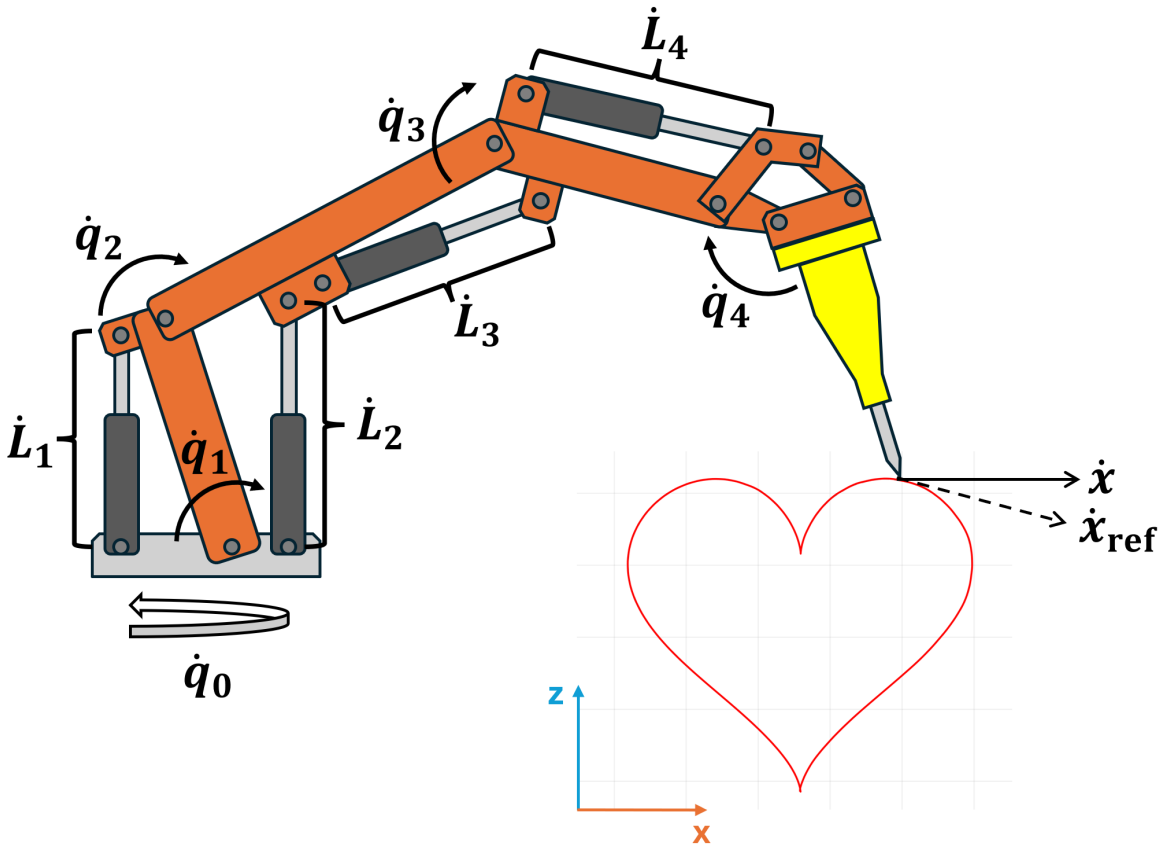
CHALMERS
UNIVERSITY OF TECHNOLOGY



Inverse kinematic optimization and MPC on a four-link hydraulic manipulator

A motion control algorithm for human-operated manipulators

Master's thesis in Systems, Control and Mechatronics

Konrad Billing
Filip Djäknegren

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Inverse kinematic optimization and MPC on a four-link hydraulic manipulator

A motion control algorithm for human-operated manipulators

Konrad Billing
Filip Djäknegren

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical engineering
Division of Systems and Control
Chalmers University of Technology
Gothenburg, Sweden 2026

Inverse kinematic optimization and MPC on a hydraulic demolition robot
A motion control algorithm for human-operated manipulators
Filip Djäknegren
Konrad Billing

Supervisor: Johan Berg, Husqvarna
Examiner: Nikolce Murgovski, Department of Electrical engineering

Master's Thesis 2026
Department of Electrical engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Inverse kinematic optimization and MPC on a hydraulic demolition robot
A motion control algorithm for human-operated manipulators
Konrad Billing
Filip Djäknegren
Department of Electrical engineering
Chalmers University of Technology

Abstract

Operating individual joints on a hydraulic manipulator can be tedious and unproductive with limited experience. Therefore, this thesis presents a method to ease operator strain and lessen experience requirements by enabling cartesian control of the robot tool. This is achieved with differential closed-loop inverse kinematics and linear model predictive motion control. References are generated with a pseudo-inverse Jacobian algorithm formulated as a quadratic program with a secondary null-space objective to exploit redundant joints. A sensitivity analysis of a non-linear hydraulic model resulted in a linear first-order model which was used to design an MPC that tracks the references. Results from a simulated robot show robust tracking performance even when the robot is exposed to noise and unmodeled disturbances. The designed null-space objective manages to control null-space motion and the CLIK is shown to reject unmodeled disturbances.

Keywords: Inverse kinematics, motion control, model predictive control, quadratic programming, trajectory planning, hydraulic.

Acknowledgements

We want to thank our supervisors at Husqvarna, Johan Berg and Jacob Larsson for their support and guidance during this thesis. We also want to thank our examiner Nikolce Murgovski for all the expert advice and feedback we have gotten. Lastly we want to thank all Husqvarna employees who have answered our many questions.

Konrad Billing, Gothenburg, June 2026
Filip Djäknegren, Gothenburg, June 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AGV	Automated guided vehicle
C1	Chamber 1, cap end of cylinder
C2	Chamber 2, rod end of cylinder
DH	Denavit-Hartenberg
DOM	Degrees of motion
DOF	Degrees of freedom
EE	End-effector
FABRIK	Forward and backward reaching inverse kinematics
FK	Forward kinematics
LPV	Linear-Parametrically Varying
LQR	Linear quadratic regulator
MPC	Model Predictive Control
NMPC	Non-linear Model Predictive Control
OAT	One-at-a-time
QP	Quadratic Program

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

i	Index corresponding to robot joint, link, coordinate frame or input
j	Index corresponding to robot joint
k	Index corresponding to discrete time step or robot joint
m	Index corresponding to trajectory number in Morris method, Section 2.3.2

Sets

$\mathbb{R}$	Real numbers
$\mathcal{N}$	Null-space

Parameters

h	Sampling interval
m	Number of inputs, Sections 2.4, 3.4
m_i	Mass of unit i
n	Number of joints
n	Number of states, Sections 2.4, 3.4
n_q	Number of DOF
n_x	Number of DOM
N	Prediction horizon length
p	Number of outputs, Chapter 2.4, 3.4

p_i	The i:th parameter in the parameter space, Section 2.3.2
t	Time
β	Hydraulic fluid bulk modulus
c_d	Valve discharge coefficient
c_l	Internal cylinder leakage coefficient
c_s	Internal pump leakage coefficient
c_v	Viscous drag coefficient
ρ	Hydraulic fluid density
a	Offset along X-axis (DH)
α	Rotation around X-axis (DH)
d	Offset along Z-axis (DH)
θ	Rotation around Z-axis (DH)
$\alpha_{a,i}$	Linkage angle corresponding to joint i
$\alpha_{c,i}$	Linkage angle corresponding to joint i
b_i	Linkage length corresponding to joint i
c_i	Linkage length corresponding to joint i
f	Linkage length corresponding to joint 4
g	Linkage length corresponding to joint 4
j	Linkage length corresponding to joint 4
o	Linkage length corresponding to joint 4
A_{C1}	Piston area (cap-end)
A_{C2}	Piston area (rod-end)
A_c	Cylinder piston area
A_r	$A_{\mathrm{C2}}/A_{\mathrm{C1}}$
K_{LS}	Load-sense gain
L_{hose}	Cylinder hose length
L_{max}	Maximum cylinder length
L_{min}	Minimum cylinder length
L_{pump_hose}	Pump hose length
m_p	Piston mass
P_t	Pressure in hydraulic fluid reservoir
c_{dead}	Cylinder hose dead volume
s_{dead}	Pump hose dead volume
τ_{LS}	Load-sense time constant

τ_v	Valve time constant
V_0	Dead volume of cylinder
V_c	Cylinder volume
V_d	Angular volume displacement of pump
V_s	Dead volume of pump
K_i	Integral gain
K_p	Proportional gain
$\mathbf{W}$	Joint weight matrix
Γ	Previous input selector matrix
Λ	Slack variable penalty matrix
P_f	Terminal state penalty matrix
Q	State penalty matrix
R	Control input penalty matrix
S	Input rate of change penalty matrix
Θ	Input difference matrix
A_{cart}	Cartesian position inequality constraint matrix
A_{cyl}	Cylinder position inequality constraint matrix
A_{eq}	Equality constraint matrix
A_{in}	Inequality constraint matrix
$A_{v,cart}$	Cartesian velocity inequality constraint matrix
$A_{v,cyl}$	Cylinder velocity inequality constraint matrix
b_{cart}	Cartesian position inequality constraint vector
b_{cyl}	Cylinder position inequality constraint vector
b_{eq}	Equality constraint vector
b_{in}	Inequality constraint vector
$b_{v,cart}$	Cartesian velocity inequality constraint vector
$b_{v,cyl}$	Cylinder velocity inequality constraint vector
x_{max}	Maximum cartesian position
x_{min}	Minimum cartesian position
$\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$	Discrete system matrices
$\mathcal{A}_c, \mathcal{B}_c$	Continuous system matrices
Δ	Discrete perturbation step in parameter space, Section 2.3.2
I	Identity matrix
I_{nq}	Identity matrix $n \times q$

$\boldsymbol{g}$	Gravity vector
$\mathcal{I}$	Inertia tensor

Variables

Δu	Input rate of change
ϵ	Slack variable
u	Input vector
$\mathbf{u}$	Input decision variables
x	State vector
$\mathbf{x}$	State decision variables
y	Output vector
$\mathbf{z}$	Combined decision variables
ϕ	Auxiliary variable
f	Linear term of cost function or general function
H	Quadratic term of cost function
$\mathcal{V}$	Cost function
$\bar{J}$	Cylinder-space to task-space Jacobian
J	Jacobian
$J^{\dagger}$	Pseudo-inverse Jacobian
J_{cm}	Center of mass Jacobian
J_0^n	Jacobian of EE
J_v	Linear velocity components of Jacobian
J_ω	Angular velocity components of Jacobian
Σ	Singular value matrix
U	Matrix with left singular vectors
V	Matrix with right singular vectors
ξ	Cylinder-space to joint-space Jacobian
$\mathbf{H}_{i-1}^{i}$	Homogeneous transformation matrix of frame i relative to $i-1$
$\mathbf{R}_{i-1}^{i}$	Rotation matrix of frame i relative to $i-1$
$\mathbf{t}_{i-1}^{i}$	Translation vector of frame i relative to $i-1$
$\mathbf{q}$	Joint configuration vector
q	Joint angle
q_i	Angle of joint i

$q_{desired}$	Desired joint configuration
q_{ref}	Reference joint angle
$\dot{q}_*$	Designed joint velocity vector
r_i	Link length between frame i and $i-1$
$\mathbf{r}_i$	Distance vector between frame $i-1$ and point of reference
τ_k	Torque for joint k
τ_i	Torque for joint i
τ_q	Joint torque
z_i	Z-axis vector of frame i
e	Task-space position error
v_i	Cartesian linear velocity of frame i
$\dot{x}$	Generalized cartesian velocity
x	Generalized cartesian pose
x_i	Cartesian pose of frame i
x_{ref}	Reference task-space position
ω_i	Angular velocity of frame i
$\dot{L}_*$	Designed cylinder velocity vector
L	Cylinder length
L_i	Total length of cylinder actuating joint i
$L_{desired}$	Desired cylinder length configuration
L_{target}	Target cylinder length from IK
$\alpha_{a,2}$	Linkage angle corresponding to coupled joint 2
$\alpha_{b,i}$	Linkage angle corresponding to joint i
β_1	Linkage angle corresponding to joint 4
β_2	Linkage angle corresponding to joint 4
b_2	Linkage length corresponding to coupled joint 2
δ	Linkage length corresponding to joint 4
θ_1	Linkage angle corresponding to joint 4
C	Christoffel matrix
$\tilde{C}$	Cylinder space Christoffel matrix
D	Inertia matrix
$\tilde{D}$	Cylinder space inertia matrix
g	Gravity matrix
$\tilde{g}$	Cylinder space gravity matrix

$\mathcal{K}$	Kinetic energy
$\mathcal{L}$	Robot Lagrangian
$\mathcal{P}$	Potential energy
A_v	Area of orifice opening
ΔP	Actual orifice pressure drop
ΔP_{LS}	Desired orifice pressure drop
F_{drag}	Viscous drag of hydraulic fluid in cylinder
F_{ext}	External force applied to rod of cylinder
F_{frict}	Friction force inside hydraulic cylinder
F_{hyd}	Hydraulic force acting on piston
α_{SP}	Normalized swash-plate angle
P_{C1}	Pressure in chamber 1
P_{C2}	Pressure in chamber 2
P_{LS}	Sensed load pressure
P_s	Source pressure from pump
Q_{C1}	Flow rate into chamber 1
Q_{C2}	Flow rate into chamber 2
Q_p	Pump flow rate
Q_{sys}	Total system flow rate consumption
Q_v	Valve flow rate
$Q_{v,ref}$	Reference valve flow rate
$Q_{v,target}$	Target valve flow rate from IK
u_v	Input valve spool displacement
v_c	Cylinder rod velocity
w_a	Area gradient of orifice opening
x_c	Cylinder rod position
x_v	Valve spool displacement
$x_{v,ref}$	Reference valve spool displacement
ω_s	Angular velocity of pump
$E_i^{(m)}$	Elementary effect of the i:th parameter for trajectory m, Section 2.3.2
μ_i	Arithmetic mean of elementary effects for the i:th parameter, Section 2.3.2
μ_i^*	Mean of absolute elementary effects for the i:th parameter, Section 2.3.2

σ_i	Standard deviation of elementary effects for the i:th parameter, Section 2.3.2
Y	Model output, Section 2.3.2
λ	Damping penalty

Contents

List of Figures

List of Tables

1
Introduction

1.1 Background

Since their introduction, remote-controlled demolition robots have contributed to safer working environments and more efficient demolition work within the industry. As these systems become more widely adopted, Husqvarna is seeking ways to expand the capabilities of their demolition robots to improve its capabilities and enable new use cases. Husqvarna's DXR lineup of demolition robots are currently operated by actuating individual joints using a dual joystick, comparable to a conventional excavator. This control scheme is simple but requires a skilled operator to achieve accurate end-effector (EE) motion. For an unskilled operator, the per-joint control strategy frequently results in jerky and inconsistent movements of the end-effector and unintended motion.

Hence it is desirable to develop a more intuitive way for operators to control demolition robots, for example by controlling the end-effector in cartesian or polar coordinates, instead of individual joints. This has been tested and determined by Husqvarna to lower operator strain and required experience. Being able to manipulate the end-effector accurately through motion control would also introduce new use cases and enable further integration of new technologies focusing on automatization.

One use case example is to drill a sequence of holes in a concrete wall or floor. Linear movement along a surface and collinear movement along the axis of a tool would require significant skill of the operator when controlling each individual joint. Controlling the motion of the end-effector in a cartesian space would simplify these operations significantly. Such a control system calls for a way to constrain cartesian- and cylinder-space trajectories while respecting the dynamics of the hydraulic system.

1.2 Problem definition

The key problem is that it is hard for operators to move the EE accurately, prompting the need for a motion controller. The current implementation of this uses decoupled PID-controllers to control the angle of each joint. This approach achieves good precision but requires a significant amount of work to tune and is not easily transferrable between individual machines or variants, raising the need for model-based control methods, since it may be more intuitive to tune model parameters than gains.

The constraints of the machine are important for safe operation, but are currently handled as edge cases within the inverse kinematic solution, and not in the control problem. Hence, it is difficult to predict the behavior of the machine when near these limits. To better deal with constraints, model predictive control (MPC) will be investigated as an alternative that is capable of handling the physical constraints of the robot within the controller.

Instead of electric motors, demolition robots typically use hydraulic actuators, which represents a big challenge since hydraulic models tend to be large, non-linear and discontinuous. These properties are all undesirable when designing an MPC due to their generally high computational requirements. To be as computationally efficient as possible, linear MPC would be preferred with as small of a model as possible, which opens the door for model reduction and linearization. Thus, ways of reducing the model by identifying dominant dynamics will be investigated, along with methods for rejecting the resulting unmodeled disturbances.

Husqvarna has also identified the need to exploit the kinematic redundancy of the robot to achieve secondary motion objectives, such as penalizing movement in certain joints. With the current forward and backward reaching inverse kinematics (FABRIK), this is not possible since it is not Jacobian-based. Thus Jacobian algorithms will be investigated as well as quadratic program formulations of such algorithms, since it is necessary to constrain the reference trajectory to feasible robot motion.

1.3 Research questions

The problem analysis raises the following questions:

- Which dynamics have the largest impact on tracking performance for the robot? What is most important to model?
- Can the hydraulic actuators be decoupled?
- How should unmodeled dynamics, such as hydraulic coupling, friction and load-sense dynamics be handled? Will MPC be able to reject those disturbances?
- How should references that exploit redundant joint motion be generated?
- Can the reference tracking and reference generation problems be combined and solved by one MPC?

1.4 Aim

The aim of this thesis is to develop an MPC-based motion controller for a four-link hydraulic robot manipulator to control the end-effector in cartesian space. To achieve this aim, the following objectives are formulated:

- Formulate a non-linear model of the hydraulic dynamics and the manipulator dynamics.
- Conduct a sensitivity analysis on the non-linear model to identify a reduced linear control model.
- Analyze the manipulator kinematics and hydraulic kinematics to map cylinder and task-space.
- Design a constrained inverse kinematics algorithm for generating references with capability to utilize redundant joint motion.
- Formulate an MPC with positional, velocity and hydraulic constraints that controls cylinder velocity in order to track provided references.

1.5 Scope

The following boundaries has been set for the project:

- The main control scheme of interest is model predictive control due to its constraint handling.
- The robot is considered stationary any time the arm is operated.
- The swing joint which rotates the whole robot at the base will be considered disabled such that the robot can be assumed to operate in a plane. Effectively reducing the robot from five links and 4-DOM to four links and 3-DOM.
- Cylinder position is the only measurable system quantity.
- Frictional and viscous effects will be considered but important parameters will not be accurately identified as a part of this project.

1.6 Ethics and sustainability

As automatization continues to proliferate in society it will only have an increasing impact on peoples lives. It is up to engineers to ensure these effects are positive and serve society. One of the greatest advantages of demolition robots is that they reduce risk of injury to people by taking their place in hazardous environments.

However, this leads to a common concern with robots, that they will replace human labor. Ethical development should therefore consider employment impact and how workers can contribute as operators of robots rather than manual laborers, allowing more efficient and faster demolition, thus increasing the value of their work.

The overarching aim of this project is to create a more intuitive way of controlling a demolition robot rather than increasing autonomy. This means a lower risk of unintentional movements and thereby reducing the risk of harm to humans or surrounding structures.

2
Theory

The need of a dynamic model description is fundamental to predict the behavior of a system over time and obtain an optimal decision or control signal [2]. This thesis focuses on a hydraulic four-link robot, which is a complex and multi-domain system, requiring modeling of both hydraulic actuator dynamics as well as robot manipulator dynamics. Both systems are highly non-linear in addition to hydraulic systems also containing hybrid dynamics. This chapter presents one hybrid model and two simplified dynamic models as well as a description of the manipulator dynamics.

2.1 Hydraulic model

A reduced single actuator hydraulic system can be modeled with three main parts: the pump, a control valve and an actuator.

2.1.1 Actuator model

The actuators on the demolition robots are double-acting cylinders, meaning hydraulic flow can be directed to either side of the piston to make it extend or retract. The rod-end side of the piston will be denoted as chamber 2 (C2) and the opposing cap-end side will be denoted chamber 1 (C1), see Figure 2.1.

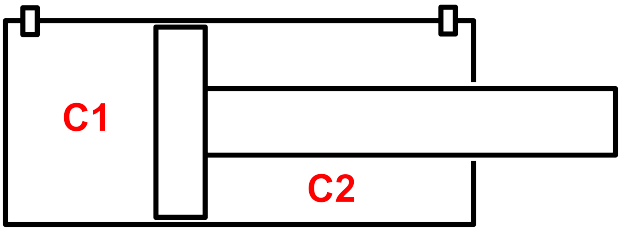


Figure 2.1: Cross-section of hydraulic cylinder

Focusing on this actuator, the velocity and acceleration of the piston can be modeled as

$$\dot{x}_c = v_c \tag{2.1}$$

$$\dot{v}_c = \frac{F_{\text{hyd}} - F_{\text{frict}} - F_{\text{drag}} - F_{\text{ext}}}{m_p} \tag{2.2}$$

$$F_{\text{hyd}} = P_{\text{C1}} A_{\text{C1}} - P_{\text{C2}} A_{\text{C2}} \tag{2.3}$$

where x_c is the position of the piston in the cylinder and v_c is its velocity. F_{hyd} is defined as the hydraulic force resulting from the pressure difference over the piston. F_{frict} is mechanical friction between rod and chamber, F_{drag} is viscous drag and F_{ext} is external force applied to the cylinder rod [3]. These three forces will be described in following sections. The areas A_{C1}, A_{C2} and the pressures P_{C1}, P_{C2} are the piston areas and pressures corresponding to each cylinder chamber respectively. Since the rod reduces the effective area of the rod-side considerably, the cylinder will be stronger during extension compared to retraction.

2.1.2 Friction and drag modeling

In Equation 2.2, F_{frict} and F_{drag} are forces resulting from friction and viscous drag within the cylinder respectively. Most of the friction within hydraulic cylinders are caused by chamber- and rod-seals [4], and viscous drag is a result of fluid dynamics within the cylinder, which is often described as a linear velocity dependent term

$$F_{\text{drag}} = c_v v_{\text{c}} \tag{2.4}$$

where c_v can be interpreted as a lumped viscous drag coefficient [3]. The classical model of friction is described with static friction (stiction) and Coulomb (sliding) friction and is a well known simplification of friction. Models that switch between static and sliding friction involve either discontinuous saturation- or sign functions to determine the direction and magnitude of the friction force. For simulation purposes, a continuous tanh function can be used instead to avoid these problems.

$$F_{\text{frict}} = F_{\text{c}} \tanh(k_{\text{tanh}} v_{\text{c}}) \tag{2.5}$$

where F_{c} is Coulomb friction constant and k_{tanh} is a numerical smoothing constant which determines the rate of change near zero cylinder velocity v_{c} [5]. However, friction is often a very non-linear phenomenon because of multiple other effects caused by material properties and lubrication. The Stribeck effect describes an apparent phenomenon where lubrication effects between two materials can cause a temporary reduction in friction as velocity increases. This effect often results in what is called stick-slip behavior which can cause stability issues and oscillations [6]. Friction in hydraulic cylinders have in some cases been shown to be direction dependent and time varying based on the age and wear on the cylinder, as well as the quality of the hydraulic fluid [6] [7] [8]. It is therefore important to characterize friction in a hydraulic cylinder from a control perspective, and there are several experimental studies that show significantly improved control by friction compensation, [9] [7]. Modeling friction accurately is therefore a relevant for continued research if determined to be a significant factor for exact positioning of the cylinder [5].

2.1.3 Pressure dynamics modeling

Pressure dynamics in both the cylinder and pump is governed by the continuity equation [3]

$$\sum Q = \frac{V}{\beta}\frac{dP}{dt} \tag{2.6}$$

where Q is the net flow into a volume V with bulk modulus β. When a force acts on the cylinder, the instantaneous volume for the cylinder chambers can be described as

$$V = V_0 + A_c x_c \tag{2.7}$$

where V_0 is a constant volume associated with the cylinder, including connecting hoses, fittings and other dead space within the cylinder. V_0 affects the characteristics of the cylinder as it must still be pressurized despite it not being displaced by the cylinder. A_c describes a generalized piston area. For a single sided hydraulic cylinder with variable volume, the continuity equation may now be written as

$$\dot{P} = \frac{\beta}{V_0 + A_c x_c}(Q - c_l P - A_c \dot{x}_c) \tag{2.8}$$

where Q is fluid inflow from the valve and c_l is the leakage coefficient of the cylinder [3]. The leakage between the two chambers can often be ignored [7]. The model shows that the piston displacement x_c in the denominator creates a non-linear coupling between pressure, velocity and displacement. For a double acting cylinder as shown in Figure 2.1, both chambers contribute to the hydraulic force in Equation 2.3 and must each be modeled. The pressure dynamics in the chambers follows directly from Equation 2.8, where the volume of C1 increases and C2 decreases as x_c increases. The leakage term c_l now acts on the pressure difference between the two chambers ($P_{C1} - P_{C2}$). The pressure equations are then formulated as

$$\dot{P}_{C1} = \frac{\beta}{V_0 + A_{C1} x_c}(Q_{C1} - c_l(P_{C1} - P_{C2}) - A_{C1} v_c)\,, \tag{2.9}$$

$$\dot{P}_{C2} = \frac{\beta}{V_0 + A_{C2}(x_{c,\max} - x_c)}(Q_{C2} + c_l(P_{C1} - P_{C2}) + A_{C2} v_c)\,, \tag{2.10}$$

where $x_{c,\max}$ is the maximum stroke length of the piston. The cylinder dead volume V_0 is assumed equal for both chambers. This is a simplification, as in practice the dead volume may differ due to differing hose lengths or fitting geometry.

2.1.4 Valve model

All flow to and from the cylinder is metered by a four-way proportional directional control valve. It consists of a cylindrical spool capable of both positive and negative displacement inside a valve body. Depending on the spool position it covers and reveals orifices that direct fluid from the pump to one side of the cylinder and returns fluid from the other side to the hydraulic fluid reservoir (tank) of the hydraulic

system. Steady, high Reynolds number flow of an incompressible fluid through an orifice in the valve can be modeled using the orifice equation [3]

$$Q_v = A_v(x_v) c_d \sqrt{\frac{2}{\rho} \Delta P} \tag{2.11}$$

where $A_v(x_v)$ is the area of the orifice opening as a function of the spool displacement x_v. c_d is the discharge coefficient, ρ is fluid density and ΔP is the pressure difference over the valve. Note that the square root of the pressure creates another non-linear coupling between pressure and valve flow. Due to the inherent workings of directional control valves required for double acting cylinders, there will be hybrid dynamics as a result of switching the pump pressure and reservoir between the two sides of the cylinder depending on if the piston is extended or retracted. This manifests as switching between two sets of valve flow equations with for example an if statement in the implementation, depending on if the valve spool is displaced positively or negatively [10]. The valve is critically centered, meaning only two of the four orifices are open at any given time. The flow rate through each open orifice can be modeled as

$$Q_{C1} = \begin{cases} c_d \, A_v(x_v) \sqrt{\frac{2}{\rho} \left| P_s - P_{C1} \right|} \operatorname{sign}(P_s - P_{C1}) & x_v \geq 0, \\ -c_d \, A_v(x_v) \sqrt{\frac{2}{\rho} \left| P_{C1} - P_t \right|} \operatorname{sign}(P_{C1} - P_t) & \text{otherwise,} \end{cases} \tag{2.12}$$

$$Q_{C2} = \begin{cases} -A_r \, c_d \, A_v(x_v) \sqrt{\frac{2}{\rho} \left| P_{C2} - P_t \right|} \operatorname{sign}(P_{C2} - P_t) & x_v \geq 0, \\ A_r \, c_d \, A_v(x_v) \sqrt{\frac{2}{\rho} \left| P_s - P_{C2} \right|} \operatorname{sign}(P_s - P_{C2}) & \text{otherwise,} \end{cases} \tag{2.13}$$

where P_s and P_t are supply pressure and reservoir (tank) pressure respectively [10] and $A_v(x_v)$ is the metering area as a function of spool position [3]. The A_r factor in Equation 2.13 aims to account for asymmetric piston areas in asymmetric spool valves. Finding an asymmetric valve $A_r = A_{\mathrm{C2}}/A_{\mathrm{C1}}$ might not always be possible given standard asymmetric valve dimensions, but literature suggests that selecting a valve with metering orifice area close to the cylinder area ratio is the most adequate configuration for ensuring good performance of the system [11].

Some spool valves are equipped with internal flow regulators, which are controlled using external flow references. In one such valve system, the displacement x_v with relation to a displacement reference $x_{\mathrm{v,ref}}$ can be approximated as a first order response equivalent to

$$\dot{x}_v = \frac{x_{\mathrm{v,ref}} - x_v}{\tau_v} \tag{2.14}$$

where $x_{v,\mathrm{ref}}$ is the spool position reference and τ_v is an experimentally found time constant of the valve. A feed-forward (FF) control structure can be used to obtain

a closed loop valve flow control structure that uses flow references as input [12]. By inverting the orifice equation 2.11, a spool valve area reference can be calculated from a flow reference as

$$A_v(x_{\mathrm{v,ref}}) = \frac{Q_{\mathrm{v,ref}}}{c_d\sqrt{\frac{2}{\rho}\Delta P}} \tag{2.15}$$

where $Q_{\mathrm{v,ref}}$ is the valve flow reference and ΔP is the measured pressure drop over the valve. The reference spool position $x_{\mathrm{v,ref}}$ can then be derived from the orifice area relationship $A_v(x_{\mathrm{v,ref}})$ provided sufficient knowledge about the valve orifice, such as its shape and size [12]. Of note is that the orifice equation is only valid under the assumptions mentioned previously, but for modeling purposes, the orifice equation is also used in circumstances with unsteady, slightly compressible or low Reynolds number flow, for mathematical convenience [3].

2.1.5 Pump model

The pump considered is a variable displacement axial-piston pump with load sensing. This type of pump can alter its flow output by controlling the angle of the swash-plate. The load sensing function allows the pump to continuously vary the displacement such that it delivers flow that results in a desired pressure drop over an orifice. This means in practice that the pump will adjust to deliver a pressure that always exceeds the load or consumer pressure, and for example provides a constant piston extension velocity even if the external force varies. The pump pressure model is based on the continuity equation [3]

$$\dot{P}_s = \frac{\beta}{V_s}(Q_p - Q_{\mathrm{sys}} - c_s P_s) \tag{2.16}$$

$$Q_{\mathrm{sys}} = \max(Q_{C1}, 0) + \max(Q_{C2}, 0) \tag{2.17}$$

where Q_p is the pump discharge flow rate, Q_{sys} is the system flow rate consumption. c_s is the internal pump leakage coefficient and V_s is the total volume associated with the pump. Pump flow can be calculated as the product of angular displacement V_d, angular velocity ω_s and the normalized swash-plate angle $\alpha_{SP}\in[0\ \ 1]$ [10] [13].

$$Q_p = V_d\omega_s\alpha_{\mathrm{SP}} \tag{2.18}$$

The swash-plate is connected to each piston and alters the stroke length of each piston by pivoting about the rotating drive shaft, and thus also total displacement. A piston controlled by a spool valve pushes on the swash-plate, creating a lever that actuates the swash-plate. The valve is fed by both the pump outlet pressure and the load pressure across an orifice. Whenever the pump pressure does not equal the load pressure plus the pressure from a valve spring, the piston pushes on the swash-plate until the pressures are in equilibrium, meaning the desired pressure differential is achieved. While there exists many descriptions of such systems, the change in swash-plate angle α_{SP} is fundamentally controlled by the arrangement of mechanical components, and their exact dimensions [14]. Because these details are not usually provided in datasheets, some researchers have had to disassemble and

measure dimensions and weights to ensure model accuracy [15]. Consequently, a generalized dynamic model of the swash-plate is difficult to derive, so a simplified representation is required for simulation purposes.

2.2 Robot modeling

A robot model typically involves kinematics that describe the movement of the chained links, as well as dynamics describing the torque and forces on the links, arising from movement. Both aspects are essential for successful robot control.

2.2.1 Forward Kinematics

To describe the kinematics of a manipulator, cartesian coordinate systems are attached to each link together with a world coordinate system at the base [16] [17]. Homogeneous transformation matrices $\mathbf{H}_{i-1}^{i}$

$$\mathbf{H}_{i-1}^{i} = \begin{bmatrix} \mathbf{R}_{i-1}^{i} & \mathbf{t}_{i-1}^{i} \\ 0 & 1 \end{bmatrix} \tag{2.19}$$

describe the rotation and positional offset in each frame i relative to frame $i-1$, where $\mathbf{R}_{i-1}^{i}$ is a 3×3 rotation matrix and $\mathbf{t}_{i-1}^{i}$ is a 3×1 position vector. By computing each transformation in the chain of links, an overall transformation between the end link, called the end-effector(EE), and the base can be derived to calculate the pose of the EE. Assignment of coordinate frames is typically done according to the Denavit-Hartenberg notation. In this notation, the coordinate axle Z_i of every frame i is aligned with the rotation axis of joint i. Coordinate axle X_i is aligned with the common normal of Z_i and Z_{i-1}. With this specified, the transformation can then be described with four parameters:

- θ: Rotation around Z-axis.
- d: Offset along Z-axis.
- a: Offset along X-axis.
- α: Rotation around X-axis.

With these parameters, the transformation from frame $i-1$ to i can be computed with the following expression:

$$\boldsymbol{H}_{i-1}^{i} = \begin{bmatrix} \cos(\theta) & -\sin(\theta)\cos(\alpha) & \sin(\theta)\sin(\alpha) & a\cos(\theta) \\ \sin(\theta) & \cos(\theta)\cos(\alpha) & -\cos(\theta)\sin(\alpha) & a\sin(\theta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.20}$$

An example of a Denavit-Hartenberg table for a five-link robot is shown below and lists the transformations along a jointed chain of links that describe the location of each frame in reference to a base frame. q_i is the joint angle and r_i is the link length.

Link	θ	d	a	α
1	q_1	r_1	0	$\frac{-\pi}{2}$
2	q_2	0	r_2	0
3	q_3	0	r_3	0
4	q_4	0	r_4	0
5	q_5	0	r_5	0

Table 2.1: DH-Parameters for a five-link robot

The transformations in Table 2.1 can be visualized in Figure 2.2, where Z_i is blue, X_i is red, Y_i is green and the base frame is black.

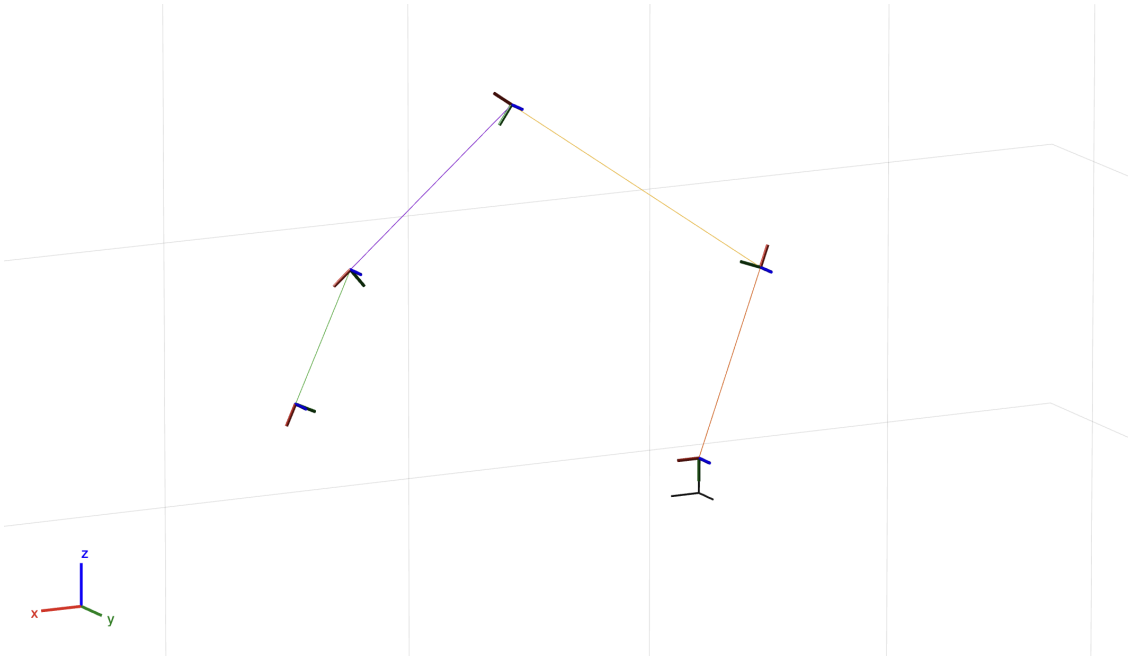


Figure 2.2: Robot frame visualization

2.2.2 Jacobian

Robotics often concerns two types of spaces, joint-space and task-space. Joint-space encompasses the angular movement of all joints while task-space is the cartesian space where end-effector tasks are often defined. The mapping between joint-space and task-space is defined by the manipulator jacobian, representing differential kinematics [16]. The linear and angular velocity of the end-effector $\dot{x}$ can be related to the vector of joint velocities $\dot{q}$ as

$$\dot{x} = J\dot{q} \tag{2.21}$$

where J is the Jacobian. $\dot{x}$ can be divided into linear (v) and angular (ω) components as

$$\dot{x} = \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2.22}$$

Assuming all joints are revolute joints, the angular velocity of the end effector can be expressed as a sum of the contribution from each joint, given that they are expressed relative to the same base frame. The angular velocity ω_i of frame i relative to frame $i-1$ is the i^{th} joint velocity $\dot{q}$ about the axis of rotation $\mathbf{z}_{i-1}$ which consists of the third column of the rotation matrix belonging to the corresponding homogeneous transformation matrix. [16].

$$\omega_i = \dot{q}_i \mathbf{z}_{i-1} \tag{2.23}$$

The linear velocity of a point of interest, such as the end-effector for example, due to rotation of joint i can then be expressed as

$$v = \omega_i \times \mathbf{r}_i \tag{2.24}$$

where $\mathbf{r}_i$ is

$$\mathbf{r}_i = \mathbf{t}_0^n - \mathbf{t}_0^{i-1} \tag{2.25}$$

which represents the distance between the point of interest and the axis of rotation of the joint i, expressed as the difference of the vectors stretching from the base frame to the axis of joint rotation and base frame to the point of interest.
The Jacobian is assembled column by column for each joint that has a contribution to the movement of the point of interest. By inserting Equations 2.23, 2.24 and 2.25, the i^{th} column of the end-effector Jacobian can be defined as

$$J_0^n = \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{t}_0^n - \mathbf{t}_0^{i-1}) \\ \mathbf{z}_{i-1} \end{bmatrix} \tag{2.26}$$

The dimensions of the Jacobian is $n_x \times n_q$, where n_x is the number of degrees of motion (DOM) of the end-effector and n_q is the number of degrees of freedom (DOF) of the robot, equivalent with the number of joints. Typically it is necessary to distinguish DOF and DOM when not all degrees of freedom are independent of each other.

2.2.3 Inverse kinematics

Inverse kinematics is the inverse of the problem discussed in section 2.2.1, namely to find the the joint angles given an end-effector pose. Direct inverse kinematics solve for a joint angle configuration given a certain end-effector pose, however there are often multiple solutions for any given pose and the equations are often highly nonlinear [17]. An alternative is differential inverse kinematics which can be utilized to solve for the joint velocity $\dot{q}$ given an end-effector velocity $\dot{x}$. Here, the inverse relation of Equation 2.21 is presented.

$$\dot{q} = J^{-1}\dot{x} \tag{2.27}$$

Inversion poses the requirement that the Jacobian is full-rank and square, which is not always the case. For robots with redundant joints, meaning joints that do not add a degree of freedom in task-space, the Jacobian will be non-square. Further problems occur if a manipulator passes a singularity, two or more joint axes may then align or become coplanar, meaning they will contribute to end-effector motion in the same direction. This results in the Jacobian having linearly dependent

columns, and thus loses rank and invertibility.

2.2.4 Pseudo-inverse Jacobian

To overcome the problems with the inverse Jacobian, the pseudo-inverse Jacobian $J^\dagger$ can be utilized.

$$\dot{q} = J^\dagger \dot{x} \tag{2.28}$$

It works as a replacement in many cases when the regular inverse does not exist and can be used to solve systems of linear equations which have either many or no unique solutions [18]. Importantly, in robotics it allows Jacobian inversion for redundant robots, as well as for robots in singular configurations.
For a redundant robot with full-rank Jacobian the pseudo-inverse Jacobian provides an exact solution and can be calculated as [17]

$$J^\dagger = J^\top (JJ^\top)^{-1} \tag{2.29}$$

If the Jacobian is also square the pseudo-inverse reduces to the regular inverse J^{-1}. For a general case the pseudo-inverse can be constructed from the Jacobian by inverting each value in $\Sigma \in \mathbb{R}^{n_q \times n_x}$ which contains the singular values of J.

$$J^\dagger = U \Sigma^\dagger V^\top \tag{2.30}$$

$U \in \mathbb{R}^{n_x \times n_x}$ and $V \in \mathbb{R}^{n_q \times n_q}$ are rotation matrices, containing the left and right singular vectors respectively.

2.2.5 Inverse kinematics as an optimization problem

Solving for joint velocities with a pseudo-inverse method means that unfeasible references could be sent to the controller. It is thus desirable to set constraints in position and velocity in cylinder-space to ensure that task-space movements cannot violate the physical constraints of the hydraulic cylinders. A relatively easy way to implement a constrained solution for inverse kinematics is by reformulating Equation 2.2.4 as an optimization problem. Solvers for quadratic problems are readily available, but is more computationally intensive than other solutions. However, the quadratic programming method offers many benefits apart from constrained solutions which are desirable [19]. A quadratic program is used to find a minimal solution to an optimization problem using a cost function and associated decision variables. The general formulation of a constrained quadratic program can be written as

$$\begin{aligned} \min_{\mathbf{z}} \quad & \tfrac{1}{2}\mathbf{z}^\top H \mathbf{z} + f^\top \mathbf{z} \\ \text{s.t} \quad & A_{in}\mathbf{z} \leq b_{in} \end{aligned} \tag{2.31}$$

$\mathbf{z}$ are the decision variables, H is a symmetric positive semi-definite cost, and f is linear cost. A_{in} and b_{in} form the inequality constraints. For the purposes of inverse kinematics, Equation 2.2.4 can be reformulated as a quadratic program by minimizing the squared difference as:

$$\min_{\dot{q}} \|\dot{x} - J\dot{q}\|^2 \tag{2.32}$$

Which is the basis of a quadratic program. Since $\|\cdot\|$ denotes the euclidean norm, or 2-norm, this expression can be expanded to quadratic and linear terms as follows:

$$\begin{aligned} \|\dot{x} - J\dot{q}\|^2 &= (\dot{x} - J\dot{q})^\top (\dot{x} - J\dot{q}) \\ &= \dot{x}^\top \dot{x} - \dot{x}^\top J\dot{q} - \dot{q}^\top J^\top \dot{x} + \dot{q}^\top J^\top J\dot{q} \\ &= \dot{x}^\top \dot{x} - \dot{x}^\top J\dot{q} - (\dot{x}^\top J\dot{q})^\top + \dot{q}^\top J^\top J\dot{q} \\ &= \dot{x}^\top \dot{x} - 2\dot{q}^\top J^\top \dot{x} + \dot{q}^\top J^\top J\dot{q} \end{aligned} \tag{2.33}$$

The corresponding cost expressions are then $H = J^\top J$ and $f = -2J^\top \dot{x}$. $\dot{x}^\top \dot{x}$ can be ignored in the optimization problem as it does not affect the solution.

2.2.5.1 Damped least-squares inverse

Though the pseudo-inverse Jacobian solves the problem of invertibility in singular configurations, it may still produce unfeasible joint velocities during such conditions. In the neighborhood of singular configurations, a damped least-squares inverse Jacobian is more numerically robust. Instead of minimizing $\|\dot{x} - J\dot{q}\|^2$ alone, one can add a penalty on joint velocity magnitude as

$$\min_{\dot{q}} \|\dot{x} - J\dot{q}\|^2 + \lambda^2 \|\dot{q}\| \tag{2.34}$$

By taking the gradient of this function and setting it to zero, one can attain an expression for the minimizer as

$$\dot{q} = (J^\top J + \lambda^2 I)^{-1} J^\top \dot{x} \tag{2.35}$$

where the term $(J^\top J + \lambda^2 I)^{-1} J^\top$ is the inverse of the damped least-squares Jacobian $J^\#$. The λ term is the damping penalty. There is however one drawback of this method, namely that motion tracking accuracy is lost by virtue of adding a second term to the objective function.

2.2.6 Jacobian null-space

As mentioned in the introduction, it would be useful to be able to control not only the motion of the EE, but also the behavior of the arm itself, which can be done by exploiting null-space movement. The null-space of the Jacobian is the set $\mathcal{N}(J)$ of joint velocities that yield null task velocity in the current configuration. For a kinematically redundant robot, such as the one discussed in this thesis, a nonempty null-space always exists [17]. By projecting a joint velocity vector onto the Jacobian null-space, a secondary objective can be added in the following way

$$\dot{q} = J^\dagger \dot{x} + (I - J^\dagger J)\dot{q}_* \tag{2.36}$$

where $(I - J^\dagger J)$ is a projector of $\dot{q}_*$ onto $\mathcal{N}(J)$. Since the secondary objective only produces null task velocities, the primary EE motion tracking objective remains

satisfied. However, by designing $\dot{q}_*$ one can add desired movement in the rest of the robot. To achieve better stability of the robot, it is possible to design $\dot{q}_*$ such that movement in joints close to the robot base is penalized. These joints are typically heavier and prioritizing movement in other joints keeps the center of gravity closer to the robot base [20]. To achieve this, $\dot{q}_*$ can be designed as

$$\dot{q}_* = 2\mathbf{W}(q - q_{desired}) \tag{2.37}$$

where $\mathbf{W}$ is a diagonal weight matrix, representing the penalty for each joint. $q_{desired}$ is the joint configuration from which deviation is penalized.

In the quadratic program formulation, the null-space projection appears in the linear term of the cost function as

$$f = -2J^\top \dot{x} + (I - J^\dagger J)^\top \dot{q}_* \tag{2.38}$$

and in the quadratic term as

$$H = 2J^\top J + 2(I - J^\dagger J)^\top \mathbf{W}(I - J^\dagger J) \tag{2.39}$$

2.2.7 Dynamics

The dynamic response of the robot is of vital interest in order to accurately control it. A common way to model robot dynamics is the Euler-Lagrange approach, which describes the evolution of a mechanical system in terms of kinetic and potential energy. The Lagrangian is formed as

$$\mathcal{L} = \mathcal{K} - \mathcal{P} \tag{2.40}$$

where $\mathcal{K}$ and $\mathcal{P}$ are the kinetic and potential energy of the system and are used to form the Euler-Lagrange equation [16]

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i \tag{2.41}$$

where $\mathcal{L}$ is the Lagrangian and τ_i is the joint torque. The kinetic energy of any rigid link can be expressed as

$$\mathcal{K}_i = \frac{1}{2} m_i v_i^\top v_i + \frac{1}{2} \omega_i^\top \mathcal{I} \omega_i \tag{2.42}$$

where m_i is the link mass, v and ω are the linear and rotational velocities and $\mathcal{I}$ is the 3×3 inertia tensor, evaluated at a frame parallel to the link frame but with origin at the link center of mass. The inertia tensor contains the inertial components consisting of the three principal moments of inertia about the X, Y and Z axis as well as the off diagonal cross products of inertia [16]. The inertia tensor for each link can often be extracted from CAD software.

If the linear and angular velocity of any point on a link can be expressed in terms of a suitable Jacobian matrix and the joint velocities, then equation 2.42 can be expanded for a a series of links [16]

$$\mathcal{K} = \frac{1}{2}\dot{q}^\top \left[\sum_{i=1}^{n} (m_i J_{v_i,cm}^\top J_{v_i,cm} + J_{\omega_i,cm}^\top \mathbf{R}_i \mathcal{I} \mathbf{R}_i^\top J_{\omega_i,cm}) \right] \dot{q} \tag{2.43}$$

$$\mathcal{K} = \frac{1}{2}\dot{q}^\top D(q)\dot{q} = \frac{1}{2}\sum_{i,j} d_{ij}(q)\dot{q}_i\dot{q}_j \tag{2.44}$$

where d_{ij} are entries of the inertia matrix D, m_i is the link mass, $\mathcal{I}$ is the inertia tensor, J_{cm} is the Jacobian of the link center of masses with either linear or angular components and $\mathbf{R}_i$ is the orientation transformation from the base frame to the inertial frame. This transformation appears naturally in conjunction with calculating the geometric center of mass Jacobian.

The potential energy of the robot is caused only by gravity and can be expressed as

$$\mathcal{P}_i = m_i \boldsymbol{g}^\top \mathbf{t}_i \tag{2.45}$$

$$\mathcal{P} = \sum_{i=1}^{n} \mathcal{P}_i \tag{2.46}$$

where $\mathbf{t}_i$ is the translational vector of the link center of mass corresponding to the orientational transformation $\mathbf{R}_i$ and $\mathbf{g}$ is the gravitational vector containing the gravitational acceleration in the base frame. The Lagrangian can thus be expressed as

$$\mathcal{L} = \mathcal{K} - \mathcal{P} = \frac{1}{2}\sum_{i,j} m_{ij}(q)\dot{q}_i\dot{q}_j - \mathcal{P} \tag{2.47}$$

When taking the partial derivative of the Lagrangian with respect to both $\dot{q}$ and q, there will appear terms that are quadratic in the first derivative of q, called Christoffel symbols. These can be divided into two types, those involving terms like q_i^2 are called centrifugal terms while those like $q_i q_j$ are called Coriolis terms [16]. The partial derivative of the Lagrangian with respect to joint velocity k, and its time derivative, is

$$\frac{\partial \mathcal{L}}{\partial \dot{q}_k} = \sum_j d_{kj}\dot{q}_j \tag{2.48}$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_k} = \sum_j d_{kj}\ddot{q}_j + \sum_{i,j} \frac{\partial d_{kj}}{\partial q_i}\dot{q}_i\dot{q}_j \tag{2.49}$$

The partial derivative of the Lagrangian with respect to joint angle k is similarly given as

$$\frac{\partial \mathcal{L}}{\partial q_k} = \frac{1}{2}\sum_{i,j} \frac{\partial d_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j - \frac{\partial \mathcal{P}}{\partial q_k} \tag{2.50}$$

For each joint k the Euler-Lagrange equations can now be written as

$$\sum_j d_{ij}\ddot{q}_i + \sum_{i,j}\left[\frac{\partial d_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial d_{ij}}{\partial q_k}\right]\dot{q}_i\dot{q}_j + \frac{\partial \mathcal{P}}{\partial q_k} = \tau_k \tag{2.51}$$

and due to symmetry of the double summation the first middle term becomes

$$\sum_{i,j} \frac{\partial d_{kj}}{\partial q_i}\dot{q}_i\dot{q}_j = \frac{1}{2}\sum_{i,j}\left[\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j}\right]\dot{q}_i\dot{q}_j \tag{2.52}$$

Thus the middle term, containing the Christoffel symbols, is calculated as

$$c_{ijk} = \frac{1}{2}\left[\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right]\dot{q}_i\dot{q}_j \tag{2.53}$$

The matrix form of the Euler-Lagrange equation can now be written as

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \tag{2.54}$$

where the $(k,j)^{th}$ element of the C matrix is

$$c_{kj} = \sum_{i=1}^{n} c_{ijk}(q)\dot{q}_i \tag{2.55}$$

and the gravity matrix is

$$g(q) = \begin{bmatrix} g_1(q_1) & \dots & g_n(q_n) \end{bmatrix}^\top \tag{2.56}$$

$$g_k(q_k) = \frac{\partial \mathcal{P}}{\partial q_k} \tag{2.57}$$

2.3 Sensitivity analysis

A sensitivity analysis is often performed to quantify to which degree unknown parameters in a model affect its output. Algorithms used for this purpose are often categorically divided into local and global methods which are used at different stages of modeling. Depending on the chosen method, metrics referred to as elementary effects and sensitivity indices are used [21] [22].

2.3.1 One-at-a-time sensitivity analysis

Local sensitivity analysis is concerned with the effect of small perturbations of parameter values. One-at-a-time (OAT) analysis is a simple method where one input parameter is varied while remaining parameters remain at their nominal value. In this case, the partial derivative of the output can be used as a metric, often approximated using finite differences [22]. While being computationally efficient, OAT analysis only shows the sensitivity of one parameter given a specific combination of other parameters. Regardless, this analysis method is often used since it is more intuitive than other methods, which is often useful in initial screening of models [21].

2.3.2 Morris sensitivity analysis

The Morris method is considered a global sensitivity analysis method with the primary goal of categorizing parameters based on the amount of linear and non-linear interaction within the model. While the OAT method varies each parameter based on one nominal model, the Morris method performs r independent trajectories, each consisting of OAT simulations starting from random parameter values selected from

the parameter space. The randomization allows this method to capture the behavior of the model across the entire the parameter range, making it a global screening method.

Following the formulation provided by Dellino and Meloni [21], the method calculates the elementary effect $E_i^{(m)}$ for the i:th parameter p by perturbing it by a discrete step Δ within the normalized parameter space

$$E_i^{(m)} = \frac{f(p_1, \ldots, p_i + \Delta, \ldots, p_k) - f(p_1, \ldots, p_i, \ldots, p_k)}{\Delta} \tag{2.58}$$

The elementary effects is then be used to calculate two indices. The first is the mean of the absolute values of the elementary effects

$$\mu_i^* = \frac{1}{r} \sum_{m=1}^{r} |E_i^{(m)}| \tag{2.59}$$

which means μ_i^* quantifies the influence of the i:th parameter on the output Y. The second index, σ_i, quantifies the standard deviation of the elementary effects as

$$\sigma_i = \sqrt{\frac{1}{r-1} \sum_{m=1}^{r} (EE_i^{(m)} - \mu_i)^2} \tag{2.60}$$

where μ_i is the arithmetic mean of the elementary effects. As noted in [21], the Morris method classifies parameters into three groups based on their position in the (μ^*, σ) plane. A parameter with low μ^* and low σ has negligible effect on the output, and could therefore be fixed at any value within its range. A parameter with high μ^* and low σ has a strong and approximately linear effect, meaning the effect of a perturbation is roughly the same regardless of the values of other parameters. A parameter with high μ^* and high σ is influential but has a non-linear behavior or interaction with other parameters. The diagonal reference line $\mu^* = \sigma$ serves as a rough boundary between the linear and non-linear groups.

2.4 Model Predictive Control

Different types of model descriptions and MPC schemes can be used depending on the structure and requirements of a control problem. This section covers linear time-invariant systems, discrete time models and the uncondensed formulation for the solution to the constrained receding horizon problem [2]. The uncondensed problem can be solved with quadratic programming tools, such as `quadprog` in MATLAB, where all states and inputs over the horizon acts as optimization variables.

A continuous linear time-invariant state space model is formulated as follows where $\mathcal{A}_c$ and $\mathcal{B}_c$ are the continuous state and input matrices.

$$\dot{\mathrm{x}} = \mathcal{A}_c \mathrm{x} + \mathcal{B}_c \mathrm{u} \tag{2.61}$$

$$\mathrm{y} = \mathcal{C} \mathrm{x} + \mathcal{D} \mathrm{u} \tag{2.62}$$

$$\mathcal{A}_c \in \mathbb{R}^{n\times n} \quad \mathcal{B}_c \in \mathbb{R}^{n\times m} \quad \mathcal{C} \in \mathbb{R}^{p\times n} \quad \mathcal{D} \in \mathbb{R}^{p\times m}$$
$$\mathrm{x} \in \mathbb{R}^n \quad \mathrm{u} \in \mathbb{R}^m \quad \mathrm{y} \in \mathbb{R}^p$$

An MPC requires a discrete model which can be attained by discretizing a continuous model in the following way where h is the discrete time step. This yields the discrete state and input matrices $\mathcal{A}$ and $\mathcal{B}$. Note that the $\mathcal{C}$ and $\mathcal{D}$ output matrices remain the same as the continuous matrices.

$$\mathrm{x}(t+h) = e^{\mathcal{A}_c h}\mathrm{x}(t) + \int_0^h e^{\mathcal{A}_c s}\mathcal{B}_c \mathrm{u}(s)\,\mathrm{d}s \tag{2.63}$$

where

$$e_c^{\mathcal{A}} = \sum_{k=0}^{\infty} \frac{1}{k!}\mathcal{A}_c^k. \tag{2.64}$$

Assume that the control signal u is piecewise constant as

$$\mathrm{u}(s) = \mathrm{u}(kh), \quad kh \le s < (k+1)h.$$

where h is the time step and k its index. And insert in previous equation with $t = kh$ to get

$$\mathrm{x}(k+1) = \underbrace{e^{\mathcal{A}_c h}}_{\mathcal{A}}\mathrm{x}(k) + \underbrace{\int_0^h e^{\mathcal{A}_c s}\,\mathrm{d}s\,\mathcal{B}_c}_{\mathcal{B}}\mathrm{u}(k) = \mathcal{A}\mathrm{x}(k) + \mathcal{B}\mathrm{u}(k), \tag{2.65}$$

The receding horizon idea consists of predicting a process response for N future time steps and picking the sequence of control inputs which minimizes a specified cost. However, only the first control input is actually applied to the plant. The rest are discarded. Over the prediction horizon N, it is thus only interesting to consider possible model-based state trajectories. These are enforced with equality constraints on the decision variables $\mathbf{z}$ according to $\mathcal{A}_{eq}\mathbf{z} = b_{eq}$ in the quadratic program, where $\mathcal{A}_{eq}$ and b_{eq} are the equality constraint matrices. The ordering of the decision variable vector depends on the structure of the constraint matrices. A sequential ordering of states and inputs is chosen.

$$\mathbf{z} = [\mathrm{x}_1,\ \mathrm{x}_2,\ \ldots\ \mathrm{x}_N,\ \mathrm{u}_0,\ \mathrm{u}_1\ \ldots\ \mathrm{u}_{N-1}]^\top \tag{2.66}$$

x_0 is the known initial condition or current state and is thus not a decision variable included in this formulation. u_0 is the optimal control input we wish to find. Over the horizon, the model evolves in the following way:

$$\begin{aligned}
\mathrm{x}_1 &= \mathcal{A}\mathrm{x}_0 + \mathcal{B}\mathrm{u}_0 \\
\mathrm{x}_2 &= \mathcal{A}\mathrm{x}_1 + \mathcal{B}\mathrm{u}_1 \\
\mathrm{x}_3 &= \mathcal{A}\mathrm{x}_2 + \mathcal{B}\mathrm{u}_2 \\
&\cdots \\
\mathrm{x}_N &= \mathcal{A}\mathrm{x}_{N-1} + \mathcal{B}\mathrm{u}_{N-1}
\end{aligned} \tag{2.67}$$

or equivalently:

$$\begin{aligned} \mathrm{x}_1 - \mathcal{B}\mathrm{u}_0 &= \mathcal{A}\mathrm{x}_0 \\ \mathrm{x}_2 - \mathcal{A}\mathrm{x}_1 - \mathcal{B}\mathrm{u}_1 &= 0 \\ \mathrm{x}_3 - \mathcal{A}\mathrm{x}_2 - \mathcal{B}\mathrm{u}_2 &= 0 \\ &\dots \\ \mathrm{x}_N - \mathcal{A}\mathrm{x}_{N-1} - \mathcal{B}\mathrm{u}_{N-1} &= 0 \end{aligned} \tag{2.68}$$

Which is the form that will be used to construct the equality constraint matrices. Note that the first element does not remove $\mathcal{A}\mathrm{x}_0$ from the right column, since it is the initial condition. To make the following matrix more visually intuitive, switch places of x_k and x_{k+1}. Now the states in $-\mathcal{A}\mathrm{x}_k + \mathrm{x}_{k+1} - \mathcal{B}\mathrm{u}_k = 0$ line up with the decision vector $\mathbf{z}$

$$A_{eq} = \begin{bmatrix} I & 0 & \cdots & 0 & -\mathcal{B} & 0 & \cdots & 0 \\ -\mathcal{A} & I & \cdots & 0 & 0 & -\mathcal{B} & \cdots & 0 \\ 0 & -\mathcal{A} & \ddots & 0 & 0 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & I & 0 & 0 & \cdots & -\mathcal{B} \end{bmatrix} \quad b_{eq} = \begin{bmatrix} \mathcal{A}\mathrm{x}_0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.69}$$

and enforces:

$$\mathrm{x}_{k+1} - \mathcal{A}\mathrm{x}_k - \mathcal{B}\mathrm{u}_k = 0, \quad k = 0, \dots, N-1 \tag{2.70}$$

In addition to equality constraints, inequality constraints are included in the problem to describe the physical limitations of the system. These constraints can be formulated as

$$\mathrm{u}_{min} \leq \mathrm{u} \leq \mathrm{u}_{max}, \text{for control signals} \tag{2.71}$$

and

$$\mathrm{x}_{min} \leq \mathrm{x} \leq \mathrm{x}_{max}, \text{for states} \tag{2.72}$$

These are usually rewritten to the matrix form $A_{in}\mathbf{z} \leq b_{in}$, in this case for states, as

$$\underbrace{\begin{bmatrix} I \\ -I \end{bmatrix}}_{A_{in}} \mathrm{x} \leq \underbrace{\begin{bmatrix} \mathrm{x}_{max} \\ \mathrm{x}_{min} \end{bmatrix}}_{b_{in}} \tag{2.73}$$

To pick the optimal sequence of control inputs, a sequence of quadratic programs is solved which minimizes the cost, represented by a cost function, with respect to constraints. The matrices Q and R define the stage cost over the prediction horizon for states x and control input u respectively, appearing as diagonal matrices, which could be viewed as the short term cost, while the terminal cost P_f. represents the cost-to-go beyond the horizon, or the long term cost. The entire cost over the horizon can be expressed in a cost function as

$$\mathcal{V} = \mathrm{x}^\top(N) P_f \mathrm{x}(N) + \sum_{k=0}^{N-1} \left(\mathrm{x}^\top(k) Q \mathrm{x}(k) + \mathrm{u}^\top(k) R \mathrm{u}(k) \right) \tag{2.74}$$

3
Method

Based on the primary research questions, aim and scope, this chapter will present the methods used to model the hydraulic manipulator, generate feasible control references and track those references using constrained MPC. Developing an effective control architecture for a multi-link hydraulic robot means addressing challenges within multiple physical and mathematical domains. The methodology presented here will be structured around system modeling, kinematic mapping and design of a model predictive controller to achieve these goals.

Firstly, the development of a non-linear flow-controlled model is proposed for simulation of the four-link hydraulic robot. To address the need for a reduced linear model, the non-linear model takes part in a sensitivity analysis study where model dynamics are analyzed. A simplified control model is then proposed.

Secondly, the generation of feasible reference trajectories using a constrained inverse kinematic optimization approach is presented. The calculation of cylinder forces given kinematic relationships on the robot is also made to introduce realistic external force disturbance F_{ext} into the simulation.

Last, the motion tracking model predictive controller will be presented, along with a simulation and a description of the closed-loop control scheme. The optimal control problem is presented along with how penalties and constraints are derived.

3.1 Model development

As presented in Section 2.1, the theoretical description of a hydraulic system contains many non-linear and discontinuous relationships, as well as simplified dynamics and unknown parameters. To aid the development of simplified and linear system models, the sensitivity analysis methods presented in Section 2.3. An OAT parameter sweep was performed to refine the model parameters, followed by the Morris method to identify their influence on chosen performance indices. This section aims to describe the non-linear the model that was developed, the sensitivity analysis procedure and the model simplification process in detail.

3.1.1 Non-linear model proposal

A non-linear hybrid model was derived based on the hydraulic system equations presented in Section 2.1. For the purposes of sensitivity analysis, two valve control methods were implemented into this model to compare the dynamic response of an open loop and closed loop valve. The controller variants take the form of a spool position-controlled valve, following open loop spool position references $x_{v,\mathrm{ref}}$, and a closed loop flow-controlled valve using the feed-forward control structure described in Section 2.1.4.

Double-acting cylinder dynamics: Equations 2.1, 2.2, 2.3, 2.9 and 2.10 form the equations of motion for a double acting cylinder using simplified friction and drag models as expressed in Equations 2.5 and 2.4.

$$\dot{x}_c = v_c, \tag{3.1}$$

$$\dot{v}_c = \frac{P_{C1}A_{C1} - P_{C2}A_{C2} - F_c \tanh(k_{\tanh} v_c) - c_v v_c - F_{\mathrm{ext}}}{m_p}, \tag{3.2}$$

$$\dot{P}_{C1} = \frac{\beta}{V_0 + A_{C1}x_c}(Q_{C1} - c_l(P_{C1} - P_{C2}) - A_{C1}v_c)\,, \tag{3.3}$$

$$\dot{P}_{C2} = \frac{\beta}{V_0 + A_{C2}(x_{c,\max} - x_c)}(Q_2 + c_l(P_{C1} - P_{C2}) + A_{C2}v_c) \tag{3.4}$$

where $k_{\tanh} = 10^3$ is a numerical smoothing constant as described in Section 2.1.2.

Valve flow dynamics: As presented in Section 2.1.4, Equation 2.12 and 2.13 define the orifice flow through a four-way proportional directional valve. The spool position x_v responsible for opening and closing the valve assumes the first-order dynamics presented in Equation 2.14. The

$$\dot{x}_v = \frac{x_{v,\mathrm{ref}} - x_v}{\tau_v}, \tag{3.5}$$

$$A_v(x_v) = w_a\,|x_v|, \tag{3.6}$$

$$Q_{C1} = \begin{cases} c_d\,A_v(x_v)\sqrt{\dfrac{2}{\rho}\,|P_s - P_{C1}|}\,\mathrm{sign}(P_s - P_{C1}) & x_v \geq 0, \\ -c_d\,A_v(x_v)\sqrt{\dfrac{2}{\rho}\,|P_{C1} - P_t|}\,\mathrm{sign}(P_{C1} - P_t) & \text{otherwise} \end{cases} \tag{3.7}$$

$$Q_{C2} = \begin{cases} -A_r\,c_d\,A_v(x_v)\sqrt{\dfrac{2}{\rho}\,|P_{C2} - P_t|}\,\mathrm{sign}(P_{C2} - P_t) & x_v \geq 0, \\ A_r\,c_d\,A_v(x_v)\sqrt{\dfrac{2}{\rho}\,|P_s - P_{C2}|}\,\mathrm{sign}(P_s - P_{C2}) & \text{otherwise} \end{cases} \tag{3.8}$$

In the feed-forward flow-controlled valve, the flow reference $Q_{v,\mathrm{ref}}$ using Equation 2.15 to obtain a spool position reference $x_{v,\mathrm{ref}}$, after which the spool dynamics and flow equations are identical to the position-controlled case (Equations 3.6-3.8).

$$\Delta P = \begin{cases} \max\Big(P_s - P_{C1},\ \varepsilon\Big) & x_v \geq 0, \\ \max\Big(P_s - P_{C2},\ \varepsilon\Big) & x_v < 0, \end{cases} \tag{3.9}$$

$$x_{v,\mathrm{ref}} = \mathrm{sat}\left(\begin{cases} \dfrac{Q_{\mathrm{v,ref}}}{c_d\, w_a \sqrt{\dfrac{2}{\rho}\,\Delta P}} & x_v \geq 0, \\ \dfrac{Q_{\mathrm{v,ref}}}{A_r c_d\, w_a \sqrt{\dfrac{2}{\rho}\,\Delta P}} & x_v < 0, \end{cases} \right) \tag{3.10}$$

ΔP is the pressure drop over the valve depending on which cylinder chamber is being measured by the flow controller. A small constant $\varepsilon = 10^{-6}$ is present to avoid division by zero. The valve area function $A_v(x_v)$ in Equation 3.6 describes a rectangular valve where the area gradient w_a is simply the width of the orifice opening [3]. While there exists many types of orifice shapes [3], a rectangular valve was chosen for ease of calculation in the flow-controlled case because of the feed-forward scheme. w_a and c_d are assumed to be constant parameters of the valve, since no further specification could be provided in this regard. Likewise, the orifice area ratio was set to $A_r = A_{\mathrm{C2}}/A_{\mathrm{C1}}$ for simplification of the system going forward.

Pump equations: The pump pressure and flow dynamics are modeled in Section 2.1.5, with pump pressure P_s described by the continuity equation 2.16 and flow Q_p is governed by the angle of the swash-plate in Equation 2.18. As discussed, a complete swash-plate model requires system-specific geometric parameters that are generally unavailable from datasheets. As disassembly of the pump would be required to gain more information, a simplified representation of swash-plate dynamics had to be used in this study. The simplified model is based on normalization of swash-plate angle α_{SP} dependent on the load-sensing line

$$\dot{\alpha}_{\mathrm{SP}} = \frac{1}{\tau_{\mathrm{LS}}} \frac{P_{\mathrm{LS}} + \Delta P_{\mathrm{LS}} - P_s}{\Delta P_{\mathrm{LS}}} \tag{3.11}$$

$$P_{\mathrm{LS}} = \max(P_{\mathrm{C1}}, P_{\mathrm{C2}}) \tag{3.12}$$

where τ_{LS} is the swash-plate time-constant, P_{LS} is the sensed pressure from the load and ΔP_{LS} is the desired orifice pressure drop. For a single cylinder, P_{LS} is the maximum of either chamber pressure [10]. This model captures the essential load-sensing behavior, where the swash-plate adjusts displacement to maintain a a desired pressure differential. τ_{LS} represents the response time of the swash-plate without requiring knowledge of detailed pump geometry. However, by reducing the dynamics to a single time-constant, higher order mechanical and hydraulic interactions within the pump are not represented.

The pump continuity equation and pump flow model, Equation 2.16 and 2.18 respectively, are now completed with swash-plate dynamics and load-sensing line according to Equation 3.11 and 3.12. The flow consumption described by Equation 2.17 finishes the description of the variable displacement pump dynamics

$$Q_p = V_d\,\omega_s \alpha_{SP}, \tag{3.13}$$

$$P_{LS} = \max(P_{C1},\, P_{C2}), \tag{3.14}$$

$$Q_{\text{sys}} = \max(Q_{C1},\, 0) + \max(Q_{C2},\, 0) \tag{3.15}$$

$$\dot{\alpha}_{SP} = \begin{cases} 0 & \text{if } \Big(\alpha_{SP} \leq \alpha_{\min} \wedge P_{LS} + \Delta P_{LS} < P_s\Big) \\ & \quad \text{or } \Big(\alpha_{SP} \geq 1 \wedge P_{LS} + \Delta P_{LS} > P_s\Big), \\ \frac{1}{\tau_{\text{LS}}} \frac{P_{\text{LS}} + \Delta P_{\text{LS}} - P_s}{\Delta P_{\text{LS}}} & \text{otherwise} \end{cases} \tag{3.16}$$

$$\dot{P}_s = \begin{cases} 0 & \text{if } \Big(P_s \geq P_{s,\max} \wedge Q_p > Q_{\text{sys}} + c_s P_s\Big) \\ & \quad \text{or } \Big(P_s \leq P_t \wedge Q_p < Q_{sys} + c_s P_s\Big), \\ \dfrac{\beta}{V_s}(Q_p - Q_{sys} - c_s P_s) & \text{otherwise} \end{cases} \tag{3.17}$$

3.1.2 Parameter range evaluation using OAT analysis

While the non-linear hybrid models presented in equations 3.1 through 3.10 are representative of the dynamics present in the hydraulic system, there are many parameters which are unknown or have varying degree of uncertainty regarding the ranges of plausible values these can assume. In particular, value ranges for cylinder leakage c_l, pump leakage c_s and viscous damping c_v could not be found in literature or component datasheets. The effect of viscous damping c_v, Coulomb friction F_c and external force F_{ext} on the system was also of interest, since these directly oppose the motion of the piston and result in the pressure affecting other components in the system. The choice of pressure margin ΔP_{LS} and load-sense time-constant τ_{LS} affect how quickly pump responds to cylinder pressure changes in the system.

A sensitivity analysis was therefore performed on the model beginning with a One-at-a-time parameter sweep as described in Section 2.3. This OAT analysis was not intended to rank parameter importance, but only to establish plausible parameter value ranges for use in the Morris analysis. The chosen simulation case was open-loop piston extension, where the effect on extension length and pressure was observed. This case was tested separately for both spool position references $x_{v,\text{ref}}$ and flow references $Q_{v,\text{ref}}$ which were then compared to evaluate the validity of parameter values. A parameter's value was determined to be plausible if the absolute cylinder and pump pressure trajectory remained within $[0, P_{s,\max}]$, where $P_{s,\max}$ is the maximum pump supply pressure. The following spool position reference was used

Valve spool position reference:

$$x_{v,\text{ref}}(t) = \begin{cases} 0.5 \cdot x_{v,\max} & 0.5 \leq t < 1.0 \\ 0 & \text{otherwise} \end{cases} \quad [\text{mm}] \tag{3.18}$$

where $x_{v,\max}$ is the maximum allowed valve spool opening length. The corresponding flow reference was calculated from this spool position reference using the orifice equation and the nominal load-sense margin pressure

$$Q_{v,\text{ref}}(t) = x_{v,\text{ref}}(t) c_d \, w_a \sqrt{\frac{2\,\Delta P_{\text{LS}}}{\rho}} \tag{3.19}$$

where $\Delta P_{\text{LS}} = 20$ bar is a typical value for the pressure margin [3] [10]. The non-linear model was simulated using `ode15s` in MATLAB because hydraulic systems tend to exhibit stiff dynamics [3], and the values presented in Table 3.1 were found. The parameter range is defined as the smallest and largest value for each parameter, but only the discrete values in the range were simulated and plotted in the OAT analysis

Table 3.1: OAT sensitivity analysis parameter values

Symbol	Description	Unit	Values	Log-scale
τ_{v}	Valve time constant	s	0.05, 0.10, 0.15, 0.20	No
τ_{LS}	Load-sense time constant	s	0.05, 0.20, 0.35, 0.60	No
c_{v}	Viscous damping	Ns/m	1, 10, 100, 1000	Yes
F_{c}	Coulomb friction	N	1, 10, 100, 1000	Yes
c_{s}	Pump leakage	$\text{m}^3/(\text{Pa} \cdot \text{s})$	10^{-15}, 10^{-14}, 10^{-13}, 10^{-12}, 10^{-11}	Yes
c_{l}	Cylinder leakage	$\text{m}^3/(\text{Pa} \cdot \text{s})$	10^{-15}, 10^{-14}, 10^{-13}, 10^{-12}, 10^{-11}	Yes
ΔP_{LS}	LS margin pressure	bar	10, 15, 20, 25, 30	No
m_p	Piston mass	kg	1, 10, 100	Yes
β	Fluid bulk modulus	Pa	0.5×10^9, 1.0×10^9, 1.5×10^9	No
F_{ext}	External force	Pa	1, 10, 100, 1000, 10000	Yes

As defined in Table 3.1, many parameters span multiple orders of magnitude to capture a larger parameter uncertainty and are thus scaled logarithmically. The table includes the external force F_{ext}, which is not considered a parameter but used to change the load case of the simulation. It is however considered a swept parameter in these simulations. Less uncertain parameters follow a linear scaling in their parameter range. The parameters not part of the parameter sweep were kept at the constant values defined in Table 3.2 below. Most of these parameters are related to geometry, configuration or specification of system components and were therefore assumed to be known.

Table 3.2: Simulation constants

Symbol	Description	Default value	Unit
ρ	Hydraulic fluid density	850	$\mathrm{kg\,m^{-3}}$
L_{hose}	Cylinder hose length	1.780	m
$L_{\text{pump\_hose}}$	Pump hose length	0.515	m
c_{dead}	Cylinder hose dead volume	5.463×10^{-5}	$\mathrm{m^3}$
s_{dead}	Pump hose dead volume	6.321×10^{-5}	$\mathrm{m^3}$
$x_{c,\max}$	Maximum stroke length of the piston	0.35	m
A_{C1}	Piston (cap-end) area	1.963×10^{-3}	$\mathrm{m^2}$
A_{C2}	Piston (rod-end) area	1.348×10^{-3}	$\mathrm{m^2}$
V_0	Constant dead volume of cylinder chambers	7.720×10^{-5}	$\mathrm{m^3}$
c_d	Valve discharge coefficient	0.67	—
w_a	Valve area gradient	2.5×10^{-3}	m
$x_{v,\max}$	Maximum valve opening	0.004	m
P_t	Reservoir (tank) pressure	0	Pa
$P_{s,\max}$	Max pump pressure	280×10^5	Pa
$Q_{p,\max}$	Max pump flow rate	5×10^{-4}	$\mathrm{m^3/s}$
ω_s	Angular velocity of pump	157.08	rad/s
V_d	Pump displacement	22×10^{-6}	$\mathrm{m^3/rad}$
V_s	Dead volume of pump	6.321×10^{-5}	$\mathrm{m^3}$

Using Table 3.1, the mean values in each parameter range can be calculated to obtain a nominal description for the system's parameters. The geometric mean was used to calculate the mean of Log-scale parameter ranges. Linearly scaled parameters use the arithmetic mean. The mean parameters were then applied to create a nominal system model. The resulting simulation trajectory is denoted $x_{c,\text{nom}}$ and serves as the reference for RMSE evaluation in the Morris analysis.

3.1.3 Morris sensitivity analysis

The Morris method described in Section 2.3.2 was applied to the parameter ranges established using OAT, Table 3.1. The flow-controlled valve model was used exclusively in these tests as it represents the relevant control method used by the real system. For the analysis, each parameter was normalized to the unit interval $[0, 1]$ prior to sampling. The mapping from normalized coordinate $x_i \in [0, 1]$ to the value p_i follows the scaling defined in Table 3.1. For log-scaled parameters the mapping is

$$p_i = p_{i,\min} \cdot \left(\frac{p_{i,\max}}{p_{i,\min}} \right)^{x_i} \tag{3.20}$$

and for linearly scaled parameters

$$p_i = p_{i,\min} + x_i \left(p_{i,\max} - p_{i,\min} \right) \tag{3.21}$$

This ensures that a uniform step Δ in the normalized space corresponds to the proportional change in value for log scaled parameters. All trajectory construction

and elementary effect calculations are done in the normalized space. The parameter values are recovered only when running the simulations to calculate the elementary effects in each trajectory. Following [21], the normalized parameter space was discretized into a $p = 4$ level grid, with the step size $\Delta = \frac{1}{p-1} = \frac{1}{3}$ giving the grid points $\{0, \frac{1}{3}, \frac{2}{3}, 1\}$. Furthermore, r independent trajectories were computed beginning from a randomly selected starting point where each coordinate is drawn uniformly from the grid levels. At each step within a trajectory, one parameter at a time is perturbed by $\pm\Delta$ in a randomly selected direction and the order in which parameters are perturbed is chosen randomly. The sign of Δ is drawn independently for each parameter and trajectory, and the starting points were restricted so that the perturbed point $x_i - \Delta$ also falls in the interval $[0, 1]$. The elementary effect for parameter i in trajectory m was the computed according to

$$E_i^{(m)} = \frac{f(p_1, \ldots, p_i + \Delta, \ldots, p_k) - f(p_1, \ldots, p_i, \ldots, p_k)}{\Delta} \tag{3.22}$$

Simulations that failed to integrate were excluded from the estimate, and the remaining samples were used to calculate μ_i^* and σ_i using Equations 2.59 and 2.60. Two output metrics f in the elementary effect formulation above was chosen for the simulation. These metrics are rise time and the cumulative RMSE of the output trajectory. Rise time quantifies how quickly the system responds to a change in reference signal, capturing the effective bandwidth of the actuator. RMSE measures the deviation of the piston trajectory compared to the nominal system response, which quantifies both transient errors and steady-state offset. Together, these two metrics expose complementary aspects of system dynamics. Rise time was defined as the time for piston position to travel from 10% to 90% of its range during the extension phase ($0.5 \leq t \leq 2.0\,\mathrm{s}$). The cumulative RMSE was computed as

$$\sqrt{\frac{1}{n_t} \sum_t \left(x_c(t) - x_{c,\mathrm{nom}}(t)\right)^2} \tag{3.23}$$

where $x_{c,\mathrm{nom}}$ is the nominal piston trajectory, and n_t is the number of time steps. The number of morris trajectories was set to $r = 200$ which exceeds the range $r \in [4, 10]$ suggested in [21]. This choice was made to ensure that estimates for μ_i^* had converged. Convergence was considered sufficient when μ^* ceased to change significantly with additional trajectories. The total number of simulation runs required is $N = r(k + 1) = 200 \times 11 = 2200$, where $k = 10$ is the number of parameters, including external force as defined in Table 3.1.

3.2 First-order flow model

This section will describe the derivation of a simplified model based on observations made in the sensitivity analysis. The hydraulic system can be reduced to a first-order model given the following assumptions:

- Hydraulic fluid compression is negligible, meaning small changes in pressure result in negligible volumetric change of the fluid [3].

- The control valve is equipped with an internal pressure compensating flow regulator, meaning that pressure variations in the pump do not significantly affect the flow through the valve. The pump and valve can therefore be treated as decoupled, and ΔP across the valve orifice can assumed approximately constant [3].
- Cylinder leakage is small relative to the valve flow under normal operating conditions and can therefore be neglected.
- External forces, friction and viscous damping are treated as disturbances acting on the system, rather than dynamics to be modeled.

Under these assumptions, the dominant dynamics of the system is the valve response. If fluid is compression, cylinder leakage and external forces are neglected, the piston velocity is entirely determined by the flow supplied by the valve. Velocity is then modeled as

$$\dot{x}_c = \frac{Q_v}{A_c}, \qquad A_c = \begin{cases} A_{C1} & Q_v \geq 0 \\ A_{C1} \cdot A_r & Q_v < 0 \end{cases} \tag{3.24}$$

where x_c is the position of the piston while Q_v and A_c is valve flow and piston area respectively. The cylinder area A_c is a scheduled parameter that can be determined by the sign of the valve flow Q_v. Assuming the ideal area ratio $A_r = A_{C2}/A_{C1}$ simplifies equation 3.24. The exact technical documentation regarding dimensions of the valves used in the four link hydraulic robot could not be provided, but experimental data showed that valve considered in this study has the characteristics of of first-order system with respect to flow references. Similar dynamics is supported by the theory presented in Section 2.1.4. An equivalent valve flow model can then be written as

$$\dot{Q}_v = \frac{Q_{v,\text{ref}} - Q_v}{\tau_v} \tag{3.25}$$

where $Q_{v,\text{ref}}$ is the desired valve flow and τ_v is the valve time-constant. A continuous time state space model with scheduled parameter A_c can then be formulated as

$$\begin{bmatrix} \dot{x}_c \\ \dot{Q}_v \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{A_c} \\ 0 & \frac{-1}{\tau_v} \end{bmatrix} \begin{bmatrix} x_c \\ Q_v \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{\tau_v} \end{bmatrix} Q_{v,ref} \tag{3.26}$$

This model captures the dominant valve dynamics while deliberately neglecting pressure dynamics, fluid compressibility and external forces. The validity of these simplifications was evaluated in comparison with the non-linear model.

3.3 Reference generation

This section will cover the methods that were used to generate reference signals for the controller. A method for kinematic inversion is presented, which is used to convert the task-space inputs, given by the operator, into joint velocities. A cylinder-joint-space map is derived to facilitate total task-to-cylinder velocity conversion which is necessary to generate the reference cylinder movements that are

sent to the MPC. The cylinder-joint-space map is also utilized to convert dynamic torque to equivalent cylinder forces that act as disturbances.

The hydraulic valves have an internal controller that regulates fluid flow based on a flow rate reference. Fluid flow rate in and out of the cylinder is directly proportional to the piston area and more importantly piston velocity, enabling use of traditional inverse Jacobian methods together with a cylinder-joint map to transform task velocity to cylinder velocity.

3.3.1 Inverse kinematics

The objective of the inverse kinematics is to obtain cylinder velocity references as the solution to the inverse kinematic problem. A pseudo-inverse Jacobian algorithm, as described in Section 2.2.4. This type of inverse-Jacobian algorithm was considered well suited for the four-link robot discussed in this thesis due to how the operator controls the robot. When tilting the joysticks, the operator essentially provides a reference task-space velocity vector. The inverse Jacobian directly maps this to joint velocities which in the end translate directly to flow references for the hydraulic control valves.

It is beneficial to reformulate the inverse-Jacobian algorithm into an optimization problem which minimizes $\dot{x} - J\dot{q}$, as shown in Section 2.2.5, to prevent infeasible references being sent to the MPC. As explained in Section 2.2.2, a transformation matrix ξ is used to change the decision variables in the quadratic program from $\dot{q}$ to $\dot{L}$ by substituting $\dot{q} = \xi\dot{L}$ into Equation 2.33. One can verify that the variable change results in the cylinder-space cost $H = \bar{J}^\top\bar{J}$ and $f = -2\bar{J}^\top\dot{x}$, where $\bar{J}$ is $J\xi$. The constraints for this optimization problem consists of limits on position and velocity in both cartesian and cylinder space. Since the decision variable is $\dot{L}$, all position constraints have to be formulated as velocity constraints. By using a finite difference approach it is possible to calculate a maximum and minimum allowable velocity and position in both cylinder and cartesian space, given the current position and the time step h of the program. *cart* and *cyl* refer to the different spaces and the subscript v distinguishes velocity constraints.

$$\begin{aligned} A_{cart} &= \begin{bmatrix} \bar{J} \\ -\bar{J} \end{bmatrix}, \quad b_{cart} = \begin{bmatrix} \frac{x_{max}-x}{h} \\ \frac{x-x_{min}}{h} \end{bmatrix} \\ A_{cyl} &= \begin{bmatrix} I_{nq} \\ -I_{nq} \end{bmatrix}, \quad b_{cyl} = \begin{bmatrix} \frac{L_{max}-L}{h} \\ \frac{L-L_{min}}{h} \end{bmatrix} \end{aligned} \tag{3.27}$$

$$\begin{aligned} A_{v,cart} &= \begin{bmatrix} \bar{J} \\ -\bar{J} \end{bmatrix}, \quad b_{v,cart} = \begin{bmatrix} \dot{x}_{max} \\ \dot{x}_{max} \end{bmatrix} \\ A_{v,cyl} &= \begin{bmatrix} I_{nq} \\ -I_{nq} \end{bmatrix}, \quad b_{v,cyl} = \begin{bmatrix} \dot{L}_{max} \\ \dot{L}_{max} \end{bmatrix} \end{aligned} \tag{3.28}$$

Note that minimum velocity is constrained as the negative maximum velocity. Concatenation of the inequality matrices and vectors result in the total inequality con-

straint matrix and vector:

$$A_{in} = \begin{bmatrix} A_{cart} \\ A_{v,cart} \\ A_{cyl} \\ A_{v,cyl} \end{bmatrix}, \quad b_{in} = \begin{bmatrix} b_{cart} \\ b_{v,cart} \\ b_{cyl} \\ b_{v,cyl} \end{bmatrix} \tag{3.29}$$

As in Section 2.2.5, the quadratic program with only the primary task tracking objective is formulated as

$$\begin{aligned} \min_{\dot{L}} \quad & \mathcal{V}(\dot{x}, \dot{L}) = \dot{L}^\top \bar{J}^\top \bar{J} \dot{L} - 2\dot{L}^\top \bar{J}^\top \dot{x} \\ \text{s.t} \quad & A_{in}\dot{L} \leq b_{in} \end{aligned} \tag{3.30}$$

However, to add null-space contribution according to Section 2.2.6 the quadratic program has to modified with contributions to both the quadratic and linear terms as

$$\begin{aligned} \min_{\dot{L}} \quad & \mathcal{V}(\dot{x}, \dot{L}) = \dot{L}^\top \bar{J}^\top \bar{J} \dot{L} + \dot{L}^\top (I - \bar{J}^\dagger \bar{J})^\top \mathbf{W} (I - \bar{J}^\dagger \bar{J}) \dot{L} - 2\dot{L}^\top \bar{J}^\top \dot{x} + \dot{L}^\top (I - \bar{J}^\dagger \bar{J})^\top \dot{L}_* \\ \text{s.t} \quad & A_{in}\dot{L} \leq b_{in} \end{aligned} \tag{3.31}$$

where $\dot{L}_*$ is $2\mathbf{W}(L - L_{desired})$ and $L_{desired}$ is the cylinder length configuration from which deviation is penalized. $\mathbf{W}$ is a diagonal weight matrix containing the individual penalties for each cylinder, where higher numbers represent higher penalties.

3.3.1.1 Closed loop inverse kinematics

As the inverse kinematics algorithm has been presented so far, it is an open loop and thus vulnerable to drift. Errors arise mainly due to external forces and linearization and integration drift related to calculating the Jacobian and then integrating the resulting joint velocity to attain a joint angle reference. The Jacobian is calculated once during each iteration, essentially linearizing the Jacobian, but is only valid for one exact joint configuration. As soon as the robot moves away from the exact configuration for which it was calculated, error is introduced. The same principle applies for integrating reference velocity to reference angles for the MPC horizon. If the actual joint velocity during an iteration varies from the exact reference velocity, which is almost always the case, integration error is introduced [23].

A relatively simple way of making the inverse kinematics algorithm more robust is to add a task-space feedback [16]. For example, a PI-feedback can be introduced as

$$\dot{q}_{ref} = J(\dot{x}_{ref} + K_p e + K_i \int e) \tag{3.32}$$

where e is the task-space positional error $x_{ref} - x$ and K_p and K_i are the proportional and integral gains.

Other ways of countering drift include using more complex integration methods such as Runge-Kutta 2 or 4 [23]. But this did not seem necessary since closing the loop proved to be an adequate solution, which can be seen in the results.

3.3.2 Cylinder-joint kinematics

This section presents how to map cylinder and joint velocities. The manipulator Jacobian is derived from homogeneous transformations and maps joint velocities and task velocities. But to calculate the flow references, cylinder velocities are needed. As such, another map between the cylinder velocities and joint velocities is required. Both the manipulator Jacobian and the cylinder-joint map can then be stacked to map cylinder-space to task-space. The cylinder-joint map is calculated by studying the geometry of the linkages and how each cylinder actuates the joint.

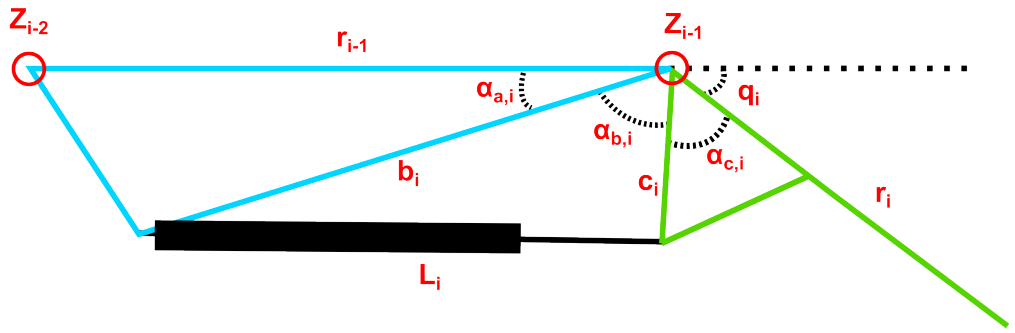


Figure 3.1: General description of cylinder-joint geometry with counter-clockwise extension actuation [1]

The main governing equation of this transformation is the Cosine theorem, with variables and geometry as in Figure 3.1, and certain modifications depending on which way the cylinder actuates the joint. The Cylinder length (black) is treated as a variable side in the triangle that is formed by L_i,b_i and c_i and is related to joint angle in the following way [12] [1] [24]

$$L_i = \sqrt{b_i^2 + c_i^2 - 2b_i \cos(\alpha_{b,i})} \tag{3.33}$$

For cylinders acting on the joint with clockwise extension actuation, joint angle q_i is related to cylinder length L_i in the following way

$$q_i = \alpha_{a,i} + \alpha_{b,i} + \alpha_{c,i} - \pi \tag{3.34}$$

where $\alpha_{b,i}$ is the angle that varies with the cylinder length and $\alpha_{a,i}$ and $\alpha_{c,i}$ are fixed angles. $\alpha_{b,i}(L)$ is always calculated as

$$\alpha_{b,i} = \arccos\left(\frac{b_i^2 + c_i^2 - L_i^2}{2b_i c_i}\right) \tag{3.35}$$

For cylinders with counter clockwise extension actuation, as in Figure 3.1, joint angle is related to cylinder length as

$$q_i = \pi - \alpha_{a,i} - \alpha_{b,i} - \alpha_{c,i} \tag{3.36}$$

The cylinder-space to joint-space Jacobian can be derived by analytical derivation of the kinematic equations with respect to cylinder length. For cylinders with counter-clockwise extension actuation the following transform is derived

$$\frac{\partial q_i}{\partial L_i} = \frac{2L_i}{\sqrt{4b_i^2c_i^2 - (b_i^2 + c_i^2 - L_i^2)^2}} \tag{3.37}$$

and for clockwise actuation the relation simply becomes negative.

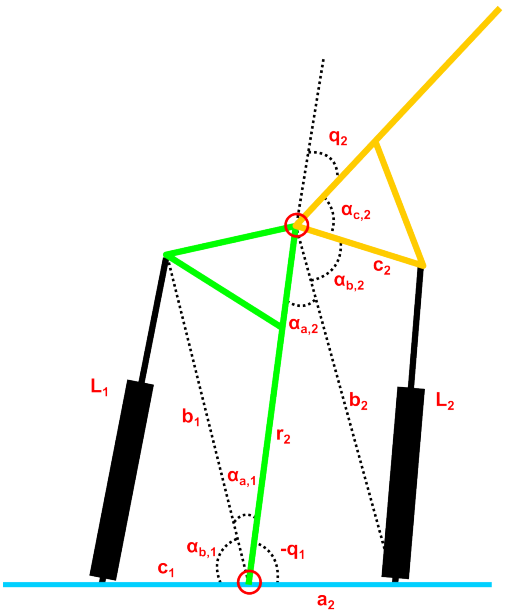


Figure 3.2: Description of cylinder-joint geometry for joint 1 and 2 with coupling. Note that q_1 is negative for joint angles above the horizontal plane

If the second cylinder is fastened to the robot base, there will be a coupling between the first and second cylinder as in Figure 3.2 , where b_2 and $\alpha_{a,2}$ becomes a function off q_1, which is a function of L_1. b_2 and $\alpha_{a,2}$ can then be calculated in the following way, where r_2 is the link length between frame 2 and 1 and a_2 is the side opposite to the, in this case, variable angle $\alpha_{a,2}$

$$b_2 = \sqrt{a_2^2 + r_2^2 - 2a_2r_2\cos(-q_1)} \tag{3.38}$$

$$\alpha_{a,2} = \arcsin\left(\frac{a_2\sin(-q_1)}{b_2}\right) \tag{3.39}$$

The partial derivative is calculated based on Equation 3.34 and is most easily derived symbolically in MATLAB. The derivative with respect to L_1 is especially tough since

the dependence of L_1 is buried in several nested derivative chains. Both $\alpha_{b,2}$ and $\alpha_{a,2}$ are trigonometric functions dependent on b_2, which is itself a trigonometric function dependent on L_1. $\alpha_{a,2}$ is also directly dependent on L_1. This results in a quite complicated expression.

$$\frac{\partial q_2}{\partial L_1} = \frac{d\alpha_{a,2}}{dL_1} + \frac{d\alpha_{b,2}}{dL_1} = \frac{d\alpha_{b,2}}{db_2}\frac{db_2}{dq_1}\frac{dq_1}{dL_1} + \frac{d\alpha_{a,2}}{dq_1}\frac{dq_1}{dL_1} + \frac{d\alpha_{a,2}}{db_2}\frac{db_2}{dq_1}\frac{dq_1}{dL_1} \tag{3.40}$$

$$\frac{\partial q_2}{\partial L_1} = \frac{\frac{2L_1 a_2 \cos(q_1)}{\sqrt{4{b_1}^2{c_1}^2-\left(-{L_1}^2+{b_1}^2+{c_1}^2\right)^2}} - \frac{2L_1 {a_2}^2 r_2 \sin(q_1)^2}{{b_2}^2\sqrt{4{b_1}^2{c_1}^2-\left(-{L_1}^2+{b_1}^2+{c_1}^2\right)^2}}}{\sqrt{{b_2}^2 - {a_2}^2 \sin\left(q_1\right)^2}} + \frac{\frac{4L_1 a_2 r_2 \sin(q_1)}{\sqrt{4{b_1}^2{c_1}^2-\left(-{L_1}^2+{b_1}^2+{c_1}^2\right)^2}} - \frac{2L_1 a_2 r_2 \sin(q_1)\left(-{L_2}^2+{a_2}^2-2\cos(q_1)a_2 r_2+{c_2}^2+{r_2}^2\right)}{{b_2}^2\sqrt{4{b_1}^2{c_1}^2-\left(-{L_1}^2+{b_1}^2+{c_1}^2\right)^2}}}{\sqrt{4{b_2}^2{c_2}^2 - \left(-{L_2}^2 + {a_2}^2 - 2\cos\left(q_1\right)a_2 r_2 + {c_2}^2 + {r_2}^2\right)^2}} \tag{3.41}$$

To simplify, b_2 and q_1 are evaluated as functions of L_1 and inserted into the expression. $\frac{\partial q_2}{\partial L_2}$ is calculated with Equation 3.37, but b_2 still makes it a function of L_1 in the case of coupling.

The end-effector cylinder is also a special case due to the tool-linkage [25], which can be seen in Figure 3.3.

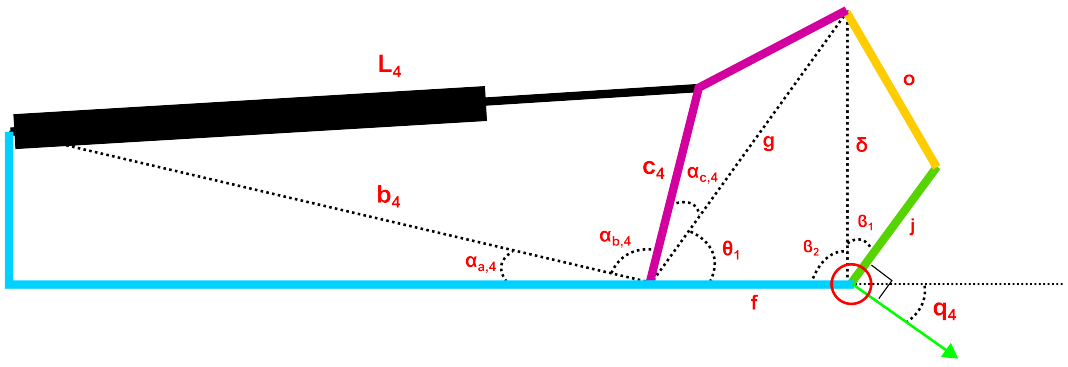


Figure 3.3: Description of cylinder-joint geometry for joint 4 with special linkage for the EE [1]

Joint 4 is related to cylinder 4 in the following way using Equation 3.36 and the Cosine theorem with respect to the geometry in Figure 3.3 where θ_1, β_1 and β_2 are

intermediary angles related to the end-effector. θ_1 is used to calculate the distance δ which is used to calculate β_1 and β_2 which are in turn used to calculate the joint angle q_4. f,j,o and g are sides in the linkage.

$$\theta_1 = \pi - \alpha_{a,4} - \alpha_{b,4} - \alpha_{c,4} \tag{3.42}$$

$$\delta = \sqrt{f^2 + g^2 - 2fg\cos(\theta_1)} \tag{3.43}$$

$$\beta_1 = \arccos\left(\frac{g^2 - f^2 - \delta^2}{-2hf}\right) \tag{3.44}$$

$$\beta_2 = \arccos\left(\frac{o^2 - j^2 - \delta^2}{-2hj}\right) \tag{3.45}$$

$$q_4 = \beta_1 + \beta_2 - \frac{\pi}{2} \tag{3.46}$$

The cylinder-joint transform for joint 4 is derived in a similar manner to Equation 3.41 as

$$\frac{\partial q_4}{\partial L_4} = -\frac{1}{\delta^2}\left(\frac{f\cos\beta_1 - \delta}{f\sin\beta_1} + \frac{j\cos\beta_2 - \delta}{j\sin\beta_2}\right)\left(\frac{fg\sin\theta_1 \cdot L_4}{bc\sin\alpha_{b,4}}\right) \tag{3.47}$$

A complete cylinder-task-space Jacobian $\bar{J}$ can now be assembled to map cylinder velocity to task velocity. The cylinder-joint Jacobian, denoted ξ will be a $n_q \times n_q$ diagonal matrix, since each cylinder typically only actuates one joint. With exception for when first and second joint are coupled, in which case ξ will be diagonal with exception for the second row which will have an extra element in the first column.

$$\bar{J} = J\xi \tag{3.48}$$

3.3.3 Torque to force conversion

The robot dynamics describes the torque applied to each joint given q and $\dot{q}$ which can be used to calculate the external force that acts on each cylinder. The weight of each link is quite small but the tool can weigh a substantial amount and results in considerable force on each cylinder when the arm is extended. Currently these external forces are calculated separately and, as described in Section 3.5, enter into the non-linear model as disturbances since the Euler-Lagrange model is non-linear and thus not compatible with the linear first-order control model.

The external forces are calculated with the following equations, derived in Section 2.2.7. Since energy and power is conserved between joint and cylinder space, one can convert joint torque to a collinear cylinder force.

$$\tau_q^\top \dot{q} = {F_{ext}}^\top \dot{L} \tag{3.49}$$

Inserting the cylinder-joint Jacobian ξ means the joint velocity can be rewritten as

$$\tau_q^\top \xi \dot{L} = F_{ext}^\top \dot{L} \tag{3.50}$$

Cylinder velocity cancels out, resulting in

$$F_{ext} = \xi^\top \tau_q \tag{3.51}$$

Inserting this expression in the Euler-Lagrange equation (2.54), along with angles, angular velocity and angular acceleration in terms of cylinder variables, gives an equation that calculates external cylinder force as a function of cylinder variables.

$$q = f(L) \tag{3.52}$$

$$\dot{q} = \xi \dot{L} \tag{3.53}$$

$$\ddot{q} = \xi \ddot{L} + \dot{\xi} \dot{L} \tag{3.54}$$

$$\underbrace{\xi^\top D(f(L))\xi}_{\tilde{D}(L)} \ddot{L} + \underbrace{\xi^\top \left[D(f(L))\dot{\xi} + C(f(L), \xi\dot{L})\xi \right]}_{\tilde{C}(L,\dot{L})} \dot{L} + \underbrace{\xi^\top g(f(L))}_{\tilde{g}(L)} = F_{ext} \tag{3.55}$$

This equation can now be used to calculate the external force on each cylinder directly with sensed cylinder variables.

$$\tilde{D}(L)\ddot{L} + \tilde{C}(L, \dot{L})\dot{L} + \tilde{g}(L) = F_{ext} \tag{3.56}$$

3.4 Motion tracking

This section covers the implementation of a Model predictive controller that tracks the generated reference signals. MPC has some interesting applications that are useful for robotic manipulators, such as energy optimization [26]. However, the main interest in this thesis is the ability to handle constraints that represent physical limitations of the system. The objective of the MPC is to find control inputs u, such that deviation from reference signals are minimized and no physical constraints (cylinder range of motion and pump and valve flow rates) are violated. The first-order model from Section 3.2 is used as the control model of the MPC and the optimal control problem is formulated as the following.

$$\min_{\mathrm{x}_k, \mathrm{u}_k, \Delta\mathrm{u}_k, \epsilon_k} \quad \sum_{k=0}^{N-1} \mathcal{V}(\mathrm{x}_k, \mathrm{u}_k, \Delta\mathrm{u}_k, \epsilon_k) + \mathcal{V}_N(\mathrm{x}_N, \epsilon_N) \tag{3.57}$$

$$\text{s.t.} \quad \mathrm{x}_{k+1} = \mathcal{A}\mathrm{x}_k + \mathcal{B}\mathrm{u}_k, \quad k = 0, \ldots, N-1 \tag{3.58}$$

$$\mathrm{x}_0 = \hat{\mathrm{x}} \tag{3.59}$$

$$\mathrm{u}_{\min} - \epsilon_k \leq \mathrm{u}_k \leq \mathrm{u}_{\max} + \epsilon_k, \quad k = 0, \ldots, N-1 \tag{3.60}$$

$$\mathrm{x}_{\min} - \epsilon_k \leq \mathrm{x}_k \leq \mathrm{x}_{\max} + \epsilon_k, \quad k = 1, \ldots, N \tag{3.61}$$

$$\epsilon_k, \geq 0 \quad k = 0, \ldots, N \tag{3.62}$$

$$\phi_{k,i} \geq \mathrm{u}_{k,i}, \quad k = 0, \ldots, N-1, \ i = 1, \ldots, m \tag{3.63}$$

$$\phi_{k,i} \geq -\mathrm{u}_{k,i}, \quad k = 0, \ldots, N-1, \ i = 1, \ldots, m \tag{3.64}$$

$$\phi_{k,i} \geq 0, \quad k = 0, \ldots, N-1, \ i = 1, \ldots, m \tag{3.65}$$

$$\sum_{i=1}^{m} \phi_{k,i} \leq Q_{p,max} \quad k = 0, \ldots, N-1, \ i = 1, \ldots, m \tag{3.66}$$

x_k is the states (cylinder length and valve flow rate) and u_k is the input (reference valve flows). x_0 is the initial condition with $\hat{\mathrm{x}}$ representing a measurement of the current state. $\mathcal{A}$ and $\mathcal{B}$ are the discrete state and input matrices of the first-order control model. ϵ and ϕ are slack and auxiliary variables respectively.

3.4.1 Deviation state formulation

Equation 2.74 describes a cost function which drives all states to zero, but that is not always the desired outcome. For setpoint tracking one can introduce deviation states which become null once a certain reference is reached [2]. This property is very useful for motion tracking since state references are supplied by inverting the kinematics. Deviation variables in x are implemented in the following way. Starting with $\mathrm{x}^\top Q\mathrm{x}$, where Q is the quadratic penalty matrix, if we describe deviation variables in x as $\mathrm{x} - \mathrm{x}_{ref}$ we get:

$$(\mathrm{x} - \mathrm{x}_{ref})^\top Q(\mathrm{x} - \mathrm{x}_{ref})$$

Expanding the expression yields:

$$(\mathrm{x} - \mathrm{x}_{ref})^\top Q(\mathrm{x} - \mathrm{x}_{ref}) = \mathrm{x}^\top Q\mathrm{x} - \mathrm{x}^\top Q\mathrm{x}_{ref} - \mathrm{x}_{ref}^\top Q\mathrm{x} - \mathrm{x}_{ref}Q\mathrm{x}_{ref}$$

Since Q is symmetric, $Q = Q^\top$. Transposing any of the two middle terms in the expression yields

$$\mathrm{x}^\top Q\mathrm{x}_{ref} = \mathrm{x}_{ref}^\top Q\mathrm{x}$$

Substituting any of the middle terms for the other results in:

$$\begin{aligned} \mathrm{x}^\top Q\mathrm{x} - \mathrm{x}^\top Q\mathrm{x}_{ref} - \mathrm{x}_{ref}^\top Q\mathrm{x} - \mathrm{x}_{ref}Q\mathrm{x}_{ref} = \\ \mathrm{x}^\top Q\mathrm{x} - 2\mathrm{x}_{ref}^\top Q\mathrm{x} - \mathrm{x}_{ref}Q\mathrm{x}_{ref} = \\ \mathrm{x}^\top Q\mathrm{x} - 2\mathrm{x}^\top Q\mathrm{x}_{ref} - \mathrm{x}_{ref}Q\mathrm{x}_{ref} \end{aligned} \tag{3.67}$$

Where the first term is quadratic in x, and the second term is linear in x. Thus the quadratic term belongs in the Hessian matrix H, and the linear term should be included in f. The last term is a constant and does not affect the optimization problem.

3.4.2 Rate of change

Penalizing rate of change in the control is used for smoothing the control signal and is implemented by adding a move suppression term to the cost function, Equation 2.74, resulting in

$$\mathcal{V} = \mathrm{x}^\top(N)P_f\mathrm{x}(N) + \sum_{k=0}^{N-1}\Big(\mathrm{x}^\top(k)Q\mathrm{x}(k) + \mathrm{u}^\top(k)R\mathrm{u}(k) + \Delta\mathrm{u}(k)^\top S\Delta\mathrm{u}(k)\Big) \tag{3.68}$$

where $\Delta\mathrm{u}(k) = \mathrm{u}(k) - \mathrm{u}(k-1)$ and S is the diagonal cost matrix for the rate of change. Expanding the term $\Delta\mathrm{u}(k)^\top S\Delta\mathrm{u}(k)$ similarly to Equation 3.67, it can be seen that it will contribute a quadratic term $2\Theta^\top S\Theta$ and a linear term $-2\Theta^\top S\Gamma u_{prev}$,

where Θ is a difference matrix on the following form, given a system with one input variable. It becomes clear when looking at the rows of Θ and its transpose that inputs which differ from either its positive or negative neighbor are penalized.

$$\Theta = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \in \mathbb{R}^{Nm \times Nm} \tag{3.69}$$

Γ is a selector matrix that selects the previous control input u_{prev} for the first iteration of each horizon, taking the following form

$$\Gamma = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{Nm \times m} \tag{3.70}$$

3.4.3 Slack variables

Slack variables relax the inequality constraints when the solver otherwise would not have found a solution to the quadratic program [2]. This adds another layer of robustness as it ensures there always exists a solution. It is especially useful when constraining multiple domains such as cylinder-space and cartesian space as in Section 2.2.5. Slack variables are extra decision variables that are introduced into the inequality constraints as

$$\begin{bmatrix} A_{in} & -I \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \epsilon \end{bmatrix} \leq b_{in} \tag{3.71}$$

$$\epsilon \geq 0 \tag{3.72}$$

where the identity term adds one slack decision variable ϵ to each inequality constraint. This allows ϵ to absorb any constraint violation, though at a typically high cost as violation is still undesirable. The slack is then penalized by adding $\epsilon^\top \Lambda \epsilon$ to the quadratic term of the cost function.

$$\mathcal{V} = \mathrm{x}^\top(N) P_f \mathrm{x}(N) + \sum_{k=0}^{N-1} \left(\mathrm{x}^\top(k) Q \mathrm{x}(k) + \mathrm{u}^\top(k) R \mathrm{u}(k) + \Delta \mathrm{u}(k)^\top S \Delta \mathrm{u}(k) + \epsilon^\top \Lambda \epsilon \right) \tag{3.73}$$

3.4.4 Maximum flow constraint

The hydraulic system of the robot is constrained by the maximum flow output of the pump. If the inverse kinematics generate a too fast movement such that the hydraulic cylinders consume more flow than the pump can output, the robot will

behave unpredictably and invalidate the first-order model. It is therefore necessary to ensure that no movement can consume more hydraulic flow than the pump is able to deliver. This can be done by constraining the combined flow control input, but due to linearity in the first-order model, cylinder retraction is commanded with a negative flow. Thus, the constraint will limit the sum of the absolute values of the control inputs. This type of constraint can be implemented by utilizing auxiliary decision variables ϕ in each iteration k of the horizon [27].

$$\phi_i \geq \mathrm{u}_i \tag{3.74}$$

$$\phi_i \geq -\mathrm{u}_i \tag{3.75}$$

$$\phi_i \geq 0 \tag{3.76}$$

$$\sum_{i=1}^{m} \phi_i \leq Q_{p,max} \tag{3.77}$$

This constraint will essentially approximate the absolute value of the input with ϕ and enforce an upper bound on the total hydraulic flow. ϕ is not included in the objective function since u_i is feasible so long as any ϕ_i exists such that $\sum_{i=1}^{m} \phi_i \leq Q_{p,max}$ is satisfied, meaning that ϕ will always approximate the absolute value when the constraint is active. The value of ϕ_i is otherwise irrelevant.

3.5 Simulation

This chapter covers how the motion control algorithm was tested and evaluated as well as how the results were attained. Regrettably, no physical tests on the actual robot were conducted due to a lack of time.

The results were measured during a simulation of a square two-dimensional trajectory in task space. This is was determined to be an effective test since it tests both extension and retraction of each cylinder and also contains abrupt direction changes which tests how the controller reacts to when different motion vectors are commanded. The final implemented control system is visualized in the block diagram in Figure 3.4.

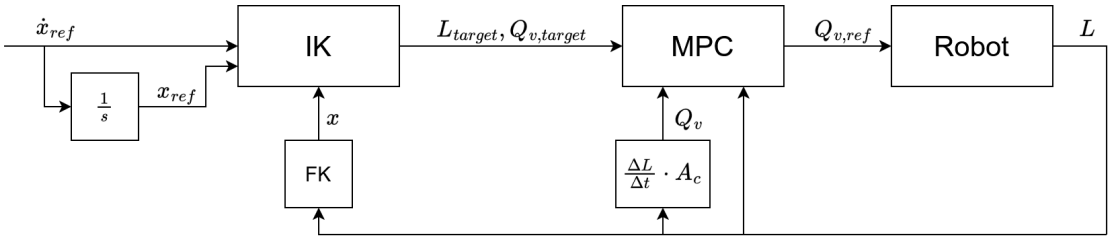


Figure 3.4: Block diagram of control scheme

The simulator script was written in MATLAB with the Robotics System Toolbox and can be summarized in a couple points presented below.

Initialization The DH parameters as well as link masses are loaded to define the robot. Initial conditions are defined including cylinder lengths as well as maximum and minimum values for position and velocity constraints in task- and cylinder-space.

Reference generation Using the operator input velocity vector and reference positions, a cylinder velocity reference is calculated with the pseudo-inverse cylinder-task Jacobian $\bar{J}$. The cylinder velocity references are converted into flow targets by multiplying with the piston area. Since the cylinder length targets vary over the horizon, these have to be calculated for each iteration in the horizon. This is done by integrating the cylinder velocity references over each step of the horizon. Alternatively the inverse kinematic algorithm can be run for each step in the horizon at the cost computational efficiency, since it would add 9 additional optimization problems to solve. Forward kinematics (FK) with the current cylinder configuration also provide a task-space feedback for the CLIK.

To satisfy the different use cases outlined in the introduction, three different control modes can be used by disabling one row of the Jacobian corresponding to pitch or by transforming the operator input to the EE frame:

- Control of all 4 DOFs.
- Control of linear DOFs with pitch free.
- Control of all 4 DOFs in the EE frame.

MPC The first-order model is discretized and measurements of current cylinder lengths are taken. Cylinder velocities, and the corresponding valve flows, are measured by taking the derivative of the cylinder lengths. These measurements are used to formulate equality constraints which are, along with inequality constraints and the generated references, passed to the QP to calculate optimal flow rate inputs. The optimal flow rates, along with the separately calculated external force, increment the plant consisting of the non-linear model in Section 3.1.1.

To simulate real sensor noise, zero mean gaussian noise with standard deviation 0.0002 meters is added to each cylinder length measurement. Zero mean gaussian noise with standard deviation 0.5 bar is also added to the hypothetical pressure sensors in the non-linear plant model

4

Results

This chapter will present the results from the sensitivity analysis that were used to synthesize the reduced first order hydraulic model. Results of the tracking performance of the Model predictive controller will also be presented, as well as the impact of the secondary null-space objective and the closed-loop IK on these results.

4.1 Sensitivity analysis results

The OAT analysis produced many step response trajectories, many of which did not result in significant deviation from the nominal model trajectory. Thus, only step responses with significant trajectory deviation is presented below for position- and flow-controlled spool valves.

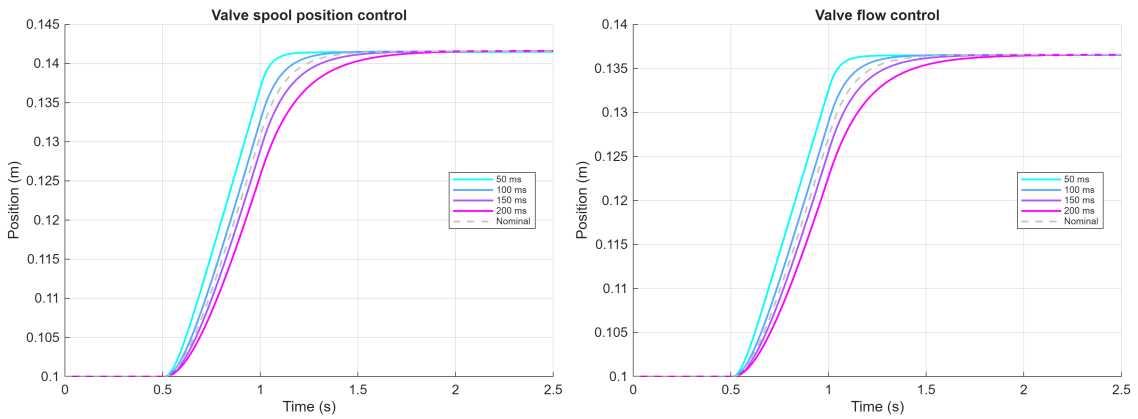


Figure 4.1: OAT analysis piston trajectories: Valve time-constant τ_v

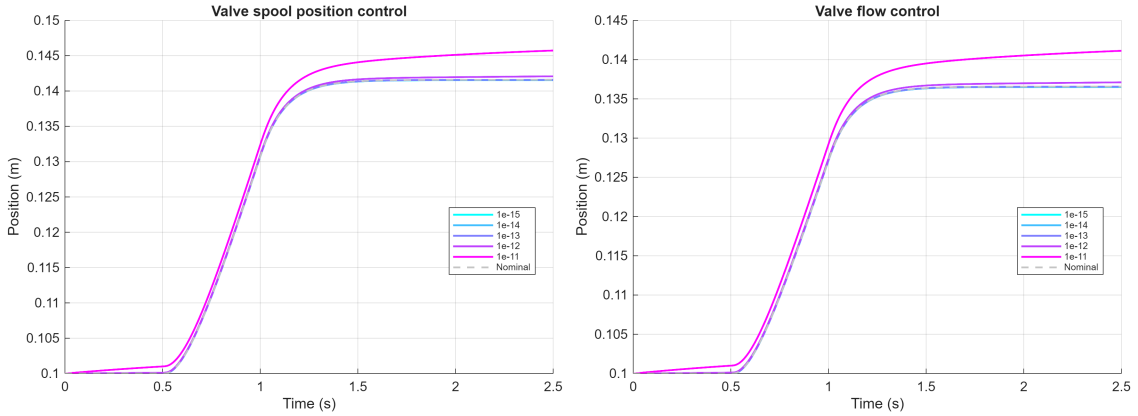


Figure 4.2: OAT analysis piston trajectories: Cylinder leakage c_{l}

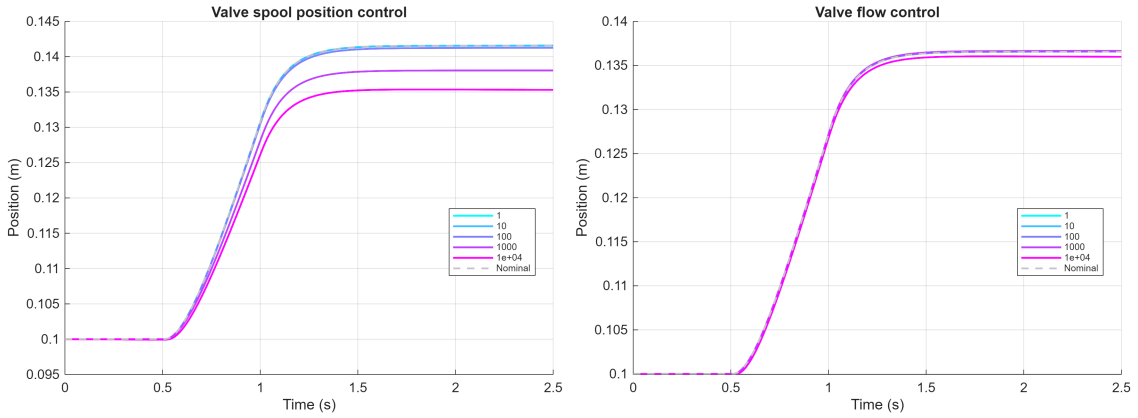


Figure 4.3: OAT analysis piston trajectories: External force F_{ext}

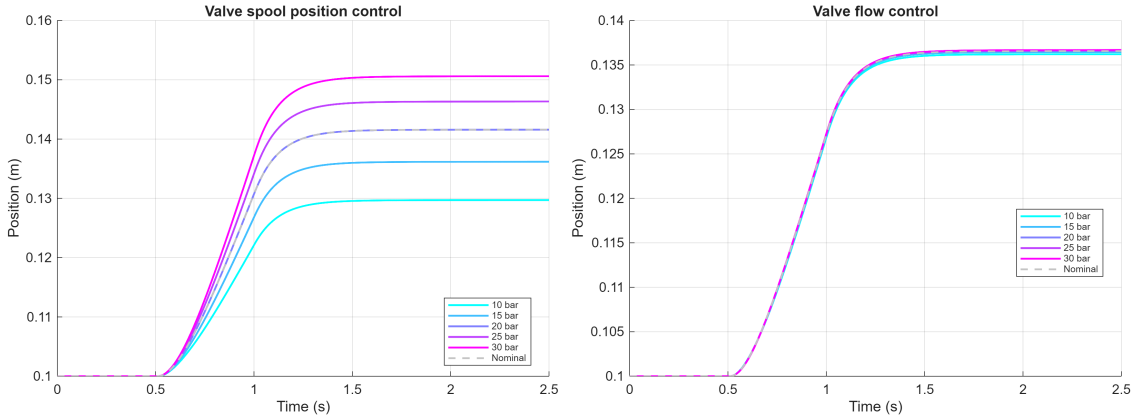


Figure 4.4: OAT analysis piston trajectories: Load-sense margin pressure ΔP_{LS}

The Morris method used parameter ranges defined in Table 3.2, and the following figures show the approximate global effect on rise time and trajectory RMSE system metrics, Figures 4.5 and 4.6. The convergence graphs of these metrics is shown in Figures 4.7 and 4.8 respectively.

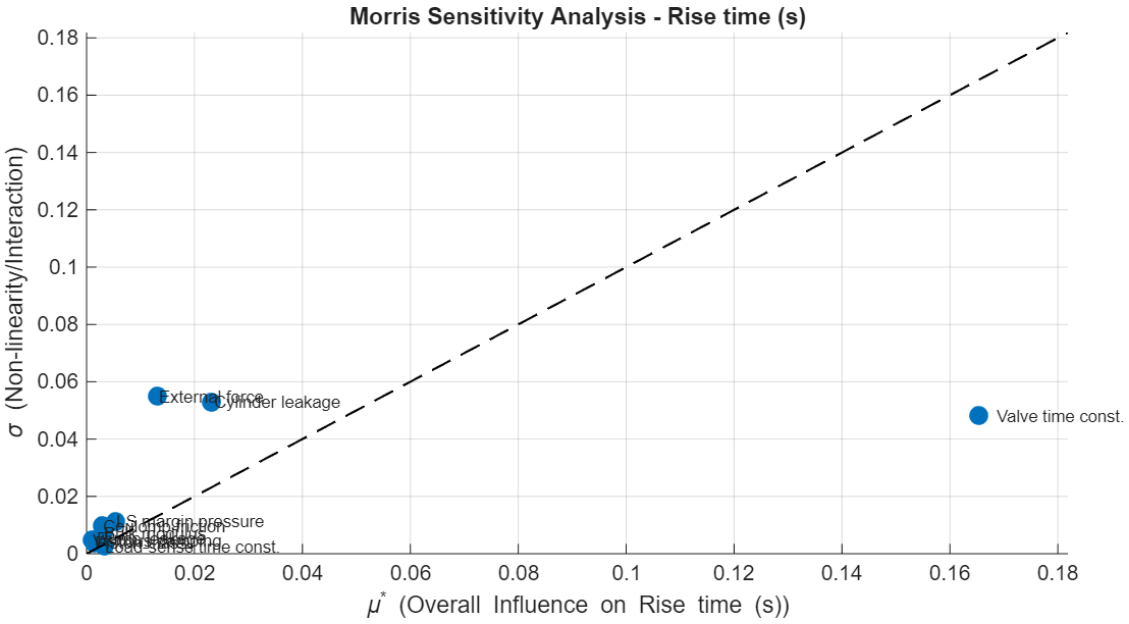


Figure 4.5: Morris analysis: parameter influence on rise time of step response

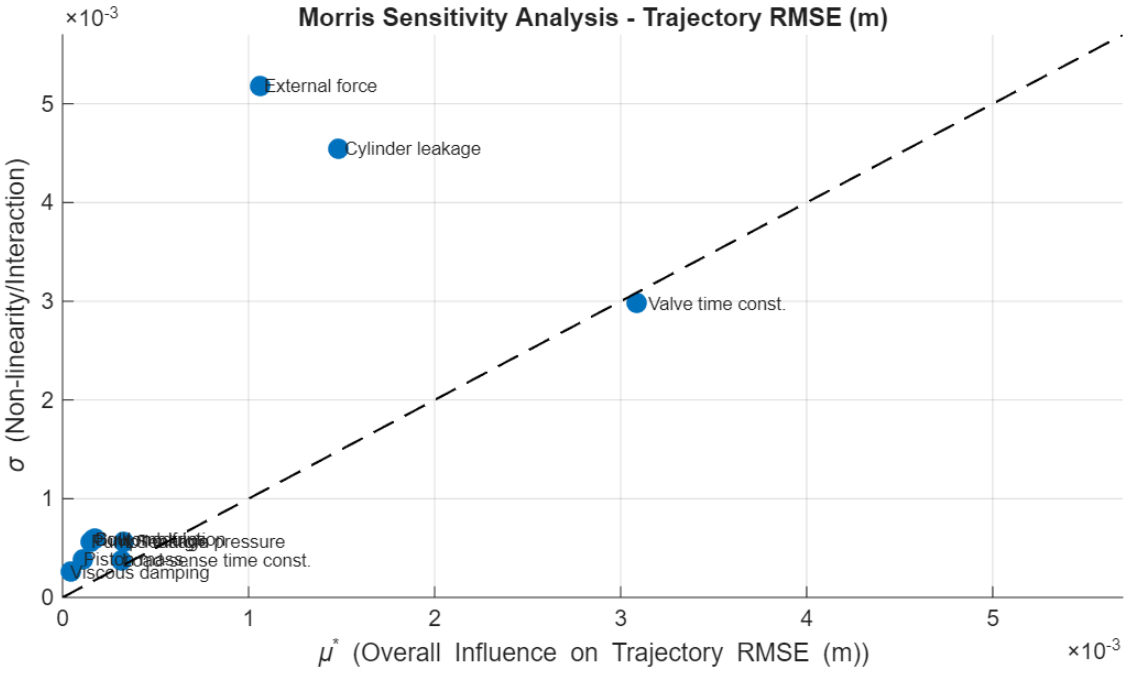


Figure 4.6: Morris analysis: parameter influence on RMSE of step response

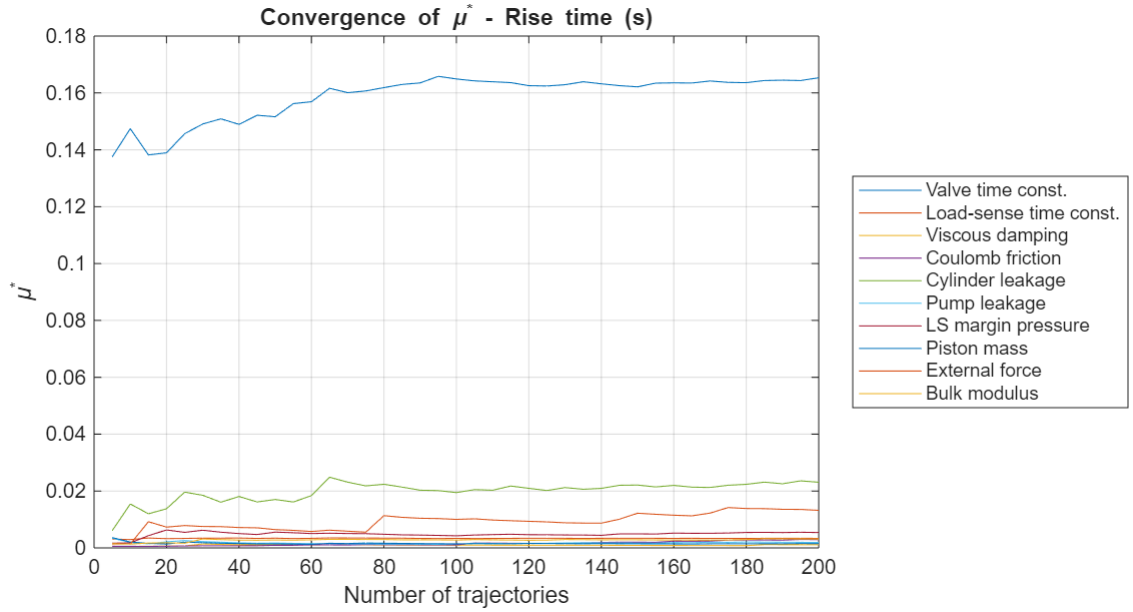


Figure 4.7: Morris analysis: convergence of overall influence $\mu*$ on rise time of step response

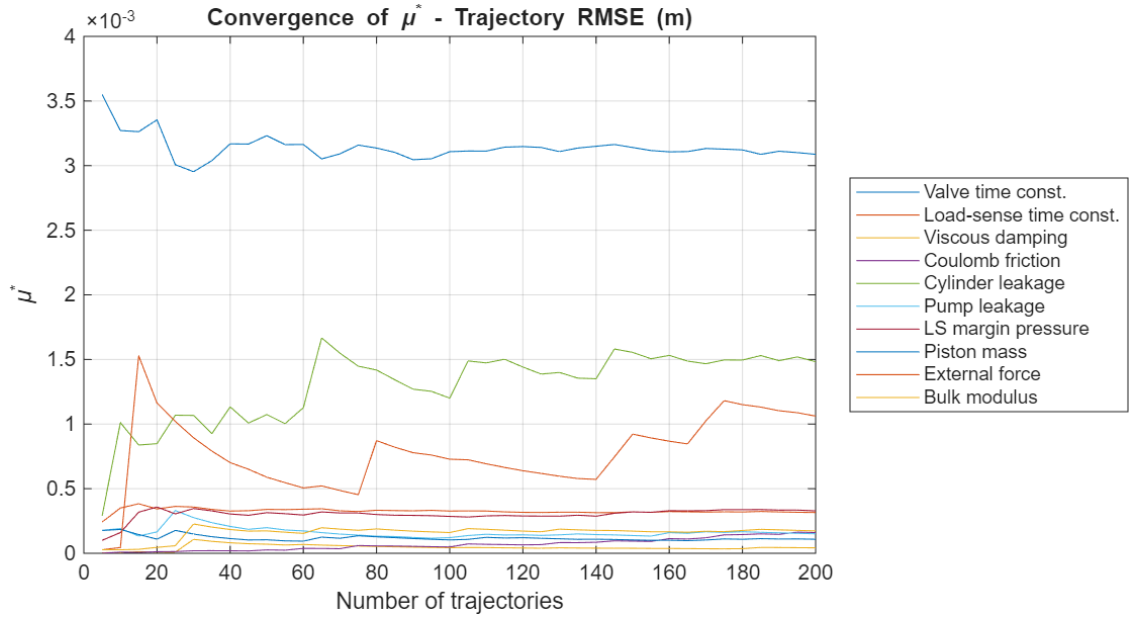


Figure 4.8: Morris analysis: convergence of overall influence $\mu*$ on RMSE of step response

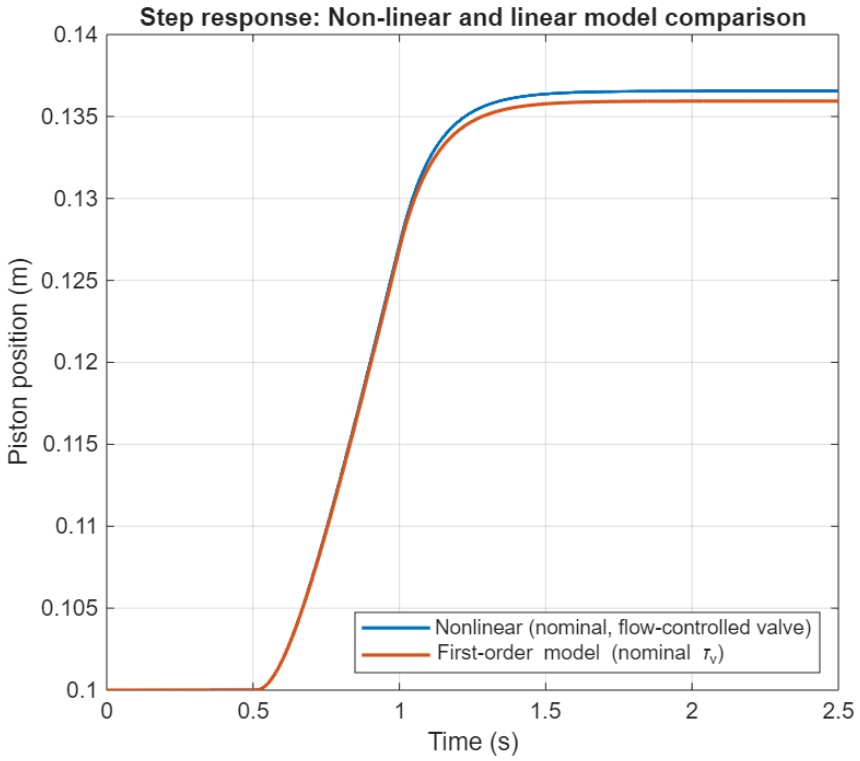


Figure 4.9: Comparison of step response for non-linear flow-controlled model and first-order simplification using nominal parameter values

4.2 Square trajectory simulation results

The results from simulating a square trajectory in XZ-plane with the implemented control scheme is presented in Figures 4.10, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19 which show the simulated square trajectory, position tracking error, velocity tracking error, cylinder velocity and position tracking, cartesian position and velocity tracking, valve flow rates and finally the sum of the valve flow rates respectively. This simulation will act as a nominal baseline where each figure is presented as a stacked pair, the upper one without noise and the lower one with noise to test best case motion tracking performance and the impact of sensor noise. After this nominal simulation, several more cases were tested to evaluate the impact of null-space movements, open-loop inverse kinematics, large model errors and input rate of change penalty.

Noise in the nominal simulation was generated as a zero mean gaussian disturbance, added to the cylinder position measurements and to measurements of the plant pressures. The first control mode corresponding to controlling all 4 DOFs was used during the simulation. Each link weighs 50kg and the EE carries a tool that weighs 100kg. The parameters for the MPC during the simulation, such as penalties and

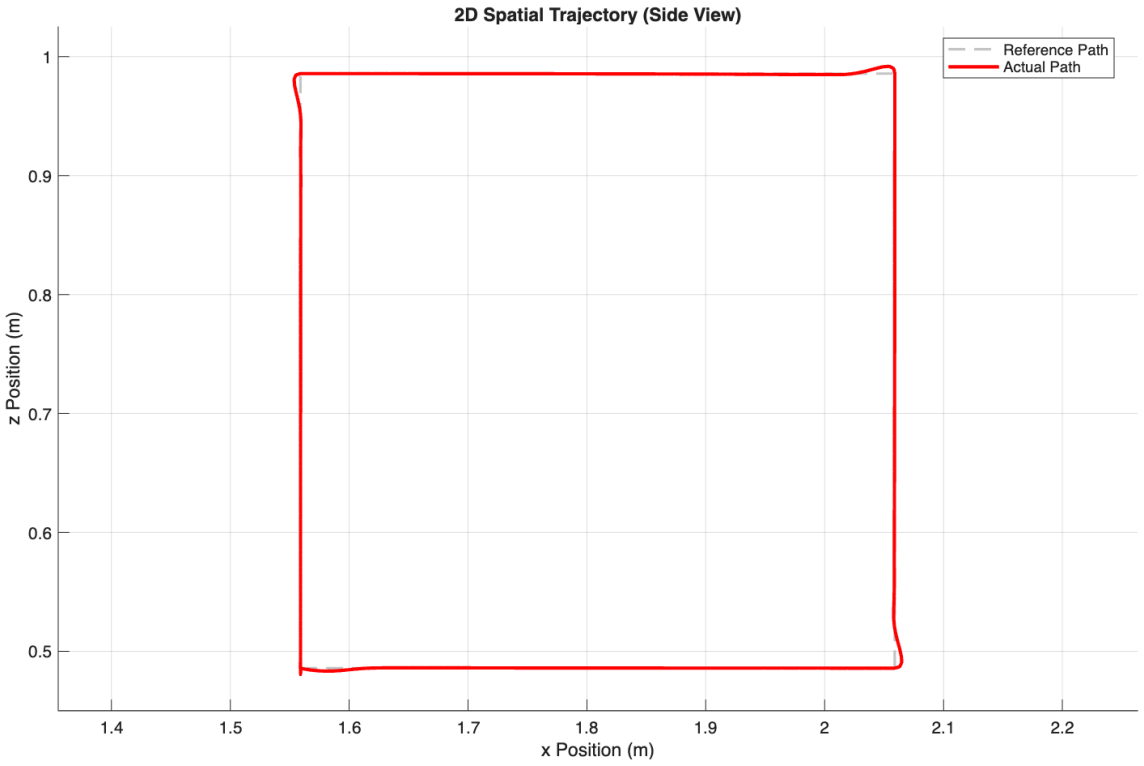
horizon length, were selected as: $H = 10$, $P_f = Q$ $S = 10^6$, $\Lambda = 10^5$,

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1000000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1000000 \end{bmatrix} \tag{4.1}$$

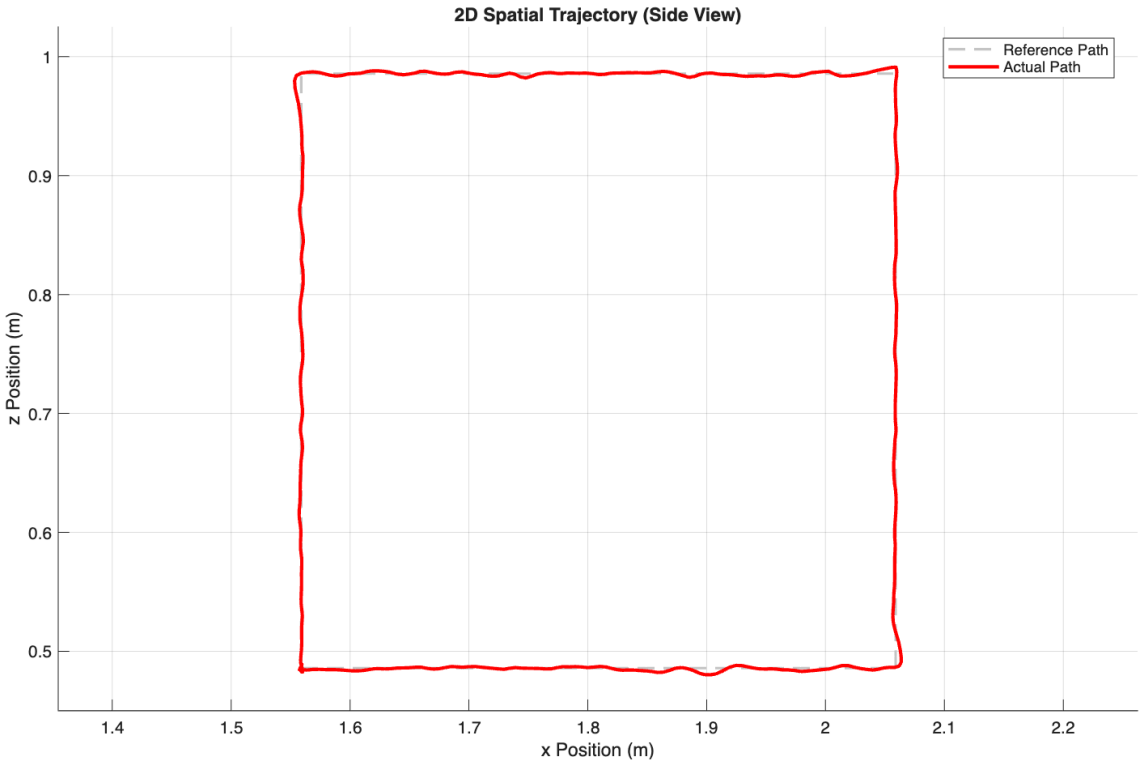
and

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

Note that it is only the reference for the valve flow rates in Figure 4.17 and 4.18 that represent actual inputs to the plant. References in other figures are based or integrated from either the operator task input or cylinder velocity solved with inverse kinematics.

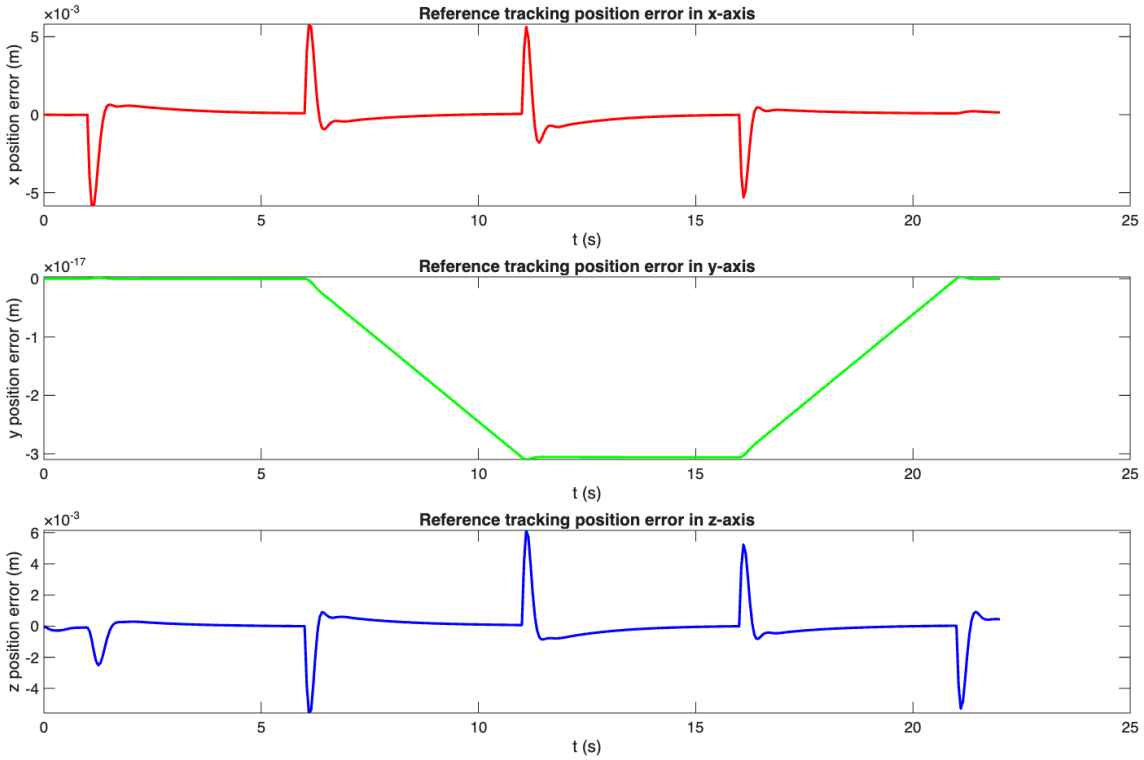


(a) Without noise

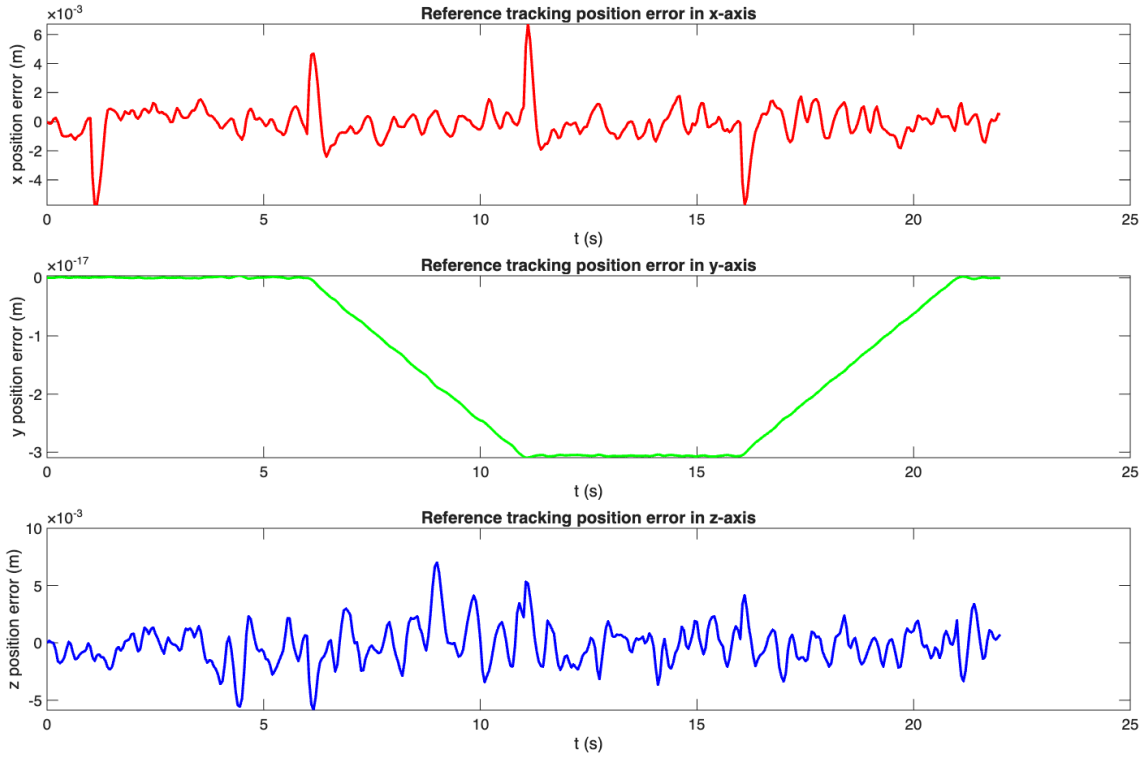


(b) With noise

Figure 4.10: Side view of simulated square trajectory with the EE starting in the lower left corner and traveling counter-clockwise

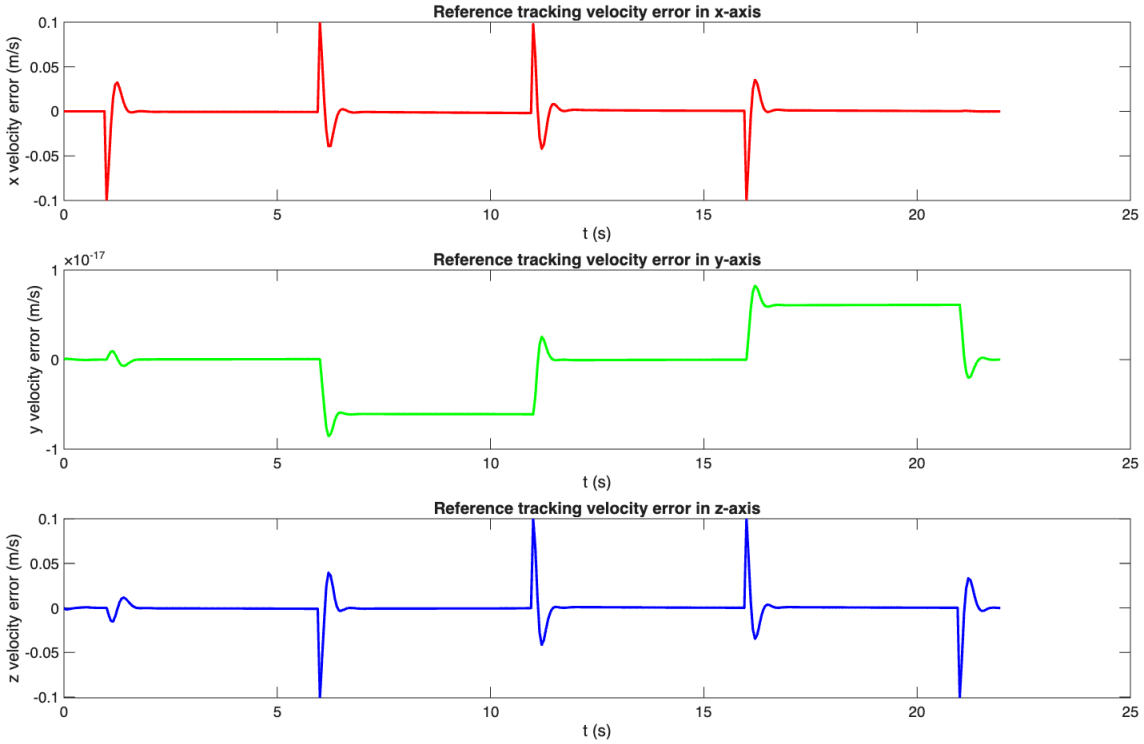


(a) Without noise. Maximum error(m): x:0.0059, z:0.0061

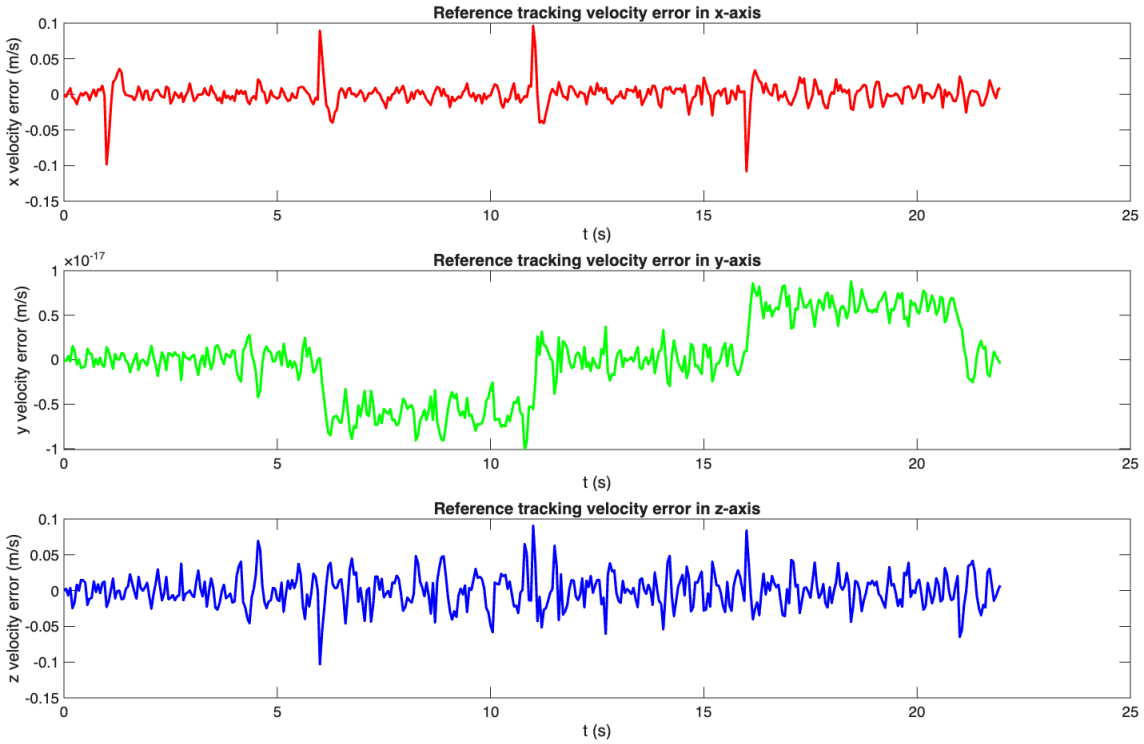


(b) With noise. Maximum error(m): x:0.0067, z:0.0070

Figure 4.11: Cartesian position tracking error during square trajectory

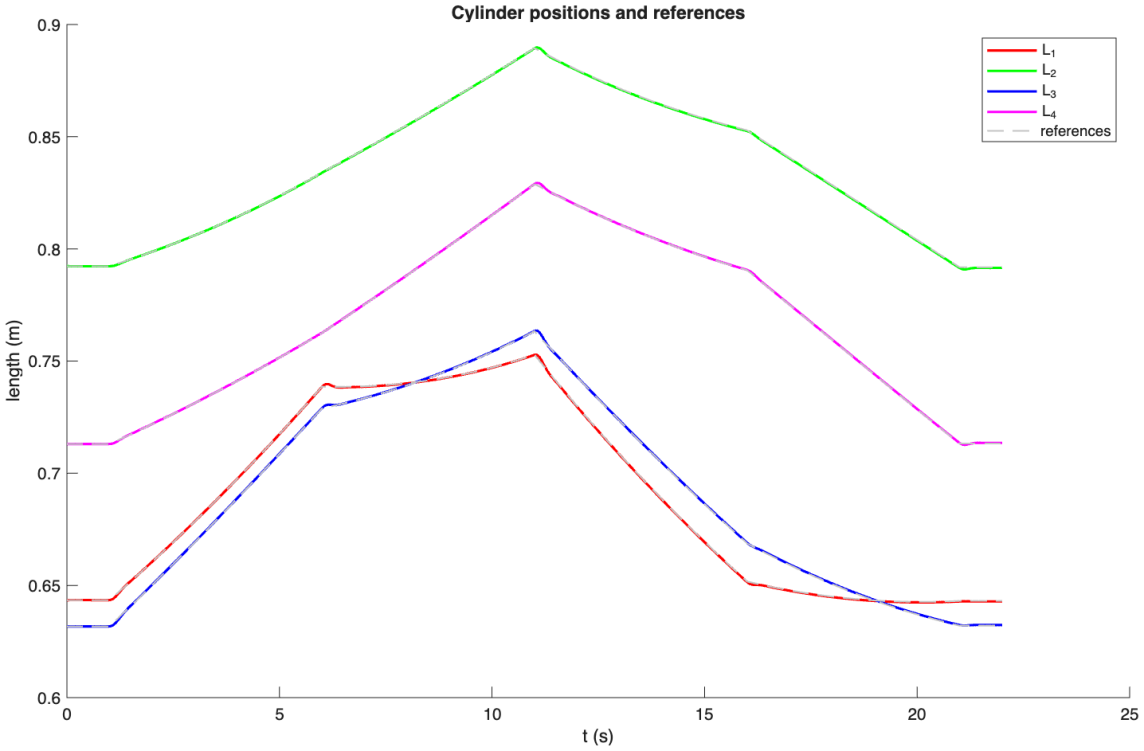


(a) Without noise. Maximum error(m/s): x:0.1000, z:0.1009

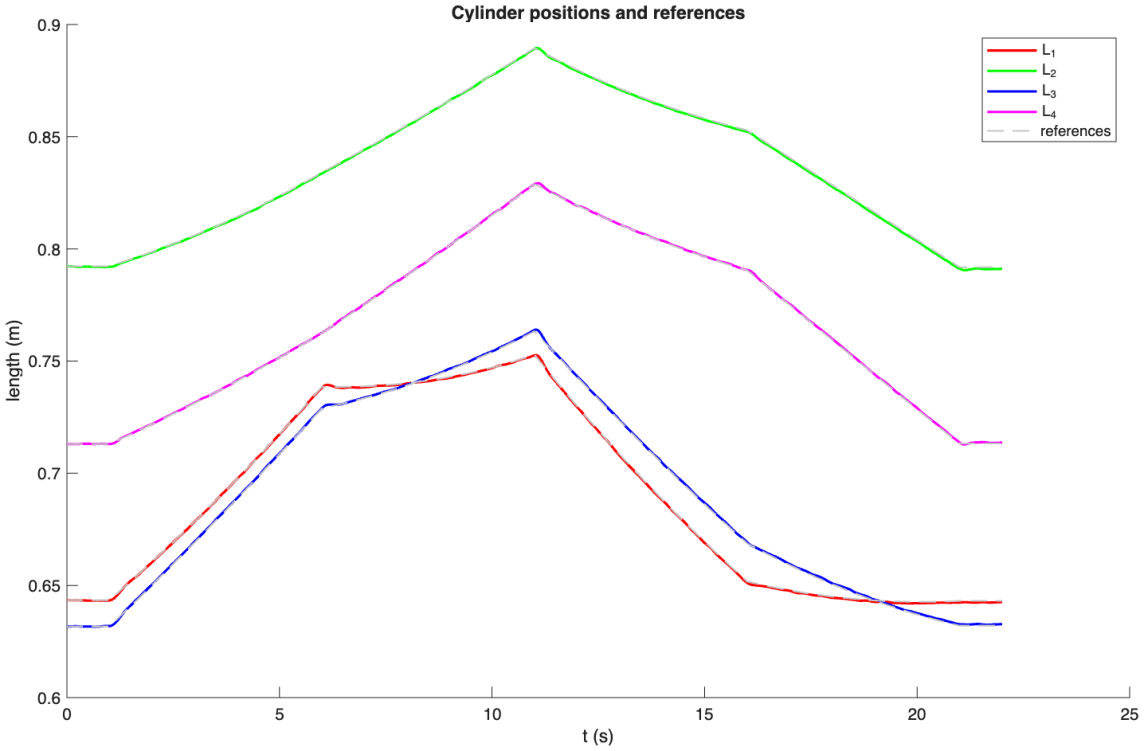


(b) With noise. Maximum error(m/s): x:0.1081, z:0.1032

Figure 4.12: Cartesian position tracking error during square trajectory

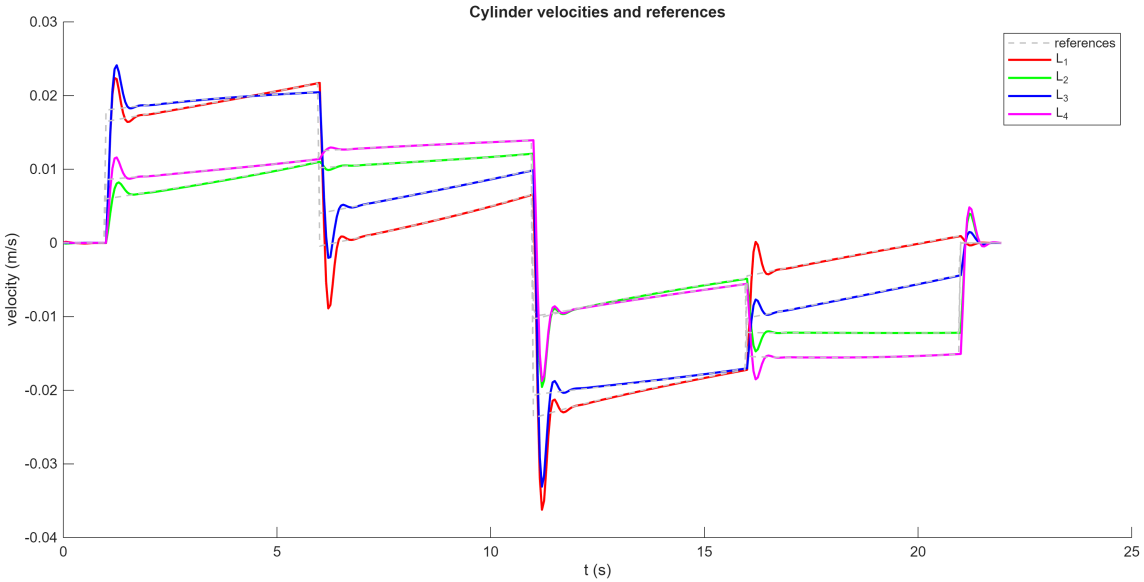


(a) Without noise

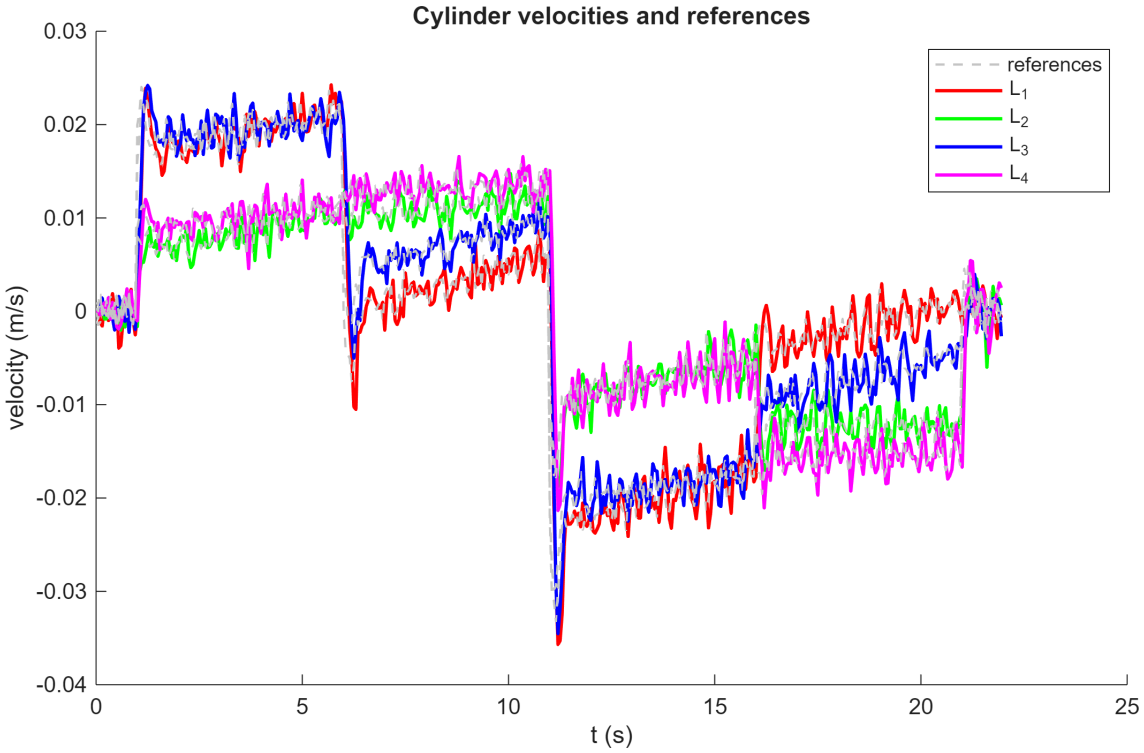


(b) With noise

Figure 4.13: Cylinder positions and references during square trajectory

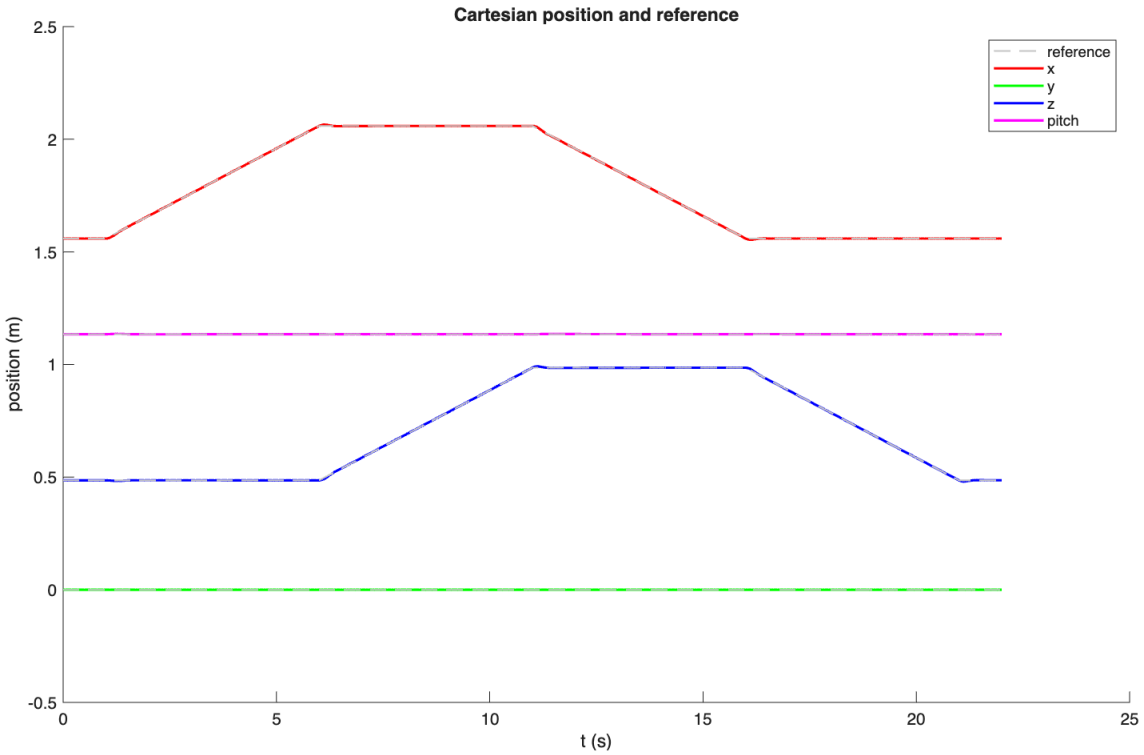


(a) Without noise

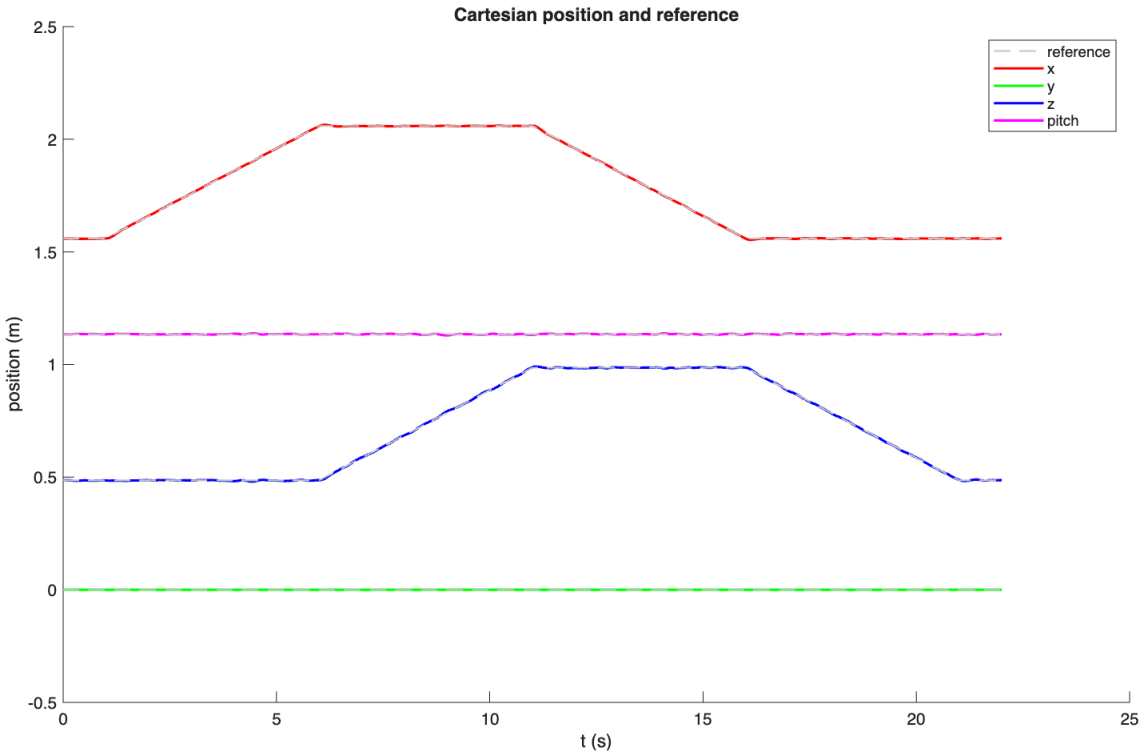


(b) With noise

Figure 4.14: Cylinder velocities and references during square trajectory

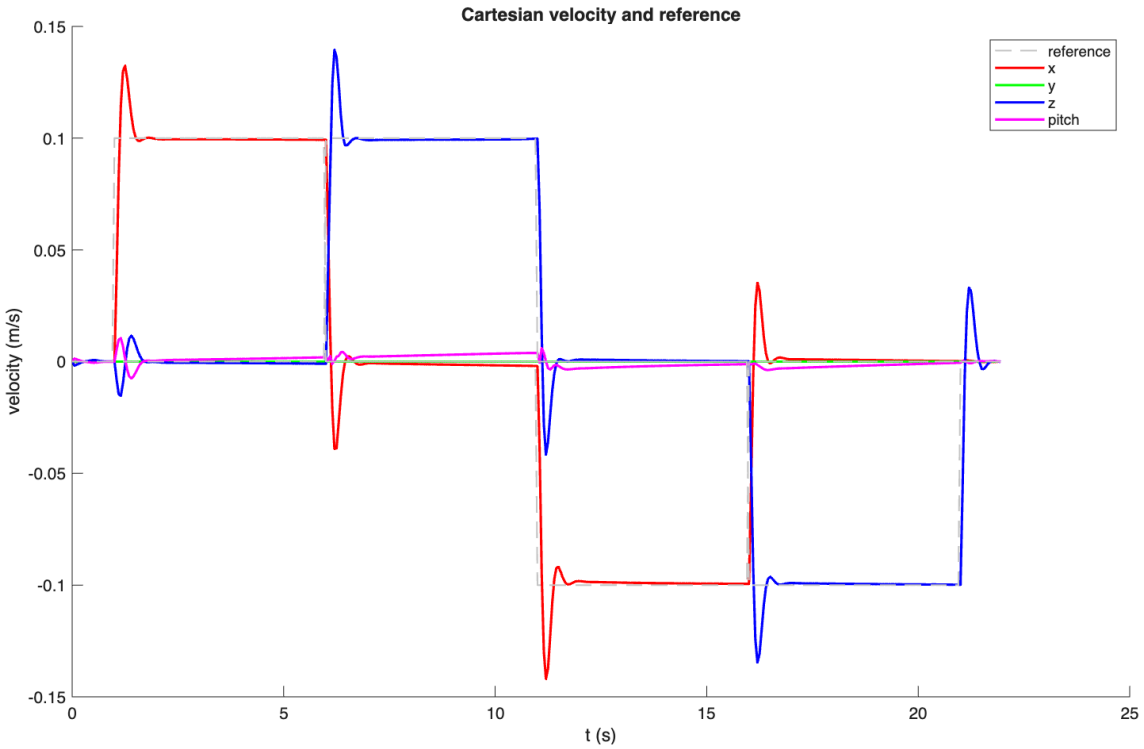


(a) Without noise

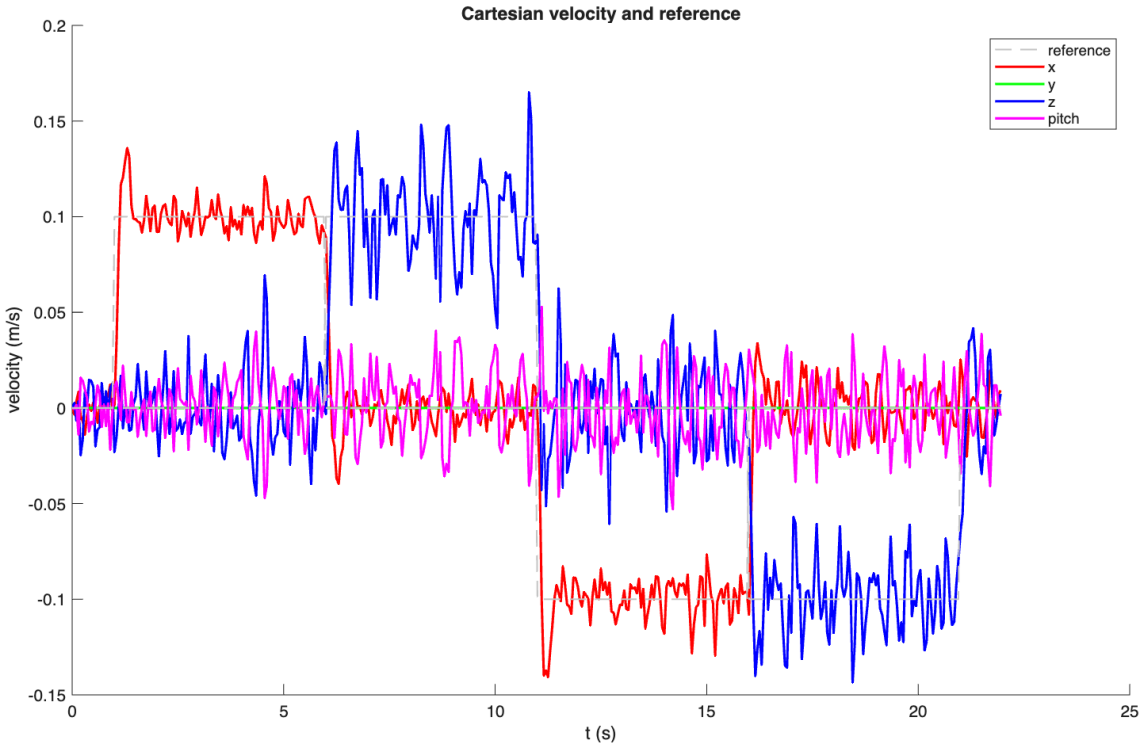


(b) With noise

Figure 4.15: Cartesian position and reference during square trajectory

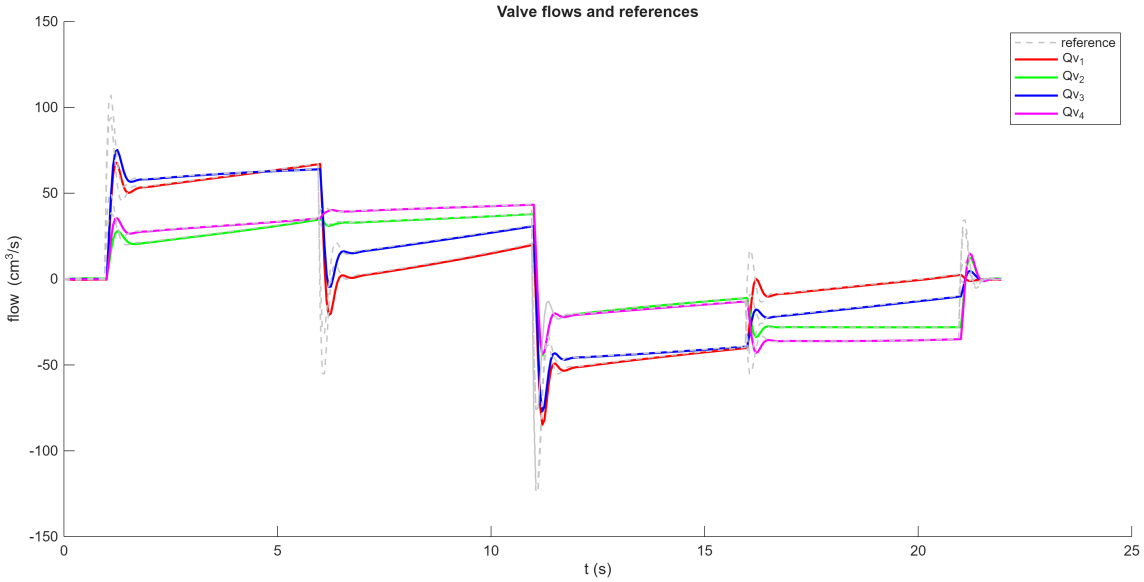


(a) Without noise

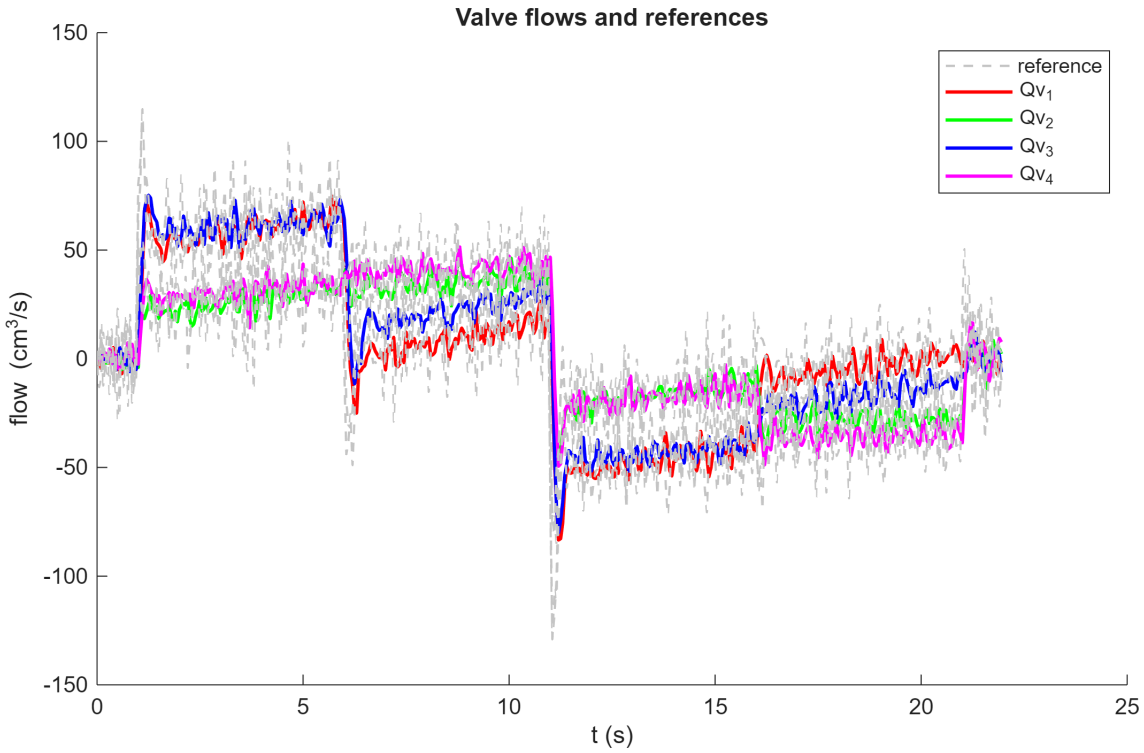


(b) With noise

Figure 4.16: Cartesian velocity and reference during square trajectory

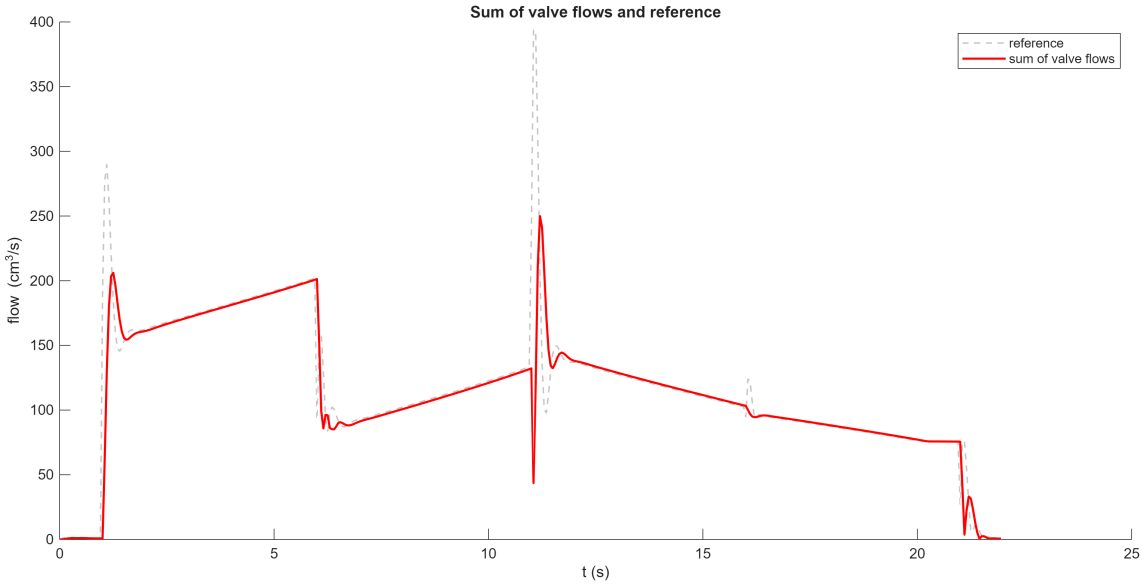


(a) Without noise

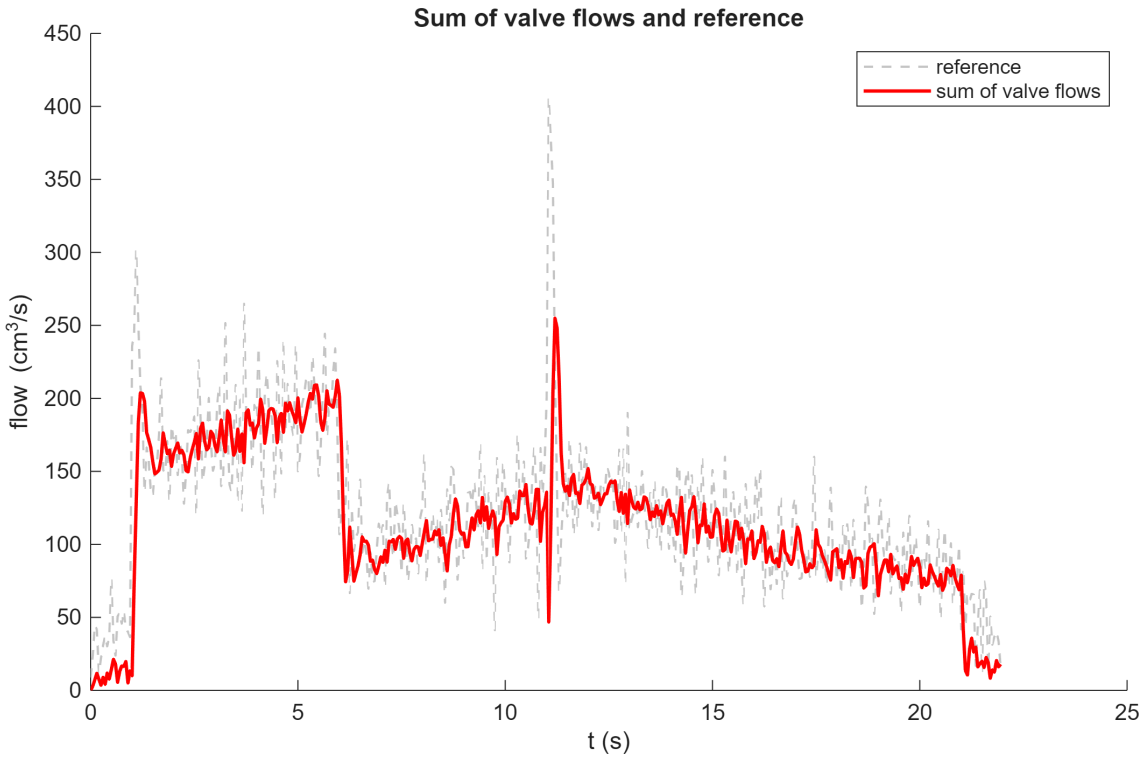


(b) With noise

Figure 4.17: Valve flows and references during square trajectory

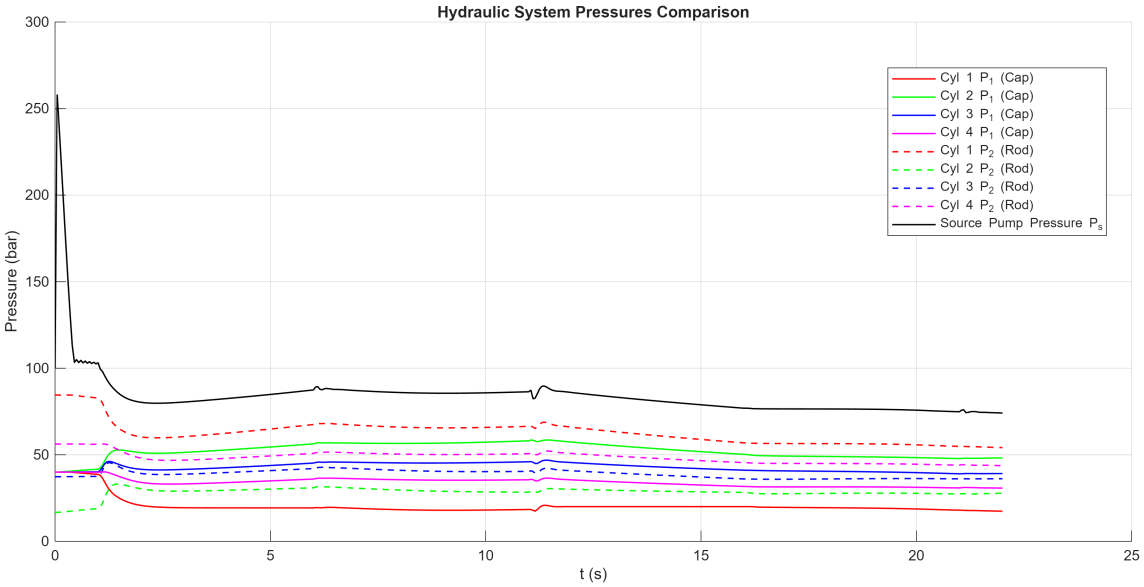


(a) Without noise

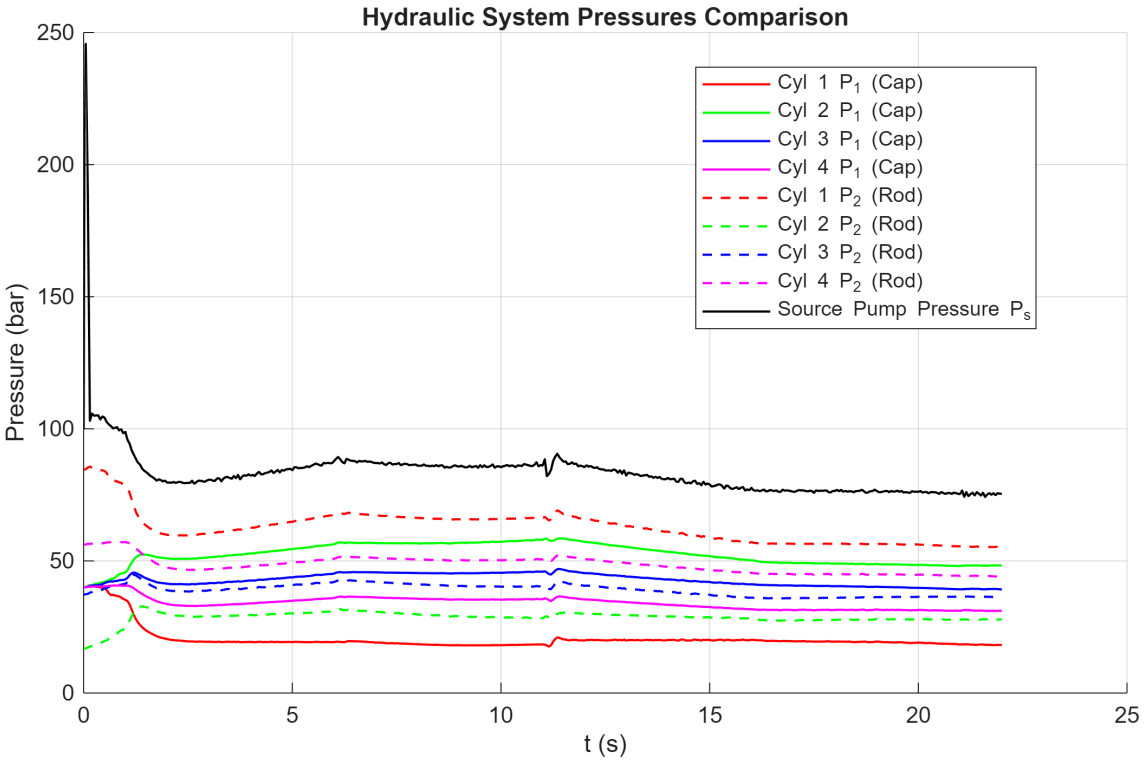


(b) With noise

Figure 4.18: Sum of valve flows and references during square trajectory



(a) Without noise



(b) With noise

Figure 4.19: Hydraulic pressures in each cylinder chamber

4.2.1 Null-space movements

The effect of the null-space movements from secondary objective of the inverse kinematic optimization can be seen in Figure 4.20 and 4.21. In Figure 4.20, deviation

from the desired cylinder lengths, set as the initial configuration, was penalized with

$$\mathbf{W} = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

which corresponds to penalizing the first and second cylinder. In Figure 4.21, no secondary objective was included in the objective function. These results are without noise.

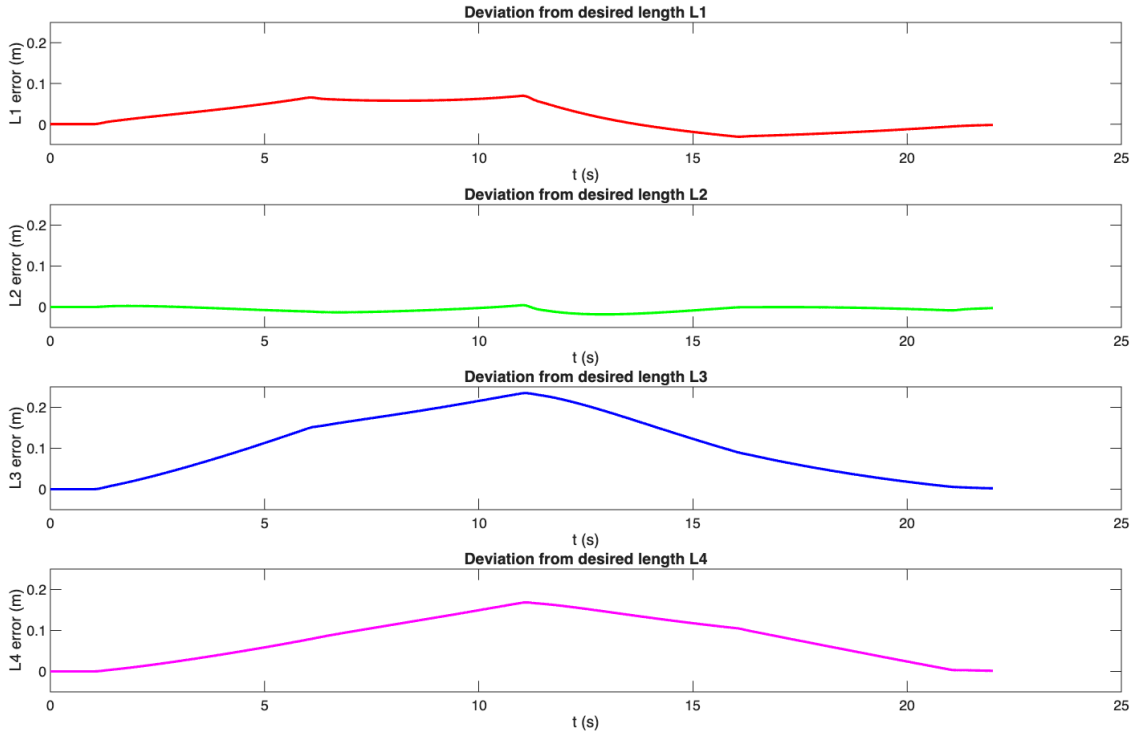


Figure 4.20: Deviation from desired cylinder length with active secondary objective

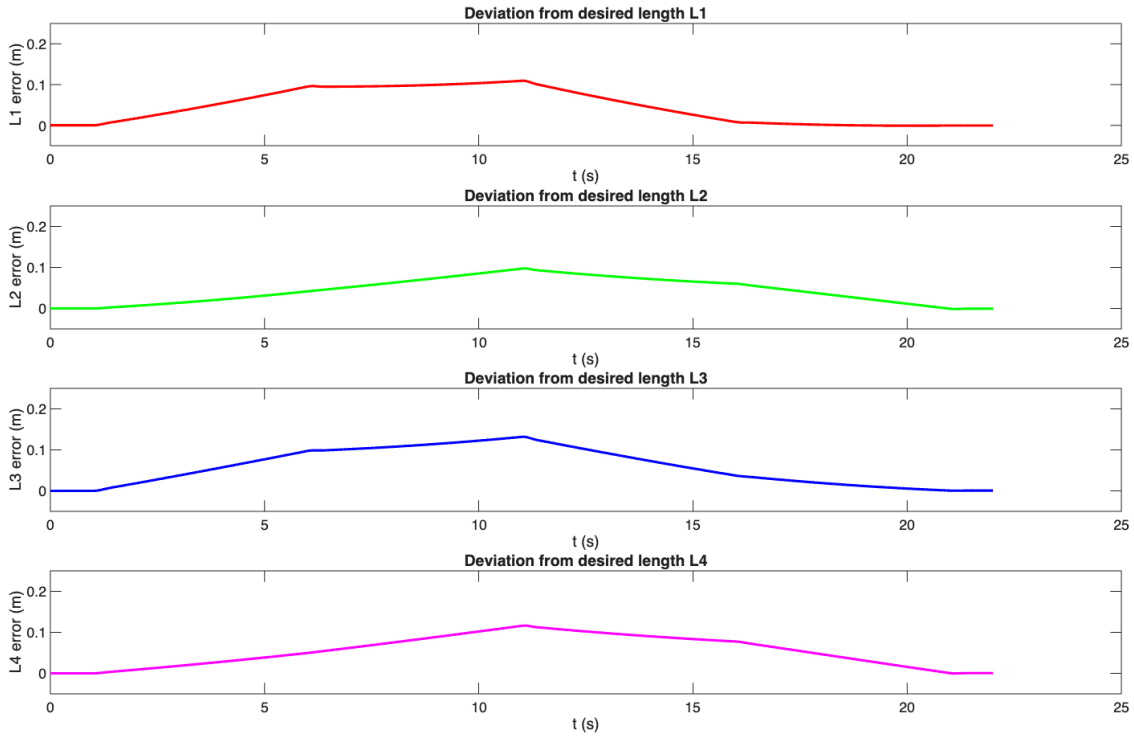


Figure 4.21: Deviation from desired cylinder length without secondary objective

4.2.2 Open-loop inverse kinematics

The Figures 4.22 and 4.23 show the noise-free case when the task-space feedback is disabled by setting the proportional and integral gains to zero, making the inverse kinematics algorithm (3.30) open-loop. The maximum position tracking errors(m) were x:0.0089 and z:0.0263.

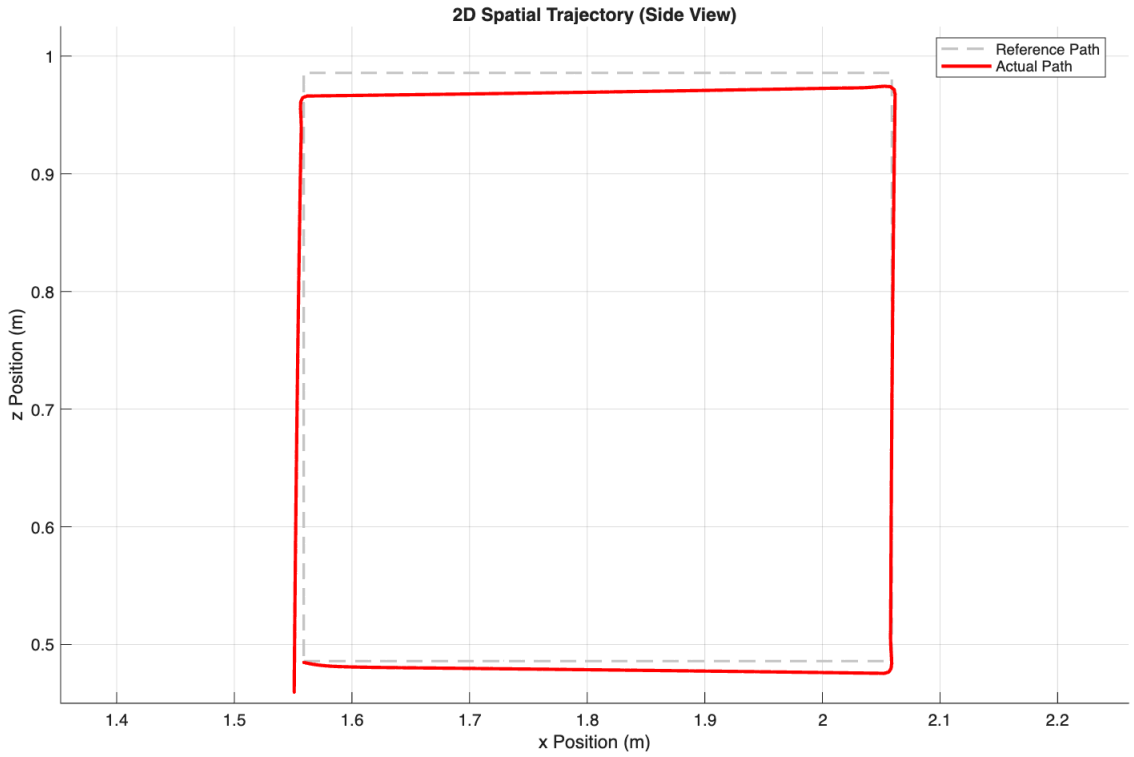


Figure 4.22: Side view of square trajectory with open-loop inverse kinematics

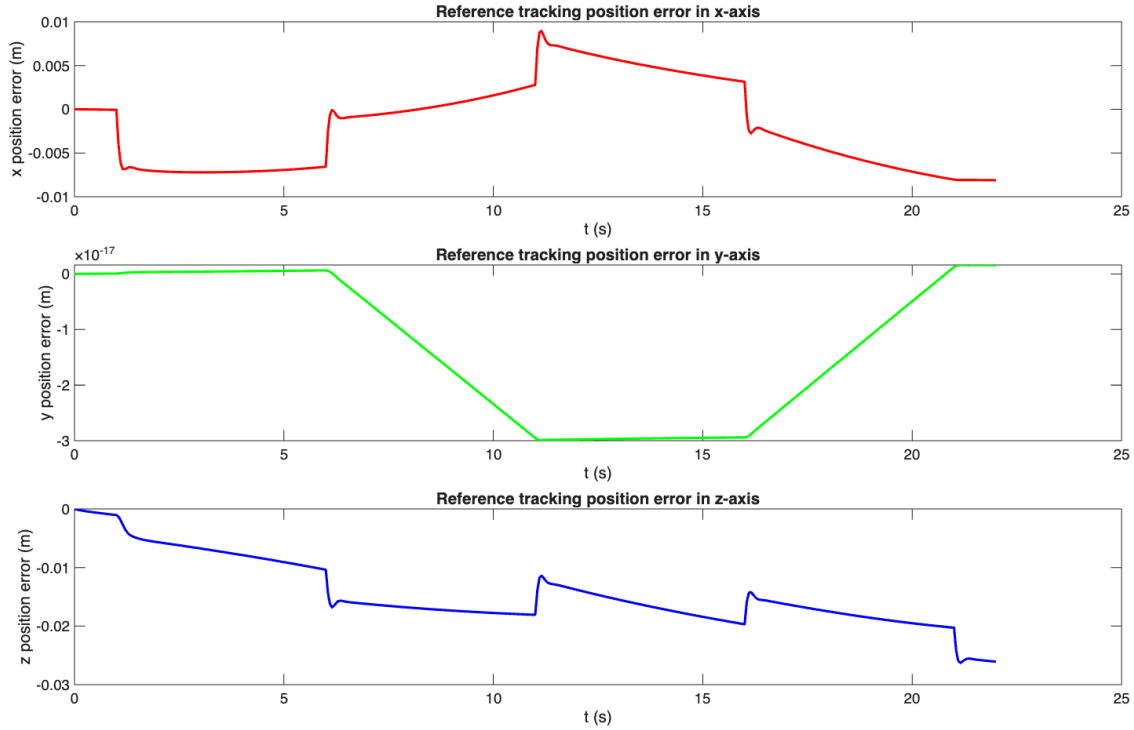


Figure 4.23: Tracking error with open-loop inverse kinematics square trajectory

4.2.3 Large model error

A simulation with parameters tuned to the maximum of Table 3.1 was tested to evaluate heavy model error disturbance rejection. These values represent what is believed to be an unrealistic plant that should be harder to control since the parameters differ greatly from the model from which the control model was derived. The simulation, seen in Figures 4.24 and 4.25, was successful, but only with CLIK. If CLIK was disabled, the robot eventually deviated too far from the reference trajectory and hit a limit.

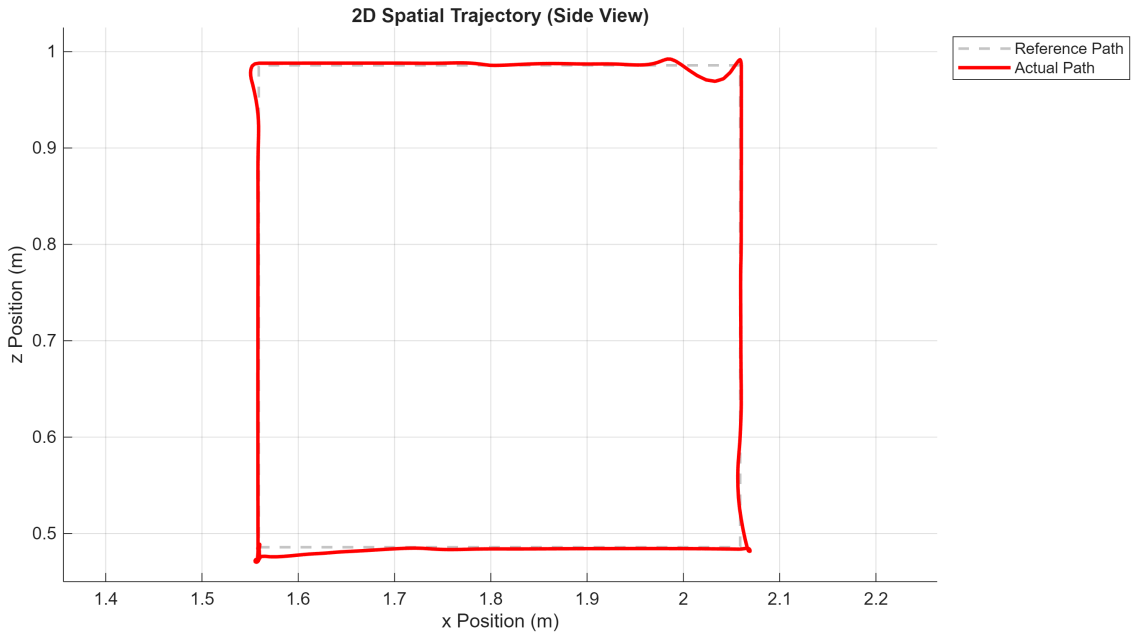


Figure 4.24: Side view of square trajectory with large model errors

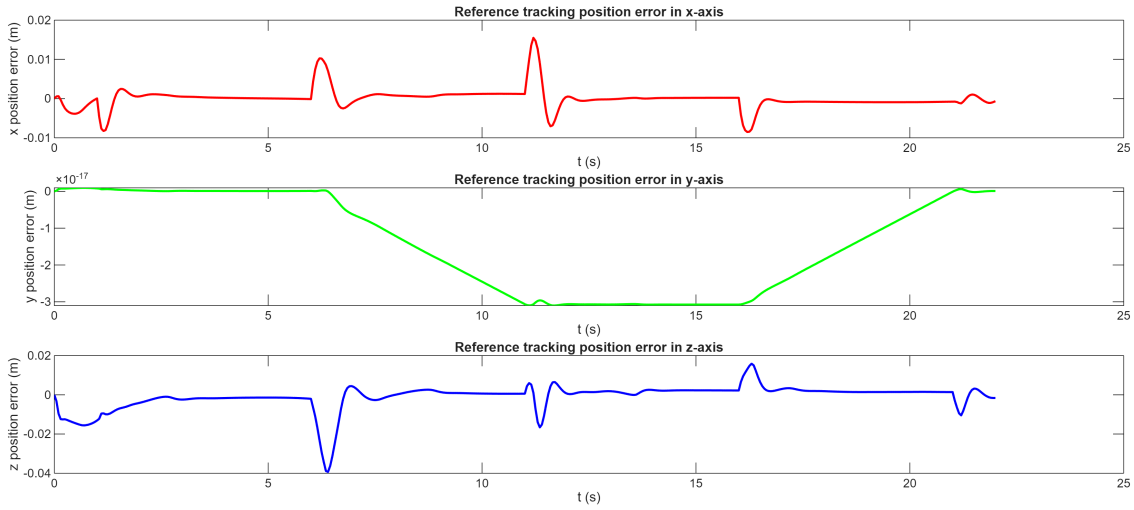


Figure 4.25: Tracking error of trajectory with large model errors

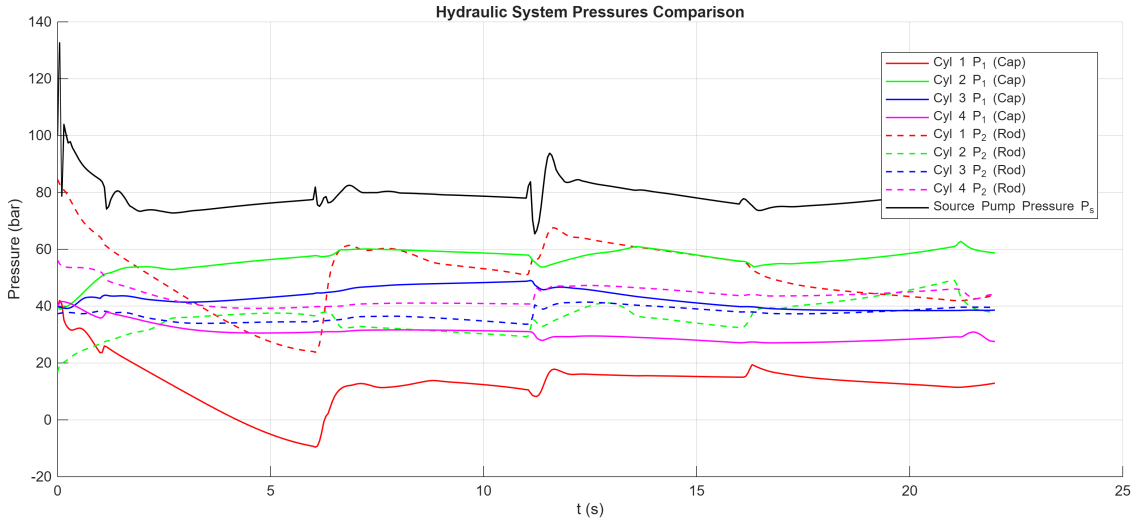


Figure 4.26: Hydraulic pressures in each cylinder during trajectory with large model errors

4.2.4 Input rate of change penalty

Figure 4.27 and 4.28 show a noisy simulation without any penalty on the rate of change of the input.

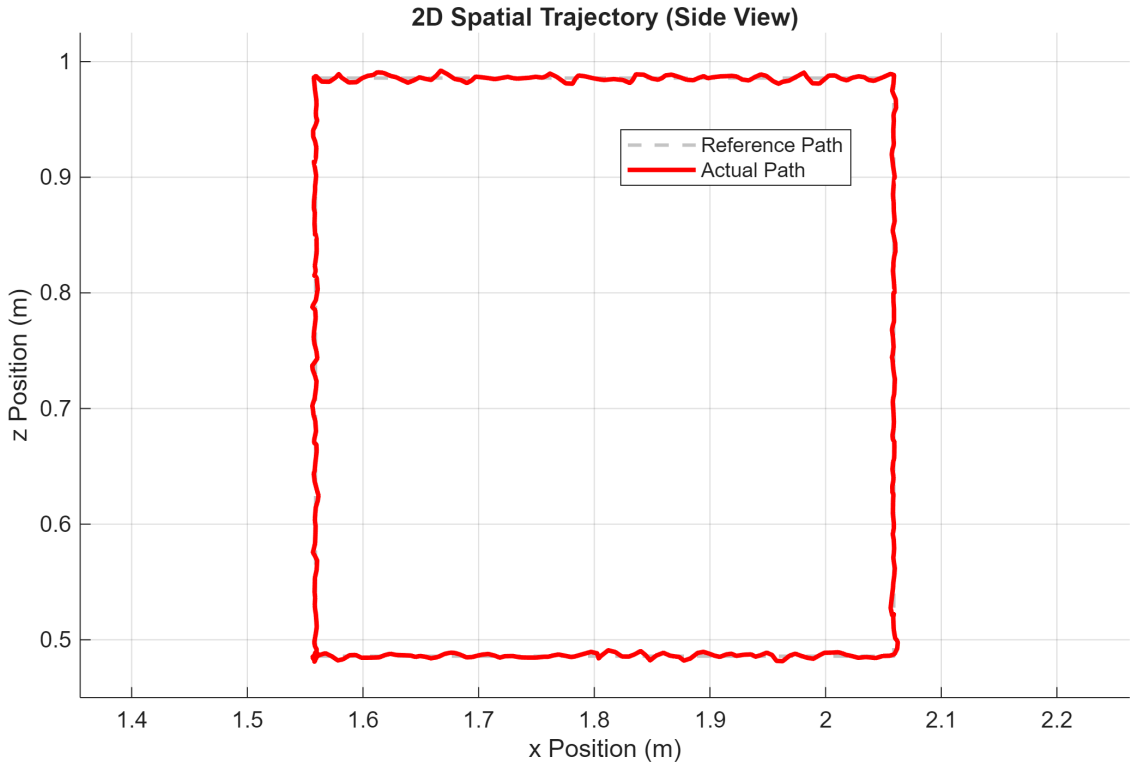


Figure 4.27: Noisy square trajectory with no input rate penalty

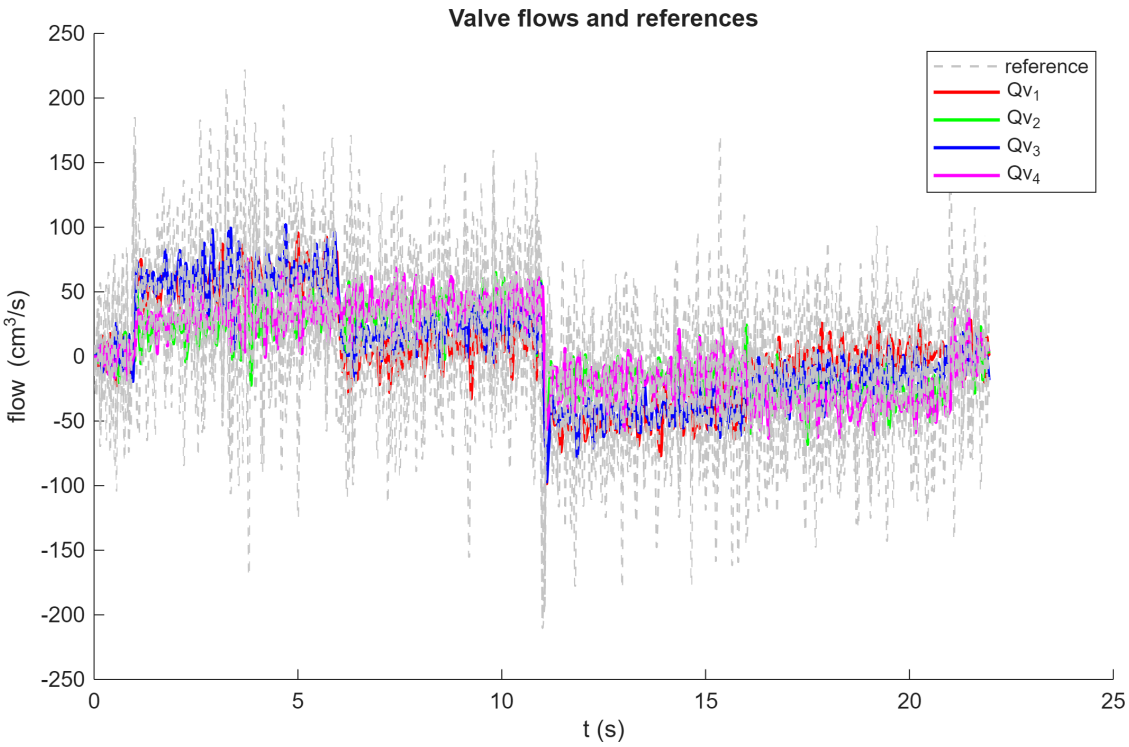


Figure 4.28: Noisy valve flow rates with no input rate penalty

5
Discussion

This chapter will discuss the methods used, if the objectives of the thesis have been fulfilled as well as areas of future research. The chapter will also seek to provide answers to the research questions posed in the introduction.

5.1 Discussion of method

This section will evaluate the applied methods and suggest further areas of research that could improve the quality of the control scheme.

5.1.1 Model development

Many of the dynamic equations presented in Section 3.1 are based on well established and commonly used in modeling of hydraulic systems. However, hydraulic systems are in practice much more nuanced. For example, safety components such as pressure relief and counterbalance (load-holding) valves were not added in the model due to lack of information. The decision of only considering dynamics of a load-sensing pump, a four-way valve and a double-acting cylinder was based on the availability of sources describing them, and to some extent the perceived importance of the component. For example, while additional pressure- or flow-compensating components may be fitted in series with the pump or valve to improve system behavior, adding components would not necessarily benefit the methodology. Therefore this thesis only focus on only the main components of a hydraulic system. An improved methodology may investigate the contribution of individual components in a similar system, provided sufficient knowledge of their dynamics.

Early in model development, the valve was considered important since it is the only input of the system. Modeling its dynamics well is therefore important, and consequentially appears to be a thoroughly researched topic. The modeling theory described in 2.1.4 provides a model with minimal modeling uncertainty. However, the use of the orifice equation (Equation 2.11) introduces some uncertainty, since it it is only an approximation for actual flow behavior [3]. Also, the exact implementation of the feed-forward flow control structure is based on a previous article [12] but does not implement the full control structure described, as feed-forward was considered a sufficient simplification for simulation purposes.

As for the hydraulic cylinder modeling, its motion is described well by Newton's

second law and several fundamental expressions as described in Section 2.1.3. The contribution of friction and viscous forces however was not investigated in detail, yet it could cause much undesirable movement, such as stick-slip motion [28]. The modeling of friction is thus subject for further research if identified as a significant factor in positioning of pistons. Furthermore, the question whether the motion of multiple cylinders can be decoupled or not was asked in the introduction, but this could not be definitively answered since no physical tests were conducted with the proposed control scheme. However, as explained in Section 3.2 each valve only sees a set maximum pressure differential due to its internal pressure compensation. This in theory decouples the individual valves as long as the pump can keep up the required pressure margin and is not overloaded by excessive motion. Since the pump output is shared between all valves, its performance could greatly affect how valves and cylinders interact with each other. Therefore, the maximum flow constraint in Section 3.4.4 was implemented to ensure that the pump is not overloaded. The pressures in each cylinder during the simulation can be seen in Figure 4.19, where the pump always maintains a pressure margin over the highest pressure in any cylinder.

The pump model is perhaps the biggest source of uncertainty in the non-linear model. While the continuity equation and pump flow describes the evolution of pressure and flow of the pump model well, swash-plate dynamics could not be considered in detail as described in Section 2.1.5. Provided the previous reasoning about decoupling of pump dynamics, the accuracy of the exact pump dynamics might not be important if the valve is sufficiently compensated. While the first order swash-plate model might be inaccurate, it still allows for some dynamic motion as opposed to removing it completely.

Considering what has been discussed in this section, it must be stated that modeling is perhaps the biggest uncertainty of this methodology. The development of models could be improved provided proper component specifications and testing. The quality of any result or conclusion of this report is thus contingent on the validity of the current models since no physical tests were performed.

5.1.2 Sensitivity analysis and parameter uncertainty

The development of simpler model descriptions was initially driven by the need for a computationally efficient control strategy, where linear MPC was the only practical choice considering other requirements. Developing a simpler model means identification of superfluous dynamics in the non-linear model, and unknown or uncertain model parameters has to be defined. Without proper specifications, representative parameter values and their dynamic effects had to be found methodically, which is why a combination of local and global sensitivity analysis was preformed. It was decided that the simplest configuration of a single pump, valve and cylinder was to be tested, which disregards coupling effects that could appear in the complete system description. It was reasoned that using a simple system would isolate parameters so their effects could be understood. An analysis on the complete system could improve the quality of the analysis, test cases, parameter ranges and methodology

would likely need to be reworked.

While the OAT method was never formally defined in Section 2.3, the principle of varying one parameter at a time is useful for gaining understanding of how individual parameters affect the output trajectory. Thus it is possible to determine an approximate range of parameters that can be considered reasonable. This method inevitably introduces bias but it is likely an improvement over choosing and only considering a single value for each parameter when the value uncertainty is high. Using the defined parameter ranges is also what allows for global sensitivity analysis to be performed. The Morris method was chosen because it provides graphs that summarize the influence of parameters in an easily interpretable way. However, it does not provide a way to classify parameters numerically, meaning that results only serve as an indication of the global effect for individual parameters.

Furthermore, the Morris analysis requires choosing the amount of trajectories and step size, which appears to affect the convergence of the performance indices (non-linearity σ and influence μ^*). The range of parameter values also appear to affect convergence, as some random starting points in the parameter space can result in instability. Regardless, the method still allows for general conclusions about parameter influence.

5.1.3 Reference generation

It was briefly mentioned in Section 3.5 how the flow rate targets are kept constant over the horizon and the position targets are just integrated from the current iteration of inverse kinematics. This is not an ideal solution since the MPC looks into the future at essentially made up references that may change suddenly. The benefit of this simple approach is that the computational load is quite low, but it lacks look-ahead capability. The nature of this online motion control problem means that there is no way of predicting the input of the operator and thus no way of providing known future references to the MPC.
An alternative approach that was explored was to run the inverse kinematics algorithm at each iteration of the horizon which yields slightly reduced tracking error by eliminating some integration error but at the cost of compute efficiency since it requires solving 9 extra optimization problems, with compute time scaling linearly. In simulation the results of the two methods were very similar. But this approach still suffers from the fact that it is not possible to predict what task-space motion an operator might input. The instant that an operator commands movement in a different direction, the current prediction becomes invalid which is likely what results in the overshoot seen in the square trajectory simulation in Figure 4.10. Where this alternative approach would be useful is in cases where a complete predefined task trajectory is supplied to the MPC offline. Then, provided adequate compute power is available, it would be possible to better utilize the predictive capabilities of the MPC by computing the inverse kinematics for the whole horizon and looking ahead at known references.

Another idea that was discussed but not yet implemented is to add some lag to the operator controls. This means that in the instant the operator tilts the joystick, the reference trajectory continues in the current direction for a short while before changing direction instead of changing direction immediately. The idea is that this would allow the MPC to be able to look ahead, see the turn and then adjust for the changing reference. This method would then require kinematic inversion for the whole horizon, just as when feeding a predefined task. However, the outcome of this method remains to be investigated. The main questions would be if the lag and possible overshoot reduction is worth it over just accepting some overshoot, and how it affects the responsiveness in the machine.

5.1.4 Combining MPC tracking and trajectory generation

The question if MPC and trajectory generation could be combined was posed in the introduction and was studied in a report about automated guided vehicles (AGV) by Juncheng et. al. [29]. However, the premises of this report differs from this project. In [29], the full path is first computed and then optimized once offline with MPC before the AGV even starts moving, which is not the general case for the hydraulic robot. Due to the unpredictive nature of the operator input, reference generation has to be run online together with the MPC motion tracking. Implementing the same approach of improving the reference for the hydraulic robot would introduce yet another optimization problem and increase the computational requirements. For this reason, the unified MPC approach was dismissed. To draw a parallel to [29], one could say that instead of an MPC, the CLIK method serves as the reference improving stage in this thesis, which is not nearly as computationally intensive since it is uses a PI controller.

Another approach to combining reference generation and motion tracking is provided in a report about predictive inverse kinematics [30]. This method solves IK as an optimization problem but also estimates future Jacobians, which would likely be a necessary step in an MPC implementation. Since task-space is non-linearly related to joint space this would likely rule out a linear MPC implementation unless one linearizes this relationship in the control model, which is actually shown in [30]. From here, it would be necessary to include a model of the actuator dynamics in the control model, such as the first-order model from this thesis. To be even more rigorous one could instead of the first-order model, include a pressure-based actuator model of the cylinder force, and a non-linear model of the robot dynamics. The limitations here would likely be computational resources non-linearity and constraint handling in multiple spaces. It was noted that problems are often infeasible when constraining both joint and task-space, such as in Equation 3.29.

5.1.5 External force modeling

The external force acting on each cylinder in the system is based only on the effects of gravity on each link segment. Due to the generally low velocities and acceleration of the manipulator arm, it was determined that the inertial and Coriolis terms

of the dynamic equation were relatively small and that the gravity term would dominate due to the relatively high load capacity but slow movement. Equation 3.56 also proved considerably more complicated to calculate as it includes cylinder velocity, cylinder acceleration and the cylinder-joint Jacobian derivative. In addition to these reasons for only taking gravity into account, one can also debate the overall importance of the external force since it only acts as a disturbance in the simulation. Ideally one should like to utilize the dynamic model to actually optimize the motion tracking in the MPC. However, due to it being non-linear, that was not possible in this project. It would therefore be interesting to investigate non-linear MPC to see if any performance can be gained by basing an NMPC off of the non-linear dynamic model as well as either the first order or whole non-linear hydraulic model. Another advantage of an NMPC approach would be the possibility to minimize other costs instead of just pure tracking error, for example energy consumption.

5.2 Discussion of results

This section will try to interpret the results from the sensitivity analysis and the square trajectory simulation.

5.2.1 Sensitivity analysis and model simplification

Figures 4.1 through 4.4 show trajectories for four of the parameters presented in Table 3.1. The rest of the parameters did not result in significant trajectory deviation and were therefore not presented. Plotting the piston position while varying one parameter at a time gives an indication of the relative contribution to the behavior of the system. Generally, the piston trajectory graphs show that changing parameters generally result in a change in rise time, final position of the piston, or a combination of both. It appears that cylinder leakage, external force and load-sense margin pressure primarily cause deviation in final piston position, while the valve time constant primarily affects the rise time of the trajectory. This insight was the main driver for the choice of metrics used in the Morris analysis, namely rise time and RMSE, to understand these effects in a global scope.

Another common behavior observed in these figures is that spool position- and flow-controlled cases do not reach the same final piston position and does not necessarily have the same response to changes in parameter. For example, Figures 4.3 and 4.4 show that valve flow control can effectively compensate for pressure resulting from external forces or lowering of valve margin pressure. Cylinder leakage however causes similar trajectories in both cases, as shown in 4.2. One reason for the flow-controlled valve being insensitive to some parameter changes is that a feed-forward flow-controller is assumed where the exact valve model is provided to calculate spool position references. On the other hand, the spool-controlled valve shows how the trajectory would evolve without this idealization of flow control. Therefore the actual performance of a flow-controlled valve might perform somewhere in between these two extremes.

Considering the OAT analysis it was hypothesized that the valve dynamics would have the largest effect on metrics in the Morris analysis. As described in Section 5.1.2, this analysis is performed qualitatively and is subject to interpretation. It appears to be the case that the hypothesis is mostly correct, considering Figures 4.5 and 4.6, which show that valve time constant generally has the highest overall influence μ^* on both metrics. Cylinder leakage and external force can also be considered influential, more so in the case of trajectory RMSE, and appear to be more dependent on the value of other parameters in the system, as shown by higher non-linearity σ 4.7 4.8 in Figure 4.8. As described in Section 2.1.1 cylinder leakage is one of the parameters that is often assumed to be negligible, but it can have a significant effect on the system if large enough. As observed in pervious studies, friction between cylinder and piston can vary with the age and use [6], which could reasonably cause wear on cylinder seals and increase leakage over time. That being said, there are many parameters that has only small effect on the metrics, which formed the basis for model simplification, Section 3.2.

The convergence of the overall influence μ^* in the metrics, Figures 4.5 and 4.6, show that cylinder leakage and external force influence does not converge over the designated $r = 200$ trajectories, while all other parameters do converge. The choice of parameter range likely affect convergence, but it is unknown which parameter or parameters causes this behavior. Lowering the investigated parameter range and considering next highest and lowest values of parameter values alleviate convergence issues slightly but does not completely eliminate it. The resulting convergence graphs change slightly, but the conclusion of parameter importance is largely the same.

Figure 4.9 shows that the first-order simplification of the non-linear model result in approximately the same piston trajectory, and thus the new model was deemed sufficient for control purposes. Since the non-linear model contains some modeling error and much parameter uncertainty, the simplified model cannot account for all of these errors. This is because many of the uncertain parameters depend on pressure states, which were ignored in the first order model. Cylinder leakage and external force could not be accounted for even though their influence is evident in both OAT and Morris analysis. The simplification results in the control system being more dependent on the CLIK scheme defined in Chapter 3.3.1.1, as end-effector control error is corrected as a change in the velocity reference. The CLIK scheme can therefore be understood as the part of the controller responsible for eliminating modeling error in the MPC.

A reasonable improvement to the first-order model is to model cylinder leakage and external force, but this change is dependent on the inclusion of cylinder pressure sensors in the machine. An alternative is using the simulated external force, calculated in Section 3.3.3, to estimate cylinder pressures. The accuracy of these estimations are dependent on the identification of forces opposing cylinder movement, such as F_{drag} and F_{frict}.

5.2.2 Motion tracking performance

The results presented for the nominal simulation suggests that the implemented control structure effectively controls the non-linear plant despite many model simplifications in the control scheme. The simulated noise in the sensors does cause some accuracy loss, seen from the increased maximum position and velocity tracking error, but the EE still manages to effectively track the trajectory. When looking at just the graphs for cartesian position (Figure 4.15) or positions of the cylinders (Figure 4.13), it is hard to even tell which is which. It is worth to note however that noise probably has a bigger impact on tracking error than what the maximum tracking error may indicate. All tracking error in the noise-free simulation seems to come from overshoot, so the mean tracking error is much higher for the noisy simulation.

One can note how the reference for the valve flow rates in Figure 4.17 and 4.18 often overcorrects the state it tries to control. This is a consequence of optimal control and how MPC looks ahead. During a direction change the MPC unexpectedly sees the EE traveling in the wrong direction and it becomes more optimal to send overcorrecting flow rate references in order to drive the deviation states to zero faster. This causes some overshoot so it can be seen, especiall in Figure 4.17 a), that opposite overcorrecting inputs are sent just after to correct the overshoot.

Another interesting result is that of the impact of penalty on the rate of change of the input. Comparing Figures 4.10, 4.17 and 4.27, 4.28 one can see that the latter figures suffer much more from noise. One possible explanation is that since we use no filtering, the noise is exacerbated when deriving the cylinder lengths to calculate cylinder velocity and then valve flow rate. The faulty valve flow rate measurement then causes the MPC to constantly overcorrect which can be seen from the consistently jittering inputs in Figure 4.28. The rate of change penalty seems to work in practice like a low-pass filter and smoothens the MPC:s reaction to the faulty flow rate measurement. Even though this solution delivers improved results, it is still highly relevant to investigate the implementation of a Kalman filter to properly deal with disturbances. It is also worth to mention pressure sensors in this context. If pressure could be measured for the pump and in each cylinder chamber, it would enable observation of forces inside each cylinder which would eliminate the need for the spring model in the second-order model and make the non-linear model observable. This would also open up for other interesting control methods such as force control or for example a gain scheduled LQR based of the non-linear model.

Overall, the performance in the simulation indicates robustness. Even though the extent of the control schemes stability and feasibility cannot be entirely determined from experimental results, it can be argued that these results indicate those properties to some degree. Therefore, rigorously determining stability and feasibility naturally becomes a relevant area of further research.

5.2.3 Exploiting redundant joint motion

The goal with the null-space movements was to penalize movements in certain joints to so that joints closer to the center of the robot move less. Figure 4.20 and 4.21 show this to have been successful. There is clear difference between the figures, showing that when optimizing for the secondary null-space objective the first and especially the second cylinder move considerably less and stay relatively close to their initial amount of extension. In the other case when no secondary objective is applied, all cylinders move about the same.

There are many other ways of utilizing null-space movement [17]. Especially interesting to study further would be an objective that minimizes the position of the center of mass with respect to some desired position. This would probably be more intuitive and certainly more optimal than minimizing with respect to a joint configuration if the goal is to prevent tipping of the robot. We consider this section to have answered the relevant research question posed in the introduction.

5.2.4 Disturbance rejection

The Figures 4.22 and 4.23 show a simulation where the task-space feedback has been disabled, meaning that the inverse kinematics run in open-loop. This results in a significant performance degradation due to disturbances in the form of external forces from gravity, Jacobian linearization and joint velocity integration, as mentioned in 3.3.1.1. The trajectory visibly drifts from the reference and sinks down, presumably due to gravity. An interesting observation from this simulation is that while the open-loop simulation drifts from the reference, it suffers much less from overshoot when changing directions.

Looking at the Figures 4.24 and 4.25 from the simulation with large model error, it can be seen that tracking performance is degraded but it still manages to follow the reference. This degradation is not unexpected since the used parameters are from the maximum of the range for the parameter sweep, which can be quite unrealistic. For example, for cylinder 1 there is about a 40 bar pressure differential over the piston, as seen in Figure 4.26, which means that with a leakage coefficient c_l of 10^{-11} $\mathrm{m}^3/(\mathrm{Pa} \cdot \mathrm{s})$, almost half of a deciliter of fluid per second is leaking past the internal seals, indicating a severe leakage. When testing with and without CLIK, it was found that this had a profound impact on tracking performance, as when it was turned off the simulation failed due to the EE drifting and hitting a cartesian limit.

It seems that CLIK is quite important for overall disturbance rejection capability. It eliminates minor disturbances such as gravity and manages to keep the controller stable when faced with large model errors.

6
Conclusion

This chapter summarizes the most important points and possible further research from this thesis. The aim of the thesis and the research questions from the introduction are also revisited.

The main conclusion from this thesis should be that an MPC with CLIK shows promising results in simulations that indicate that it is likely possible to achieve good tracking performance on a four-link hydraulic manipulator with the aforementioned control scheme. The linear scheduled MPC model in combination with inverse kinematics feedback is able to handle disturbances and model-mismatch that arises from non-linear hybrid hydraulic dynamics and external forces. The simulations indicate that it is valid to linearize a non-linear hydraulic model and identify dominant dynamics of non-linear systems by using a combination of sensitivity analysis methods.

All conclusions in this report are however subject to uncertainty since no actual physical tests were conducted. Hydraulic systems are generally difficult to model and as discussed, even the non-linear model does not represent the real system exactly, meaning that there for certain exists modeling error. For accurate modeling of hydraulic systems, detailed specifications of component dimensions and performance has to be provided or found experimentally to eliminate uncertainty. Another improvement to the methodology would be to test control structures early and collect data on cylinder velocities, valve flows and pressure response of cylinders and pump.

6.1 Reflection on research questions, aims and scope

Looking back at the introduction, it can be concluded that all aims have been achieved, however with varying degree of completion. The largest challenge was the formulation of hydraulic and kinematic models, as well as interpreting the sensitivity analysis. Another challenge was addressing the constraints set in the scope of the project, one being that the only measurable quantity is the extension of the rod. The research questions were brought up in the discussion and are now also briefly answered.

Which dynamics have the largest impact on tracking performance for the robot?
From the methodology and resulting sensitivity analysis it is believed that the valve is the key component in the system, as it is also the only input available. Furthermore, if the valve can be assumed to decouple cylinder dynamics from the pump, it is the most important thing to model in order to achieve sufficient performance. However, there is evidence that cylinder leakage and external force might have non-linear interaction effects as shown in Morris analysis. While there has been previous research on the impact of friction on piston motion, it does not appear as a significant factor in this system. These factors should be investigated in further research.

Can the hydraulic actuators be decoupled?
Inconclusive, while the simulation uses an ideal feed-forward flow-controlled valve, it is not independent of pump pressure P_s. This means that there will be coupling between valves, and thus, the actuators of the system. However, assuming that the valves are pressure compensated, it is theoretically possible to decouple cylinder dynamics. Because no relevant simulation or tests on the physical system have been performed, the question of decoupling cannot be definitively answered.

Will an MPC be able to reject disturbances?
Yes, results indicate that an MPC is able to handle sensor noise, and that this capability is enhanced when including a penalty on the rate of change of the input. However when dealing with larger model errors and disturbances such as external forces due to gravity, MPC struggles by itself and requires CLIK to counter drift or in some cases to function at all.

How should references that exploit redundant joint motion be generated?
By utilizing the null-space of the Jacobian it is possible to, in addition to motion tracking, optimize for a second objective that achieves a desired behavior by adding movements that do not alter the position of the EE.

Can the reference tracking and reference generation problems be combined and solved by one MPC?
Inconclusive since this was not explored extensively, but it will likely require NMPC due to non-linearity in calculating the Jacobian and robot dynamics. A linear MPC may be possible with a linearized joint-task-space relation and if limiting dynamics to linear models. Utilizing two MPC:s with the same model, as in [29], is generally not relevant for the goal of this thesis as that solves a different problem.

6.2 Further research

The following topics of future research were identified in the discussion.

- Implement lag or smoothing to operator controls to handle overshoot during direction changes of the end effector.

- Investigate NMPC with non-linear dynamics model to optimize motion tracking.
- Investigate NMPC with alternative costs such as energy consumption.
- Investigate a unified single MPC or NMPC that handles both inverse kinematics and motion control.
- Conduct a rigorous stability and feasibility study of the closed-loop control scheme.
- Implement center-of-mass-based null-space objective.
- Investigate filtering or estimation methods to handle sensor noise, disturbance and model mismatch.
- Investigate other controller models that can utilize pressure sensors.
- Conduct physical tests to verify model assumptions made in Section 3.2 such as actuator decoupling as well as overall tracking performance.

Bibliography

[1] Martin Pär-Love Palm. Inverse kinematics control of a four link hydraulic manipulator [unpublished manuscript]. Husqvarna Construction, 2019.

[2] J B Rawlings, D Q Mayne, and M Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017.

[3] Manring N.D. *Hydraulic Control Systems*. Wiley, 2020.

[4] W.S. Owen and E.A. Croft. The reduction of stick-slip friction in hydraulic actuators. *IEEE/ASME Transactions on Mechatronics*, 8(3):362–371, 2003.

[5] S. Andersson. 4 - friction and wear simulation of the wheel–rail interface. In R. Lewis and U. Olofsson, editors, *Wheel–Rail Interface Handbook*, pages 93–124. Woodhead Publishing, 2009.

[6] Shahram Tafazoli Bilandi. *Identification of frictional effects and structural dynamics for improved control of hydraulic manipulators*. PhD thesis, University of British Columbia, 1997.

[7] Michael Ruderman. Full-and reduced-order model of hydraulic cylinder for motion control. In *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, pages 7275–7280. IEEE, 2017.

[8] Magnus Bengtsson and Ola Örsmark. Modeling and control of an hydraulic actuator. *MSc Theses*, 1999.

[9] Shahram Tafazoli, Clarence de Silva, and Peter Lawrence. Tracking control of an electrohydraulic manipulator in the presence of friction. *Control Systems Technology, IEEE Transactions on*, 6:401 – 411, 06 1998.

[10] Levi H Manring and Noah D Manring. Mapping the efficiency of a double acting, single-rod hydraulic-actuator using a critically centered four-way spool valve and a load-sensing pump. *Journal of Dynamic Systems, Measurement, and Control*, 140(9):091017, 2018.

[11] Rodrigo Szpak, José Roberto Branco Ramos Filho, Henri Carlo Belan, and Victor Juliano De Negri. Theoretical and experimental study of the pressure behavior on hydraulic positioning systems. In *Proceedings of COBEM 2009 20th International Congress of Mechanical Engineering*, Gramado, RS, Brazil, November 2009. ABCM.

[12] Rico H Hansen, Asger M Iversen, Mads S Jensen, Torben O Andersen, and Henrik C Pedersen. Modeling and control of a teletruck using electronic load sensing. In *Engineering Systems Design and Analysis*, volume 49170, pages 769–778, 2010.

[13] Noah D Manring. Mapping the efficiency for a hydrostatic transmission. *Journal of Dynamic Systems, Measurement, and Control*, 138(3):031004, 2016.

[14] Bobo Helian, Pauli Mustalahti, Jouni Mattila, Zheng Chen, and Bin Yao. Adaptive robust pressure control of variable displacement axial piston pumps with a modified reduced-order dynamic model. *Mechatronics*, 87:102879, 2022.

[15] Peter Achten and Sjoerd Eggenkamp. Barrel tipping in axial piston pumps and motors. In *Proc. 15th Scandinavian Int. Conf. Fluid Power (SICFP)*, volume 144 of *Linköping Electronic Conference Proceedings*, pages 381–391, Linköping, Sweden, 2017. Linköping University Electronic Press.

[16] M W Spong, S Hutchinson, and M Vidyasagar. *Robot Modeling and Control*. Wiley, 2020.

[17] Carlos Canudas de Wit, Bruno Siciliano, and Georges Bastin. *Theory of robot control*. Springer Science & Business Media, 2012.

[18] The MathWorks Inc. Moore-penrose pseudoinverse. `https://se.mathworks.com/help/matlab/ref/pinv.html`, Last accessed on 2026-05-06.

[19] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir. Inverse kinematics techniques in computer graphics: A survey. *Computer Graphics Forum*, 37(6):35–58, 2018.

[20] Leopold Armesto. Kinematic control of robot manipulators with null space policies 1 | robotic systems, 2021. `https://www.youtube.com/watch?v=GMnO6qntsYc`, Last accessed on 2026-05-05.

[21] Bertrand Iooss and Paul Lemaître. *A Review on Global Sensitivity Analysis Methods*, pages 101–122. Springer US, Boston, MA, 2015.

[22] Jim W. Hall, Shawn A. Boyce, Yueling Wang, Richard J. Dawson, Stefano Tarantola, and Andrea Saltelli. Sensitivity analysis for hydraulic models. *Journal of Hydraulic Engineering*, 135(11):959–969, 2009.

[23] John Butcher. Runge-kutta methods. `http://www.scholarpedia.org/article/Runge-Kutta_methods`, Last accessed on 2026-05-06.

[24] Xiaolian Yu, Zhaoqiang Wang, Zhifeng Li, and Chongzhi Sun. Design and kinematics analysis of four-linkage excavator. In *Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence*, pages 605–610, 2019.

[25] Mingjie Feng, Jianbo Dai, Wenbo Zhou, Haozhi Xu, and Zhongbin Wang. Kinematics analysis and trajectory planning of 6-dof hydraulic robotic arm in driving side pile. *Machines*, 12(3):191, 2024.

[26] Rabab Benotsmane and György Kovács. Optimization of energy consumption of industrial robots using classical pid and mpc controllers. *Energies*, 16(8), 2023.

[27] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[28] TJ Ri, CM Kang, and KS Ri. A method to avoid the stick-slip in a large hydraulic cylinder under variable load using a balanced valve. *Global Academic and Scientific Journal of Multidisciplinary Studies (GASJMS)*, 3(10):58–71, 2025.

[29] Juncheng Li, Maopeng Ran, Han Wang, and Lihua Xie. Mpc-based unified trajectory planning and tracking control approach for automated guided vehicles. In *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, pages 374–380. IEEE, 2019.

[30] Ko Ayusawa, Wael Suleiman, and Eiichi Yoshida. Predictive inverse kinematics: optimizing future trajectory through implicit time integration and future jacobian estimation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 566–573. IEEE, 2019.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY