stanford University, 1968.
- [59] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki, and K. Kosuge, "Analytical Inverse Kinematic Computation for 7-DOF Redundant Manipulators With Joint Limits and Its Application to Redundancy Resolution," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1131–1142, Oct. 2008, doi: 10.1109/TRO.2008.2003266.
- [60] P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, IEEE, Nov. 2015, pp. 928–935. doi: 10.1109/HUMANOIDS.2015.7363472.
- [61] T. Yoshikawa, "Manipulability of Robotic Mechanisms," *Int. J. Rob. Res.*, vol. 4, no. 2, pp. 3–9, Jun. 1985, doi: 10.1177/027836498500400201.
- [62] T. Foote, "tf: The transform library," in *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, IEEE, Apr. 2013, pp. 1–6. doi: 10.1109/TePRA.2013.6556373.
- [63] R. A. Ibrahim and V. N. Pilipchuk, "RECENT ADVANCES IN LIQUID SLOSHING DYNAMICS *."
- [64] M. Gornier, R. Haschke, H. Ritter, and J. Zhang, "MoveIt! Task Constructor for Task-Level Motion Planning," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, May 2019, pp. 190–196. doi: 10.1109/ICRA.2019.8793898.
- [65] R. Hartley and A. Zisserman, *Multiple view geometry in Computer Vision*. Cambridge University Press, 2004.
- [66] D. C. Brown, "Decentering distortion of lenses," *Photogrammetric Engineering*, pp. 444–462, 1966.
- [67] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Pearson, 2018.

- [68] S. Suzuki and K. Abe, “Topological structural analysis of digitized binary images by border following,” *Comput. Vis. Graph. Image Process.*, vol. 30, no. 1, pp. 32–46, Apr. 1985, doi: 10.1016/0734-189X(85)90016-7.
- [69] V. Lepetit, F. Moreno-Noguer, and P. Fua, “EPnP: An Accurate $O(n)$ Solution to the PnP Problem,” *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, Feb. 2009, doi: 10.1007/s11263-008-0152-6.
- [70] T. Kunz and M. Stilman, “Time-Optimal Trajectory Generation for Path Following with Bounded Acceleration and Velocity.” [Online]. Available: <http://www.golems.org/node/1570>.
- [71] A. D. Moore, *Python GUI Programming with Tkinter: Develop Responsive and Powerful GUI Applications with Tkinter*. Packt Publishing, 2021.

12. Appendix

Source Code

The complete source of the system described in this thesis is available at <https://github.com/vkaratrantos/thesis-repo/tree/main> .

The principal files are the following:

File	Language	Described in
<code>simple_move.cpp</code>	C++	Chapter 7
<code>aruco_world_coords.py</code>	Python	Chapter 6
<code>calibrate_camera.py</code>	Python	Section 6.2
<code>moveit_bridge_7.py</code>	Python	Section 5.5.2
<code>ros_lan_driver.py</code>	Python	Section 5.5.2
<code>gui.py</code>	Python	Chapter 8
<code>wrist_blob_gen.py</code>	Python	Section 7.4

The repository also contains the robot description, the MoveIt configuration, the reduced model generated for the swept-volume planner of Section 7.4, and the camera parameters of Section 6.2.3.