Improved crane robot control for confined space in-wing teleoperated inspection

Wade Marquette

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:
Santosh Devasia, Chair
Joseph Garbini
Lance McCann

Program Authorized to Offer Degree:
Department of Mechanical Engineering

University of Washington

Abstract

Improved crane robot control for confined space in-wing teleoperated inspection

Wade Marquette

Chair of the Supervisory Committee:
Santosh Devasia
Department of Mechanical Engineering

Visual inspection of confined spaces such as aircraft wings is ergonomically challenging for human mechanics. This dissertation presents a novel crane robot that can travel the entire span of the aircraft wing, enabling mechanics to perform inspection from outside of confined spaces. However, teleoperation of the crane robot can still be a challenge due to the need to avoid obstacles in the workspace and potential oscillations of the camera payload. Therefore, a teleoperation assistance is developed using the differential flatness of the crane-robot dynamics for designing reduced-oscillation, collision-free time trajectories of the camera payload for use in teleoperation. Autonomous experiments verify the efficacy of removing undesired oscillations while teleoperation experiments demonstrate that the controller eliminated collisions when 12 participants performed an inspection task with the use of proposed trajectory selection when compared to the case without it. Moreover, even discounting the failures due to collisions, the proposed approach improved task efficiency.

Extended to a crane-robot-based active vision system for in-wing confined space inspection, the crane robot enables teleoperated manipulators to navigate around in-wing systems while providing a dynamic view of potential collisions to the teleoperator. Sampling-based techniques, such as the model predictive path integral control (MPPI), are well suited to optimize the input to the active vision system for precisely tracking the teleoperated robot while maintaining constraints and avoiding obstacles. However, correctly selecting the input-

sample distribution for MPPI is often difficult for general systems, which can be addressed by iteratively optimizing over long preview-time horizons, increasing computational load and inducing tracking delays. Therefore, an output-sampled MPPI (oMPPI) is developed using inversion-based control, where the input to the approach becomes the desired output (reference position) for general invertible nonlinear systems, which enables sampling of the system output rather than the input. An advantage of the proposed oMPPI is that the selection of the reference output's sample distribution can be based on the desired output of the system, such as the trajectory of the teleoperated robot the active-vision case. Additionally, optimality with the proposed oMPPI is proved and conditions are provided for feasibility of the inverse. Experimental results show that oMPPI enhances sampling efficiency over MPPI by reducing the required preview time. Furthermore, oMPPI enables precision tracking by reducing both oscillations and tracking errors during teleoperation with the active vision system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ACTIVE VISION SYSTEM: A robotic camera system that adjusts its camera dynamically to assist teleoperation.

(AIRCRAFT WING) BAY: An enclosed segment of internal wing structure.

(AIRCRAFT WING) RIB: A structural wall within the wing that separates adjacent bays.

(AIRCRAFT WING) STRINGER: A structural beam adding stiffness over the length of the wing.

(AIRCRAFT WING) STRINGER CHANNEL: The channel formed between adjacent stringers.

DIFFERENTIAL FLATNESS: A property of nonlinear systems where states and inputs are functions of outputs and and a finite number of output derivatives.

FIDUCIAL MARKER: A marker used to measure position and/or orientation in vision systems.

GANTRY CRANE: A crane with a supported horizontal beam capable of linear motion along its tracks.

INVERSE MODEL: A model used to compute required inputs to achieve a desired system output.

MANIPULATOR: A robotic arm capable of interacting with and manipulating objects.

PREVIEW TIME: The time horizon over which a system plans future inputs and trajectories.

SEMI-AUTONOMOUS CONTROL: A control system that automates parts of the operation while relying on human inputs to complete tasks.

SNAP-CONTINUOUS TRAJECTORY: A trajectory continuous up to the fourth derivative.

TELEOPERATION: The remote control of a robot by a human operator.

ACKNOWLEDGMENTS

I am deeply thankful to Professor Santosh Devasia for his mentorship and guidance throughout my graduate journey and for shaping my development as a researcher. His positive attitude, patience, and expertise shaped my experience and development, encouraging me to strive for excellence in my work. I sincerely thank the remaining members of my committee for their valuable advice and feedback, which helped in refining and completing my research. I extend my thanks to my colleagues and friends in the Precision Controls Lab and Boeing Advanced Research Collaboration at the University of Washington, who have provided support and assistance throughout. Finally, I would like to thank my family and friends for their unwavering encouragement and support.

DEDICATION

to Athena

Chapter 1
INTRODUCTION

1.1 Motivation

1.1.1 Crane robot teleoperation

Confined-space inspection is a major aspect of aerospace manufacturing and maintenance, especially inside aircraft wings (where fuel is stored), which are ergonomically challenging, hazardous environments to work in [1]. For example, mechanics need to don protective suits and respirators for safety in such spaces, which makes the work cumbersome [2]. Moreover, ensuring safety requires regular check-ins from an outside partner [3]. These difficulties of operating in confined spaces motivate the development of robotic solutions that allow mechanics to perform their work from outside the confined space. A challenge with typical robotic inspection solutions is that they require repeated time-consuming installation and removal for each of the many separated internal structure segments (bays, see Fig. 1.1) of the wing in commercial aircraft architectures. The crane robot avoids the problem of installation/removal in each of the separate bays by designing a compact robot that can fit and move inside the channel formed between adjacent stringers, and thereby, access the entirety of the wing operating akin to a gantry crane with the suspended camera as the payload. The camera can be retracted to cross the ribs that separate adjacent bays, enabling movement through the entire wing without re-installation.

However, crane robot swing dynamics challenge its use in confined space teleoperation. While teleoperating the crane robot, uncontrolled swing dynamics are difficult to manually compensate for and can lead to collision. An additional challenge is that teleoperation (which takes advantage of human expertise to perform complex tasks) can be slow since it is difficult for humans to manage multiple tasks such as (i) inspection of the space and (ii) avoiding

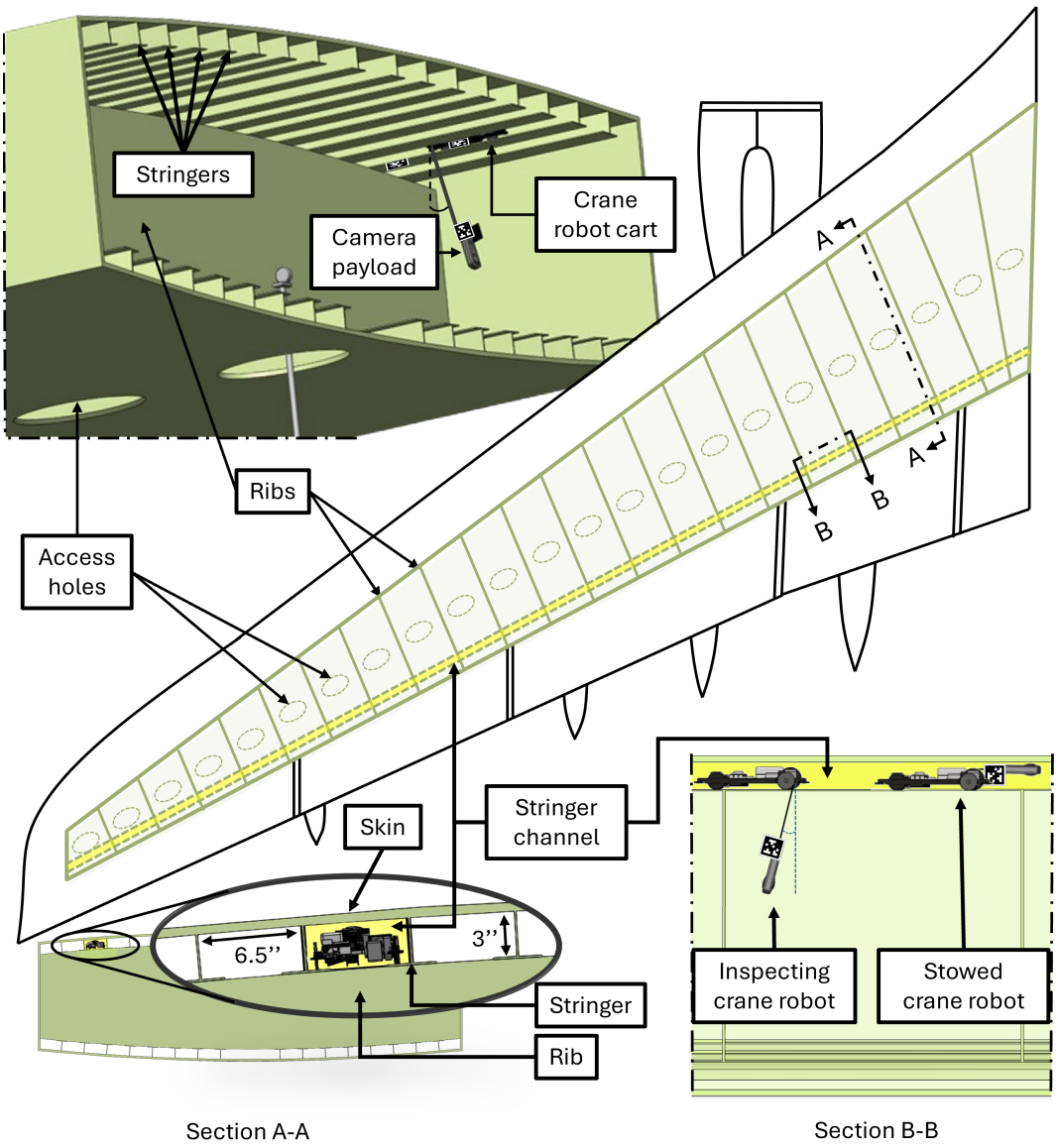


Figure 1.1: The crane robot performs inspection along the entire wing by traversing in a single stringer channel. (Top-left) Partial view of an aircraft wing bay with crane robot moving inside the channel between two stringers, suspending a camera payload for inspection. The wing bays are partitioned into separate spaces by ribs and each space is typically accessed through narrow access holes. (Center) A two-dimensional schematic of the wing with a sample stringer channel highlighted in yellow, spanning the length of the wing, which provides access (over the ribs, with the camera stowed) between adjacent bays. (Bottom left, Section A-A) Cross section with the stowed crane robot in the stringer channel (highlighted in yellow) formed below the upper skin and between adjacent stringers enabling passage over the ribs. (Bottom right, Section B-B) Crane robot in the stringer channel (yellow highlight).

obstacles – especially, in in complex confined spaces [4]. This motivates research for effective teleoperation control to ensure the crane robot operates safely and efficiently in confined spaces.

1.1.2 Crane robot active vision system

Teleoperated manipulators can traverse the wing similar to the crane robot by stowing within a motorized cart that drives between adjacent bays through stringer channels located on the lower skin of the wing, illustrated in Fig. 1.2. However, teleoperated manipulators need to navigate around in-wing obstacles such as pipelines and structural components, to carry out tasks, and providing operators with an effective visual perspective in such confined spaces is challenging. Cameras placed at access holes in the center of wing bays cannot capture manipulator configuration or proximity to obstacles. Static camera placements within confined spaces either require (a) numerous camera placements for sufficient views to navigate around obstacles or (b) frequent task-specific relocation of cameras, both of which involve substantial installation and removal effort before and after use, making them not feasible for in-wing manufacturing [5]. Moreover, relying on cameras attached to teleoperated robots leads to disorienting effects for long-term operations [6]. To address these challenges, the crane robot is deployed as an active vision system to autonomously track the lower teleoperated manipulator, demonstrated in Fig. 1.2.

The active vision system on the crane robot needs to simultaneously track the teleoperated robot while solving complex optimization problems to provide optimal camera viewpoints and incorporate other constraints, such as avoiding obstacles and meeting actuator limits. By posing the optimization problem in a model predictive control (MPC) framework, active vision systems can simultaneously plan active vision dynamics and motion while minimizing task-specific costs. However, solving these optimization problems can be computationally challenging with gradient-based MPC approaches when there are a large number of cost terms and constraints in the optimization criterion [6, 7]. This motivates the use of gradient-free, sampling-based approaches to MPC, such as model predictive path integral

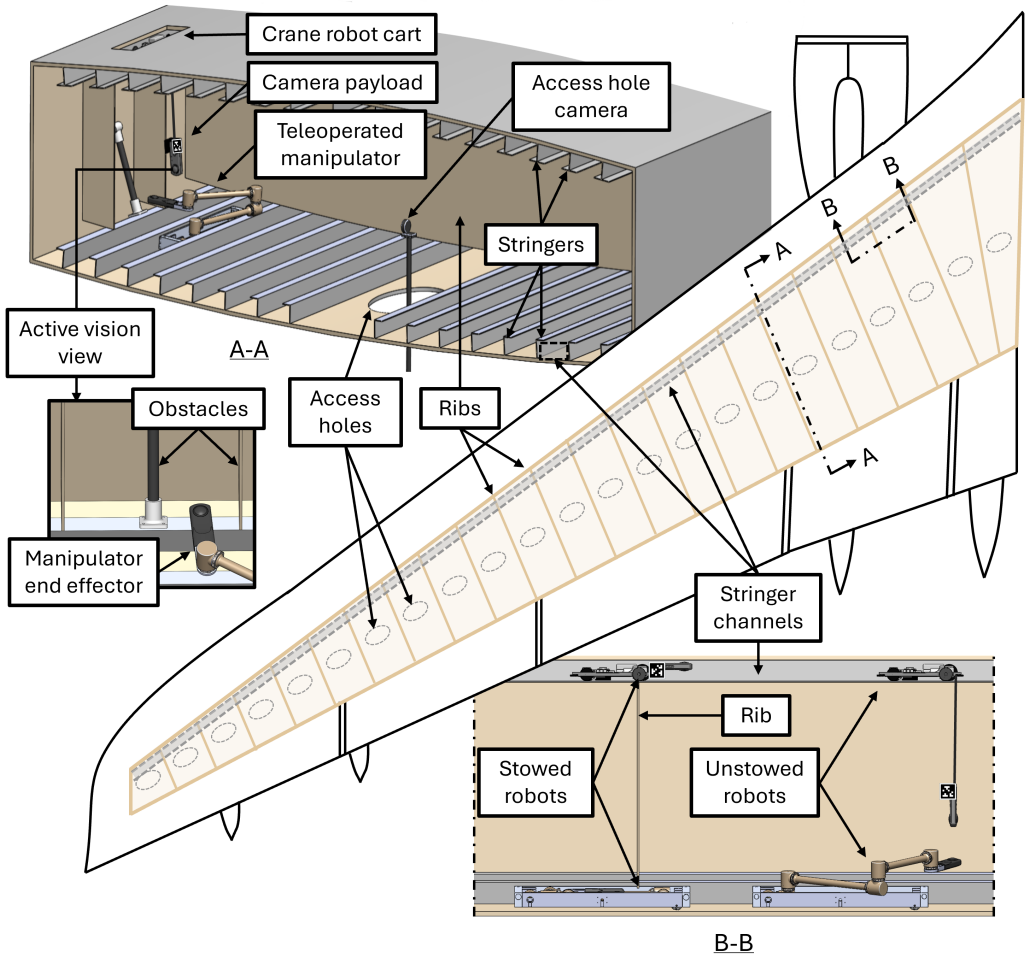


Figure 1.2: The active vision crane robot provides a view of a teleoperated manipulator's end effector to assist navigation near obstacles inside aircraft wings. (Center) A two dimensional (2-D) schematic of the internal structure of an aircraft wing demonstrating how stringer channels span the length of the wing. (Top left, Section A-A) A cross section of the wing where the active vision crane robot provides a view of the teleoperated manipulator end effector navigating near obstacles. (Bottom right, Section B-B) The stowed crane robot and manipulator move between bays past ribs through stringer channels. The robots unstow in a bay to perform manufacturing tasks.

control (MPPI), which solves for optimal input sequences while handling general nonlinear dynamics [8] and non-smooth constraints [9], while significantly reducing the computation time through parallelized computing [10].

MPPI is challenged by correctly initializing the input distribution for avoiding undesirable local minima [11]. Therefore, it can be difficult to achieve precision tracking of the desired output through MPPI. While the original MPPI formulation uses warm-starts to optimize the mean of the input sequence, based on results from the previous control timestep, it requires a long optimization horizon since it does not leverage available output information. Although optimizing over extended time horizons can help reduce the risk of selecting local minima, these methods add to the computational burden, which scales proportionally to preview time, especially considering limitations of onboard robotic edge computing [12,13].Moreover, in tracking tasks, such as active vision systems, optimizing active vision trajectories over long preview horizons introduces corresponding delays in the vision feedback to the teleoperator, which can increase error and reduce efficiency and challenges its use in the crane robot active vision system. This need to improve the output-tracking precision with MPPI, e.g., the tracking of the teleoperated manipulator with the active vision system, motivates the current effort.

1.1.3 Research overview

The focus of this research is to enhance confined space robotic teleoperation for robots capable of traversing the wing through stringer channels. This is achieved by improving the control of the crane robot for (i) teleoperated inspection and (ii) autonomous tracking of teleoperated manipulators as an active vision system. This focus leads to the formulation of the following research questions (RQs) and subsequent main contributions (MCs).

1.2 Research questions and main contributions

1.2.1 RQ1: How can the crane robot be teleoperated for in-wing inspection without sway?

Operators control the crane robot by sending horizontal and vertical camera payload velocity commands, ultimately positioning the crane robot's camera payload to perform in-wing inspection tasks. Nonlinear swing dynamics are excited through large velocities and accelerations required to perform time-efficient tasks, coupling the horizontal and vertical input forces and output camera positions. Therefore, the objective of RQ1 is to investigate if an improved teleoperated control can be derived for the crane robot.

1.2.2 RQ2: Can the crane robot autonomously track in-wing teleoperated manipulators?

To enhance operator navigation and situational awareness during confined space manipulator teleoperation, the crane robot can autonomously track a teleoperated manipulator as an active vision system, providing additional views not achievable with external cameras at the access hole. However, the crane robot must compensate for its dynamics while tracking the manipulator and avoiding collision given limited preview of the operator's motion. Therefore, the objective of RQ2 is to investigate whether the crane robot can autonomously track teleoperated manipulators using control strategies that manage swing dynamics and ensure safe tracking near obstacles.

1.2.3 Research question connections

Research questions RQ1 and RQ2 both motivate control improvements of the crane robot to enable remote mechanic in-wing operations. RQ1 motivates control improvements in teleoperation of the crane robot, while RQ2 motivates automatic control improvements of the crane robot to assist in the teleoperation of the other confined space robots.

1.3 Main contributions

1.3.1 MC1: Semi-autonomous teleoperation using differential flatness of a crane robot

MC1 answers RQ1 by using the differential flatness property of the crane-robot dynamics for designing reduced-oscillation, collision-free semi-autonomous control. Specifically, the differential flatness property is used to design reduced-oscillation, collision-free time trajectories of the camera payload [14]. The resulting semi-autonomous teleoperation controller allows the operator to directly specify camera payload trajectories while autonomously avoiding undesired oscillations and potential collisions.

1.3.2 MC2: Output-sampled model predictive path integral control for a crane robot

MC2 answers RQ2 by developing theory to support the use of output-sampled predictive path integral control (oMPPI) for active vision crane robot control, which enables efficient tracking of teleoperated manipulators. The proposed solution of oMPPI first augments the system with its inverse to generate inputs u that account for dynamics and achieve accurate output y tracking, and secondly applies the MPPI algorithm to the augmented system. A schematic comparison of the MPPI and oMPPI algorithm is presented in Fig. 1.3(a) and Fig. 1.3(b), respectively. This enables the use of the MPPI sampling methodology for the full system; however, with the new input (of the full system) being reference output trajectories $\tilde{u} \triangleq y_d$. The advantage of the proposed method is that it allows for direct design of sample means from desired output trajectories and output constraints. A significant challenge lies in ensuring that sampled output trajectories are feasible, i.e., trackable by the system. Inversion theory provides quantifiable feasibility conditions for tracking, in terms of sufficient smoothness of the desired outputs; therefore, a series of pre-filters are proposed to sufficiently smoothen the sampled reference trajectory, obtained from the teleoperated robot, prior to application to inverse in the augmented system.

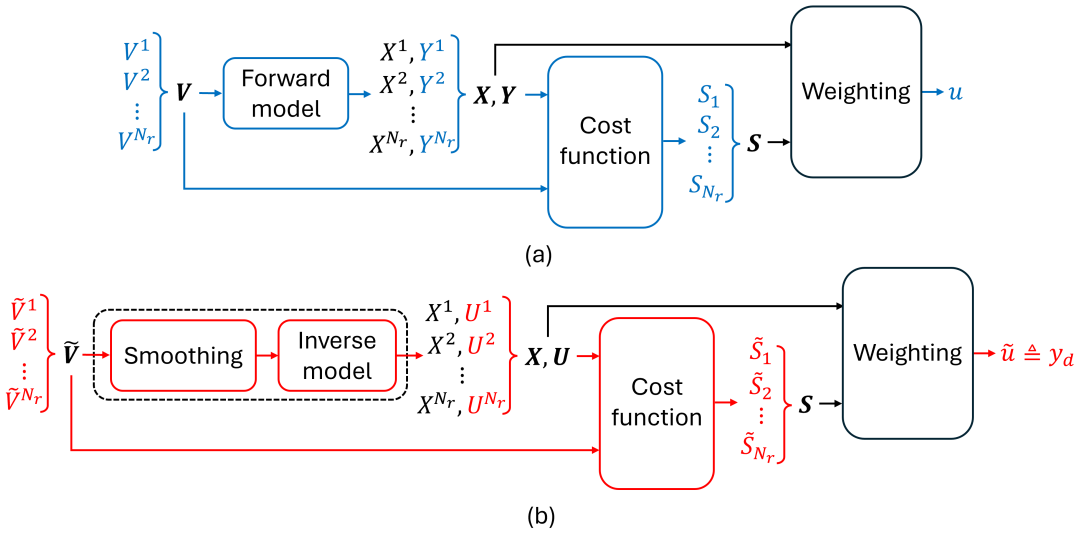


Figure 1.3: Comparison between the MPPI (top) and oMPPI (bottom) algorithms. (a) The MPPI algorithm samples N_r noisy input sequences $V \sim \mathcal{N}(U, \Sigma)$, passes them through the forward model to find the corresponding states X and outputs Y, and weights them based on costs S to find an optimal input u. (b) The oMPPI algorithm samples N_r noisy output reference sequences $\tilde{V} \sim \mathcal{N}(\tilde{U}, \Sigma)$, passes them through a smoothing function to ensure feasibility and an inverse model to find the corresponding inputs U and states X, and weights them based on costs S to find an optimal output reference $\tilde{u} \triangleq y_d$

.

1.4 Document overview

The remainder of this document is organized as follows. Chapter 2 reviews related works of the design and control of the crane robot for in-wing operations. Chapter 3 describes the crane robot's design and components. Chapter 4 presents (MC1) deriving and experimentally validating a semi-autonomous teleoperation for in-wing inspection [15]. This work has been accepted for publication in IEEE Robotics and Automation Letters (IEEE RA-L) [16]. Chapter 5 discusses (MC2) where oMPPI is developed and experimentally applied to the crane robot active vision system for autonomous tracking of a teleoperated manipulator. This work has been submitted for journal publication [17]. Chapter 6 concludes the work and presents a discussion of future research and applications. The research questions and

main contributions are summarized in Fig 1.4.

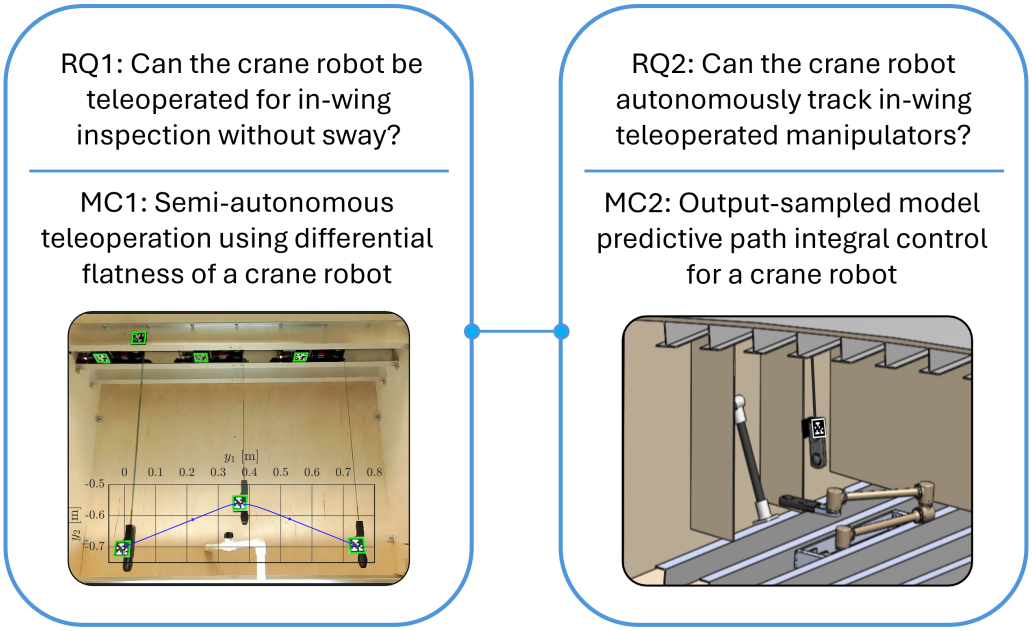


Figure 1.4: Summary of research questions and main contributions.

Chapter 2
RELATED WORKS

2.1 Design of robots for confined spaces

2.1.1 Confined space robots for aircraft wings

Currently available confined space robotic solutions cannot access the entire wing during manufacturing of the commercial aircraft architecture. For example, conventional manipulator-based, continuum-type, and snake robot architectures for aircraft confined space inspection [18–22], fastener installation [23], hole cleaning [24], and multi-tasks [25, 26] within aircraft wings cannot move between bays through the ribs over the span of the wing and require manual installation and removal for each bay.

Although some prior methods have been suggested for navigating between adjacent bays, they are unsuitable for the current work. Movement between bays is achieved with the Eeloscope [2] by swimming through the holes in the ribs when fuel is present inside the aircraft wing tanks. However, this solution is not applicable in the absence of fuel, e.g., during initial aircraft manufacturing. Another approach is using mobile robots that drive through rib cutouts in aircraft wings if the driving surfaces are smooth [27], but this solution does not apply to larger aircraft with uneven surfaces due to stiffening stringer structures on the inner-skin. The crane robot overcomes the challenge of traversing separated bays by using a small cross-sectional area to traverse the bays through narrow (6.5 in $\times$ 3 in) stringer channels that span the entire wing.

Manipulators traversing bays through stringer channels proposed in the active vision system add the versatility of manipulator architectures are capable of performing multiple tasks [25]. Deploying the crane robot active vision can improve task performance in complex environments, as it provides situational visual feedback in confined spaces for operators [28].

Prior work has demonstrated that third-person aerial views can improve robotic teleoperation performance [29]. Therefore, an active vision crane robot is proposed to provide situational views for operators from channels formed by adjacent stringers on the wing bay ceiling, traversing adjacent bays with the manipulator. These aerial views provide information on the configuration of the teleoperated manipulator and proximity to in-wing systems.

2.1.2 Confined space robots for narrow spaces

Previous works have introduced robots to move in narrow spaces such as pipes and channels, e.g., snake [30, 31] and inchworm [32] robots. However, these designs rely on the entire cross section to be continuous, but the stringer channels tend to be open away from the skin. Therefore, the crane robot presented here uses wheels to stay inside the stringer channel, similar to wheeled pipe crawling robots [33–36]. Moreover, the crane robot exploits the the opening in the stringer channel to suspend a camera via a pulley mechanism to perform inspection tasks, as shown in Fig. 1.1.

2.2 MC1: Semi-autonomous teleoperation using differential flatness of a crane robot

2.2.1 Semi-autonomous control

The crane robot (with controls to position on the stringer channel and the camera position) is analogous to a variable-length gantry crane [37]. Therefore, it shares similar challenges in teleoperation as industrial crane systems, where the operator controls both the cart position and the cable length independently [37]. This conventional control approach, where the operator specifies trajectories that do not cause undesired oscillations of the payload, can be challenging and require extensive training to learn how to manage the gantry-crane dynamics.

To avoid such challenges in teleoperation, this work develops a semi-autonomous teleoperation control for the crane robot [38]. Semi-autonomous control has been shown to make teleoperation easier in when applied to manipulation, where the operator specifies a reference position and the manipulator plans subsequent movement accounting for collision

avoidance [39]. Adapted to the crane robot, a human operator only specifies a reference point to generate trajectories for the camera payload, and handling control complexities such as obstacle avoidance and unwanted oscillations (due to payload dynamics) is managed autonomously to assist teleoperation.

2.2.2 Differential flatness of a gantry crane

To design the teleoperation assistance, the current work applies crane control techniques. Input shaping is a widely used method for reducing residual oscillations in crane positioning [40–42]. However, input shaping does not ensure payload trajectory tracking, which is important near obstacles in confined spaces. Alternatively, gantry cranes have been shown to be differentially flat, allowing all states and inputs to be represented as functions of the output and its time derivatives [43]. This work leverages the differential flatness property of the crane-robot dynamics as the basis of the teleoperation assistance where camera payload coordinates are considered as the output [14, 44, 45]. Since all states and inputs can be represented as a function of the flat outputs and their time derivatives in a differentially flat system, previous works have shown that undesired payload oscillation resulting from disturbances can be rejected during autonomous trajectory tracking [46]. In addition to crane applications, differential flatness has been widely applied to a variety of systems, including cable-suspended UAV path planning [47], UAV-UGV cooperative landing [48], and aerobatic trajectories of VTOL aircraft [49]. These autonomous applications are based on pre-defined trajectories or objectives but do not address real-time trajectory generation to assist avoiding obstacles during human teleoperation. Specifically, the proposed crane-robot assistance modulates the operator's reference input to avoid collisions, and additionally, since the trajectory is tracked accurately, the approach also reduces uncontrolled oscillations.

Differential flatness has been applied to teleoperated crane control. Specifically, sufficiently-smooth online S-curve velocity trajectory generation using differential flatness has been investigated in [50]. However, this approach relies on a fixed-length payload and a linearized model, which may not be sufficiently accurate for crane robot operations, where varying

payload lengths and rapid movements can cause larger swing angles. Therefore, this work develops a trajectory generation approach for the gantry-crane model using the differential flatness of the dynamics, which accounts for both (i) the nonlinearity due to the swing dynamics and (ii) the variable length. Moreover, the trajectory generation modulates the operator's reference inputs to avoid collisions, and since the trajectory is tracked accurately, the approach also reduces uncontrolled oscillations. Thus, the proposed approach enables easier (collision free and oscillation free), semi-autonomous teleoperation of the crane robot, which in turn, improves operator performance during confined-space inspection.

2.3 MC2: Output-sampled model predictive path integral control for a crane robot

2.3.1 Active camera systems for robotic teleoperation

Active camera systems in robot teleoperation provide operators with optimized situational awareness to navigate complex environments, such as those with obstacles, by employing a camera-equipped robot that autonomously follows the teleoperated robot to provide additional external views that improve task performance and reduce operator workload [51]. Such systems have been deployed in a variety of robot configurations, including remote manipulator teleoperation [6, 28, 51], mobile robots [52], and drones [7, 29, 53–55] across a wide range of applications, such as vehicle and drone navigation, pick and place operations [28], surgical robotics [51], and disaster response [54]. Tracking teleoperated manipulators with the crane robot shares similarities with active gantry mounted cameras, controlled by solving real time constrained optimization problems [56].

Active vision systems typically solve complex constrained optimization problems to plan camera motion and optimize viewpoints around teleoperated robots and surrounding obstacles. These optimization problems often include numerous weighted cost terms, such as avoiding collisions with the environment and the teleoperated robot, maintaining target centering, controlling distance and perspective angle, and avoiding occlusions [6, 7]. For example, in [7], drone viewpoint optimization around a teleoperated manipulator utilized

nine weighted cost function terms, while a complex gantry viewpoint optimization for a teleoperated mobile robot in [56] optimized over twenty parameters. Typically, active vision systems utilize gradient-based nonlinear optimization. Instead the computational advantages of MPPI can be leveraged for control [8].

2.3.2 Model predictive path integral control

To autonomously navigate around confined space obstacles to track lower manipulators, the crane robot can be controlled by solving a real-time optimization problem. Robotic systems have employed differential dynamic programming (DDP), which relies on quadratic approximations of the dynamics and cost functions to iteratively solve an optimal control problem over a time horizon [57, 58]. Iterative linear quadratic regulators (iLQR) have also been applied to robotic optimal control problems, where first order approximations of the dynamics lead to more efficient computation [59, 60]. However, both DDP and iLQR struggle to easily handle general state constraints, general costs, and systems with nonlinear dynamics. To address these challenges, modern advances in real-time optimization for control have led to the development of model predictive path integral control (MPPI), a nonlinear, sample-based model predictive control method that leverages parallel computation through graphics processing units (GPUs) for efficient control applications [10]. Although additional GPU hardware and software processes, such as data transfers between the central processing unit (CPU) and the GPU, are introduced, the MPPI algorithm remains scales well, even as system complexity increases. At each timestep of the MPPI algorithm, input sequences are forward-simulated through the dynamics model, weighted by cost, and combined to generate the control input. The MPPI algorithm was proven to optimize over arbitrary forms of nonlinear dynamics through its information theoretic derivation [8].

2.3.3 Output-informed MPPI distributions

The need to use the output to inform the input sampling is well known in MPPI literature. For example, recent advances in MPPI have focused on modifying input sample distributions

to improve output trajectories, e.g., Log-MPPI applies log-normal input distributions to increase exploration and avoid local minima [61], and constrained covariance steering MPPI adjusts sample covariance around probabilistic output constraints, effectively steering the system toward low-cost output trajectories [62]. Several approaches also seek to leverage output constraints to inform input sample means. For example, Rapid-exploring Random Trees (RRT) is used in RRT-MPPI to plan an output trajectory that sets the distribution mean [63], whereas in partially observable navigation problems, distribution means can be shifted toward safer output regions to avoid collision [64]. Some methods further modify the entire input distribution based on output space information. For example, Safety Controlled MPPI applies barrier functions to shift distributions toward safer output regions [65]. Learning-based approaches have also been developed to learn input-trajectory sampling distributions conditioned on environmental features [11, 66]. Similarly, oMPPI benefits from designing desired trajectory distribution means directly based on environmental configurations, such as obstacles, but unlike other methods, it enables direct specification of output sampling instead of input sampling.

Prior works have also explored various ways to integrate feedback into MPPI. Biased MPPI incorporates input sequences generated by tracking controllers [67] whereas Tube-MPPI adds an ancillary controller in parallel with an MPPI controller to bound tracking responses [9]. Robust MPPI (RMPPI) avoids some of the risks with trajectory divergence with Tube-MPPI by embedding feedback directly into the MPPI optimization process through contraction-based control for robust output-trajectory tracking [68]. In contrast, the main advantage of oMPPI is that it enables direct sampling in the output space by concatenating the inverse model with the system model. Moreover, oMPPI enables precision tracking of the selected output trajectory using inversion, and therefore, it allows trajectory stabilization using any available feedback design, including robust feedback controllers [69].

The proposed oMPPI method is an evolution of prior work in [70], which presented the oMPPI framework where sample distributions are designed in the output space This prior work showed that the selection of the reference output's sample distribution (in oMPPI with

inversion) can reduce the number of rollouts and the preview time needed when compared to input sampling (in MPPI without inversion) for an autonomous driving example. The current work develops an oMPPI that integrates both sampling and weighting directly within the output space (rather than in the input space as in [70]), which allows the use of MPPI framework to prove optimality.

2.3.4 Inverse models for oMPPI

Inverse models are used to compute system inputs to exactly follow an optimal trajectory in oMPPI. A variety of methods can be used to develop inverse models such as analytical, physics-based models [71], machine learning techiniques like Gaussian process regression and neural networks [72, 73], and hybrid approaches that learn and compensate modeling errors [74, 75]. Inverse models require sufficiently smooth reference trajectories (i.e., sufficiently differentiable) to compute corresponding inputs for exact tracking. Previous works have demonstrated performance degradation in learned inverse models when provided with insufficient output derivatives [72]. Therefore, to meet feasibility conditions on output trajectories for precision tracking, this paper uses smoothing of the measured reference positions with pre-filters.

Chapter 3

THE CRANE ROBOT

This chapter presents the crane robot description for performing in-wing inspection from stringer channels. This chapter is based on work in [16].

3.1 Crane robot description

To deploy inside the narrow stringer channel formed by the flanges of the stringer and the wing skin, in Fig. 1.1, the crane robot frame is segmented into two smaller pieces which are subsequently attached (once inside the stringer channel) with a threaded rod. This allows installation through the narrow gap of the stringer channel while being wide enough to drive on the flanges as as depicted in Fig. 3.1(a). Once installed, the crane robot drives through the channel on four vertical wheels with one wheel driven by a motor. To account for misalignment between the crane robot and stringer channel as well as variations in the channel width, the crane robot has four side wheels, shown in Fig. 3.1(b), which can contact the stringer webs and help realign the crane robot with the stringer channel.

The crane robot performs inspection tasks by suspending a camera with a polyimide belt into the confined space using a pulley mechanism. To traverse adjacent bays, the camera payload is stowed and released from the channel by wrapping and unwrapping the camera around the pulley. Excess power usage is avoided while the camera is stowed or the robot is parked at an inspection point by equipping motors with worm-gear drives making them non-backdrivable to support the payload weight. The crane robot deploys a wireless 360 camera payload, enabling the operator to perform remote inspection in any direction by panning a tablet application, similar to such use in other applications such as power-line inspection robots [76].

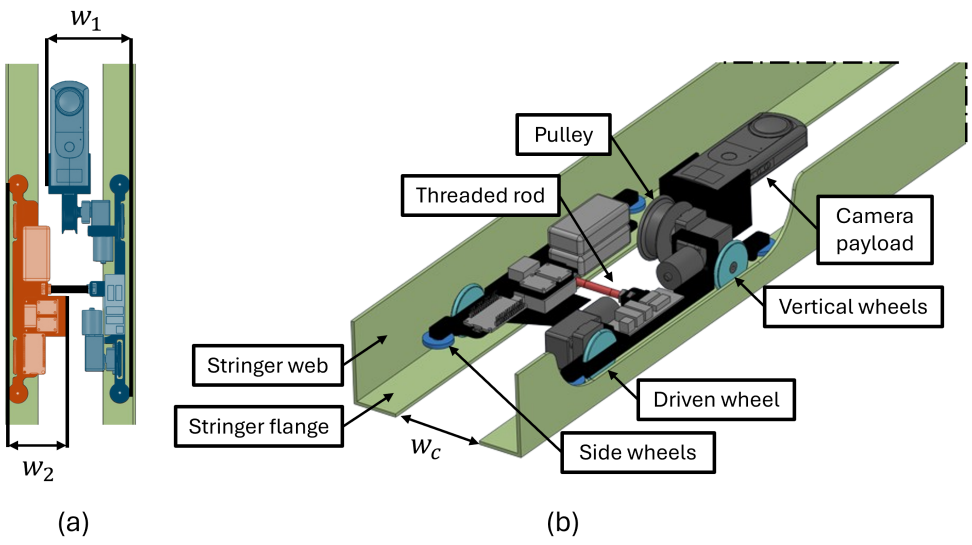


Figure 3.1: Installation of the crane robot in the stinger channel. (a) The crane robot frame segmented in two pieces with widths $w_1 = 3.75$ in and $w_2 = 2.75$ in and thickness less than 2.5 in so each segment can be installed diagonally through the channel opening of width $w_c = 3.5$ in. (b) The installation is completed by connecting the segmented frame with a threaded rod (red). The width of the connected segments is larger than the channel opening width, w_c, enabling the crane robot to drive on the stringer flanges with its vertical wheels (cyan) while its side wheels (blue) contact the stringer web to correct channel misalignment.

3.2 Crane robot teleoperation

Operators receive overall perspective of the environment using an external camera to reduce disorienting effects associated with system movement of the robot's onboard camera [6]. Specifically, an external camera, located at the access hole, provides an external view of both the crane robot and obstacles, delivering situational awareness to the operator shown in Fig. 3.2. The external camera also provides vision-based state and output feedback using fiducial markers (Apriltags [77]) located on the crane-robot's cart and camera payload along with a reference tag located in the confined space for tracking control, e.g., as in [78, 79]. Measurements from the fiducial markers provide swing angle information and provide global-correction updates for crane-position and payload-length estimates from motor encoders.

Operators use camera feedback to visualize the pose of the crane robot and use a joystick, as in industrial crane control [80], to send wireless, horizontal-and-vertical, camera-payload velocity commands.

3.3 Crane robot actuation

3.3.1 Motor current control

The crane robot utilizes a custom two-channel current amplifier to regulate motor current, which specifies motor torque and, in turn, specifies linear force inputs through the cart wheel and camera pulley radii. The schematic in Fig. 3.3 shows the circuit design of the current amplifier, which is controlled by variable voltage inputs $V_{in,1}$ and $V_{in,2}$ to specify the current of the cart motor, $i_{out,1}$, and pulley motor, $i_{out,2}$, respectively. The crane robot is powered by two 12V batteries with voltages V_s^+ and V_s^-, which are input to the current amplifier to power operational amplifiers (op-amps) and motors. A regulator steps down the voltage of V_s^+ to 5V, which powers a microcontroller (MCU). The MCU outputs 3.3V to power two digital to analog converters (DACs), then communicates with the DACs using I2C communication to output $V_{in,1}$ and $V_{in,2}$ between 0 and 3.3V to the current amplifier. The MCU also connects a constant 3.3V voltage line, V_c, to the current amplifier. With connections to the battery voltages (V_s^+ and V_s^-), DAC voltages ($V_{in,1}$ and $V_{in,2}$), and constant voltage (V_c), operation of the current amplifier can be discussed.

Consider the current amplifier channel of the cart motor, which linearly maps an input voltage $V_{in,1} \in [0, 3.3]$ V to drive the cart motor with current $i_{out,1} \in [-1, 1]$ A. A voltage divider with resistors $R_1 = R_2 = 10$ kΩ halves the constant voltage line, V_c, to generate $V_{c/2}$ as

$$V_{c/2} = \frac{R_2}{R_1 + R_2} V_c. \tag{3.1}$$

Two op-amps are configured as voltage followers to buffer the halved constant voltage, $V_{c/2}$, and DAC voltage, $V_{in,1}$. A differential op-amp uses resistors $R_3 = 10$ kΩ and $R_4 = 30$ kΩ to subtract the halved constant voltage, $V_{c/2}$, from the DAC voltage, $V_{in,1}$, shifting the DAC

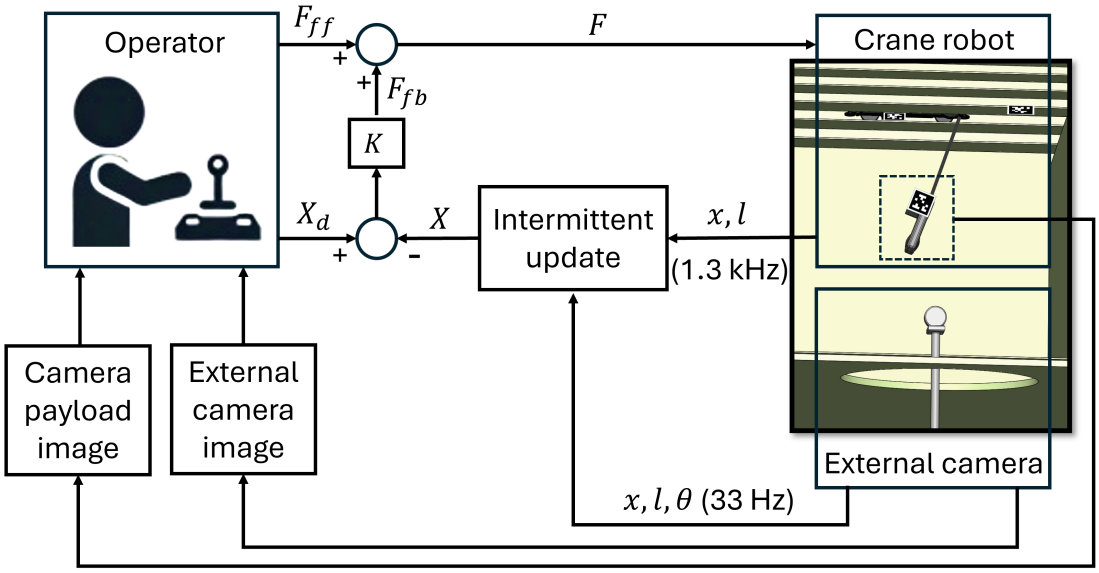


Figure 3.2: Crane-robot's control scheme. The external camera at the access hole measures the cart position, x, the payload length, l, and the swing angle, θ, from the fiducial markers at a sampling rate of 33 Hz, which intermittently updates feedback collected from the motor encoders at a sampling rate of 1.3 kHz to construct the crane robot's states, X. From the operator workstation, the feedforward input, F_{ff}, and desired states, X_d, are specified through a joystick interface by the operator observing the external camera and camera payload image. The combined feedforward force, F_{ff}, and feedback force, F_{fb}, is the applied crane-robot input, F.

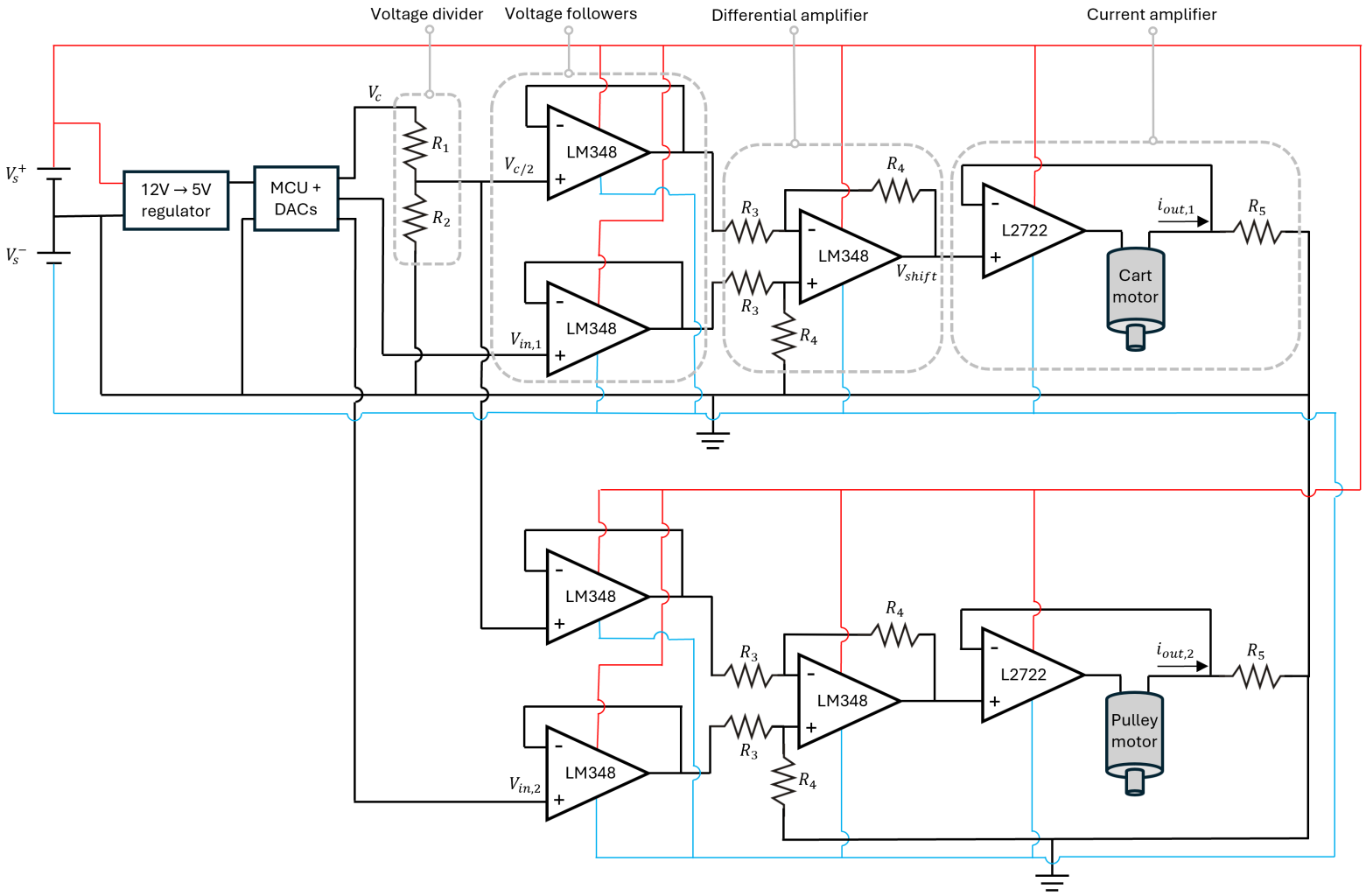


Figure 3.3: Custom two-channel current amplifier circuit schematic. The microcontroller (MCU) commands digital to analog converter (DAC) voltages $V_{in,1}$ and $V_{in,2}$ to control motor currents $i_{out,1}$ and $i_{out,2}$, respectively. Low power components (e.g. an LM348 operational amplifier) can be used for the voltage divider, voltage followers, and differential amplifiers. High power components are required for the current amplifier (e.g. an L2722 operational amplifier and a 5V resistor rated at 5W).

voltage range from $V_{in,1} \in [0, 3.3]$ V to $V_{shift} \in [-1.65, 1.65]$ V as

$$V_{shift} = \frac{R_4}{R_3}\left(V_{in,1} - V_{c/2}\right). \tag{3.2}$$

The shifted voltage, V_{shift}, is input to the current amplifier, which controls motor current, $i_{out,1}$, by regulating the voltage drop across the resistance $R_5 = 5\ \Omega$ (rated at 5W) as

$$i_{out,1} = \frac{1}{R_5} V_{shift}. \tag{3.3}$$

From an input of the DAC voltage, $V_{in,1}$, output current can be computed by combining Eqs. (3.1)-(3.3) as

$$i_{out,1} = \frac{R_4}{R_3 R_5}\left(V_{in,1} - \frac{R_2}{R_1 + R_2} V_c\right). \tag{3.4}$$

The current amplifier channel for the pulley motor is designed equivalently, linearly mapping the DAC voltage, $V_{in,2} \in [0, 3.3]$ V, to pulley motor current, $i_{out,2} \in [-1, 1]$ A. The custom two-channel current amplifier was integrated in a two-layer printed circuit board (PCB) for driving the crane robot's motors, shown in Fig. 3.4.

The circuit design can be validated by commanding DAC voltages to a custom two-channel current amplifier PCB and measuring current through the loaded cart and pulley motors of the crane robot. The nominal gain for each current amplifier channel is 0.61 A/V. DAC voltage to motor current relationships are shown in Fig. 3.5, where linear regression models ($R^2 > 0.99$) demonstrate gains of 0.58 A/V and 0.61 A/V for the cart and pulley motors, respectively. Some variation in gain is expected due to tolerances of the resistive components. With validation of the custom current amplifier circuit, a model for specifying force inputs to the motors can be created.

3.3.2 Motor force inputs

Forces of the cart motor, f_1, and pulley motor, f_2, can be commanded by specifying the corresponding DAC voltages, $V_{in,1}$ and $V_{in,2}$. In general, motor current is proportional to motor torque, making DAC voltage proportional to motor torque through the current amplifier. However, gear drives have high friction, introducing an input deadzone around zero

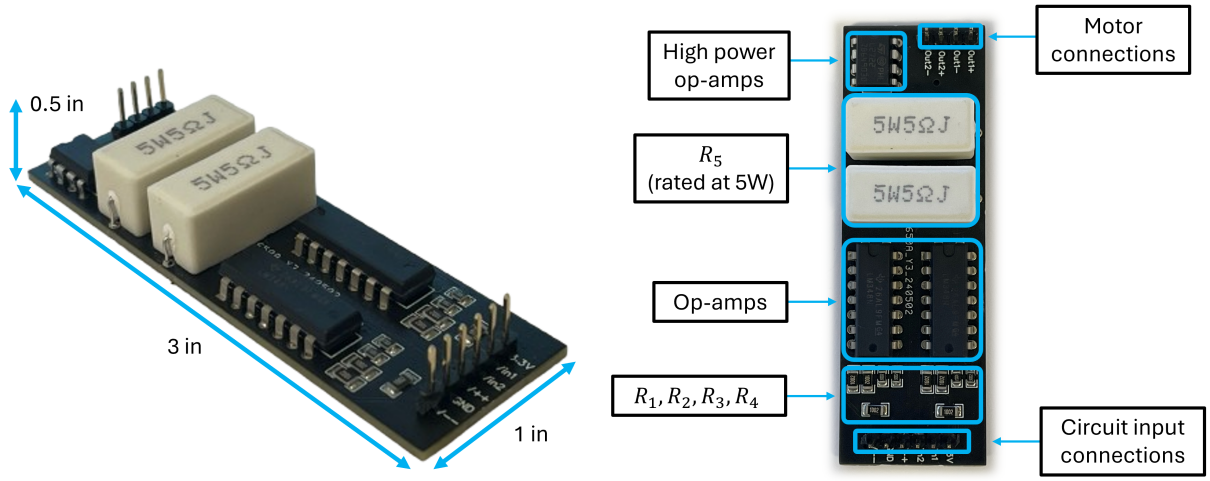


Figure 3.4: A custom current amplifier printed circuit board (PCB) for controlling current in two motors. (a) The PCB fits within a rectangular prism volume with size 0.5 x 1 x 3 in. Six input pins connect to battery voltages (V_s^+ and V_s^-), DAC voltages ($V_{in,1}$ and $V_{in,2}$), and constant voltage (V_c). A ground pin connects to the junction between battery voltages (V_s^+ and V_s^-). Four output pins connect to two motors. (b) Labeled PCB diagram showing components. Operational amplifiers (op-amps) use integrated circuit chips. The high power op-amp requires an additional heat sink (not pictured).

current, where motor torque cannot overcome gearbox friction. On either side of the dead-zone, relationships between DAC voltages and forces are expected to be linear. The following experiment is carried out to fit a model enabling force commands to the motors.

Fig. 3.6 shows a motor with an attached key, designed to hold precision weights at regularly spaced intervals. For motion to occur, the motor needs to output enough torque to counteract the torque induced by the weight, moving above horizontal. DAC voltages are increased just until movement occurs, at which time the motor has output just enough torque to lift the weight. Measurements of DAC voltage and output torque are then recorded. From these measurements, the force inputs to the crane robot, f_1 and f_2, can be computed by dividing by the cart wheel and camera pulley radii. Fig. 3.7 shows the resulting models of the experiments, where piecewise linear expressions ($R^2 > 0.99$) define the DAC voltage,

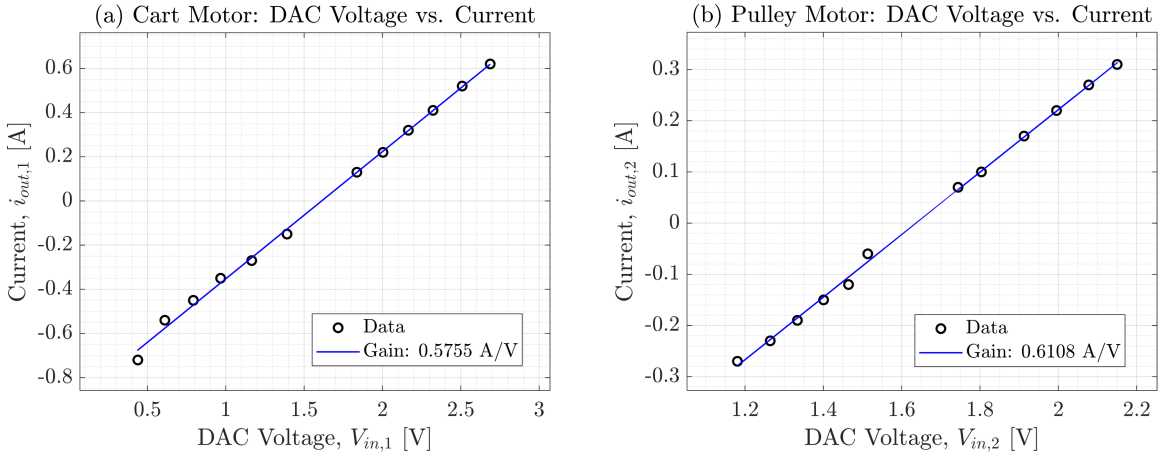


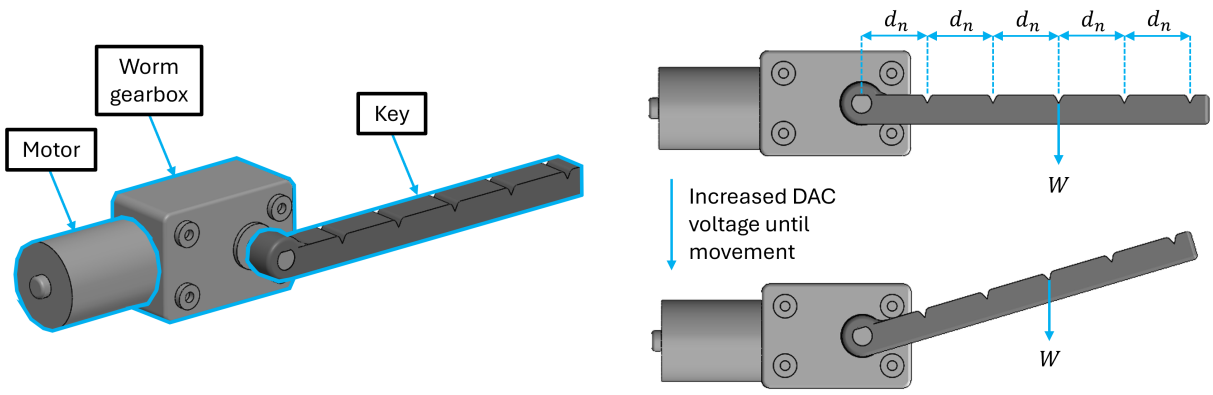
Figure 3.5: Plots relating input digital to analog converter (DAC) voltage to current amplifier printed circuit board (PCB) current output. Compared with a nominal gain of 0.61 A/V, (a) the cart motor shows a gain of 0.58 A/V and (b) the pulley motor has a gain of 0.61 A/V.

V_{in}, to specify given a desired force input, f.

3.4 Crane robot sensing

3.4.1 High frequency feedback

High frequency feedback control allows the crane robot to track motions to cancel undesired oscillations. Shaft-mounted encoders return 12-bit angle measurements for computing cart position, x, and pulley length, l, through SPI communication with the MCU at 1.3 kHz. Encoder counts can be integrated to obtain position information of x and l, and time division provides velocity information of $\dot{x}$ and $\dot{l}$. However, integration of counts is prone to numerical drift, and cart measurements can further drift due to wheel slip. Moreover, swing angle, θ, cannot be measured through encoders. Therefore, a lower frequency visual feedback using fiducial markers from external cameras mounted at the access holes intermittently updates the state estimation at 33 Hz.



(a) (b)

Figure 3.6: Motor with attached key for characterizing force inputs. (a) A motor, worm gear drive, and key designed for holding precision weights at regularly spaced notches. (b) Notches are regularly spaced at distances of d_n. Measurements relating an input digital to analog converter (DAC) voltage to output motor torque can be taken by suspending a weight, W, from a notch and increasing the DAC voltage until rotation above the horizontal configuration.

3.4.2 Low frequency feedback

Global correction updates (i.e., crane robot visual feedback) use three fiducial markers illustrated in Fig. 3.8 to measure cart position, x, camera payload length, l, swing angle, θ, and horizontal and vertical camera payload positions, y_1 and y_2, respectively. Each fiducial marker returns six degree-of-freedom measurements, with three translations and three rotations from the camera. However, marker translation measurements are more stable than rotation measurements from the access hole camera distance, so states and outputs are computed using marker center coordinates and crane-robot kinematics. The reference marker defines the origin at coordinates $(0, 0)$ with a fixed homogeneous transformation (to remove rotation axis jitter) such that horizontal and vertical distances to markers on the crane robot can be computed through homogeneous transformations as in [81]. Therefore, a marker placed on the camera payload's center of mass directly measures its horizontal and

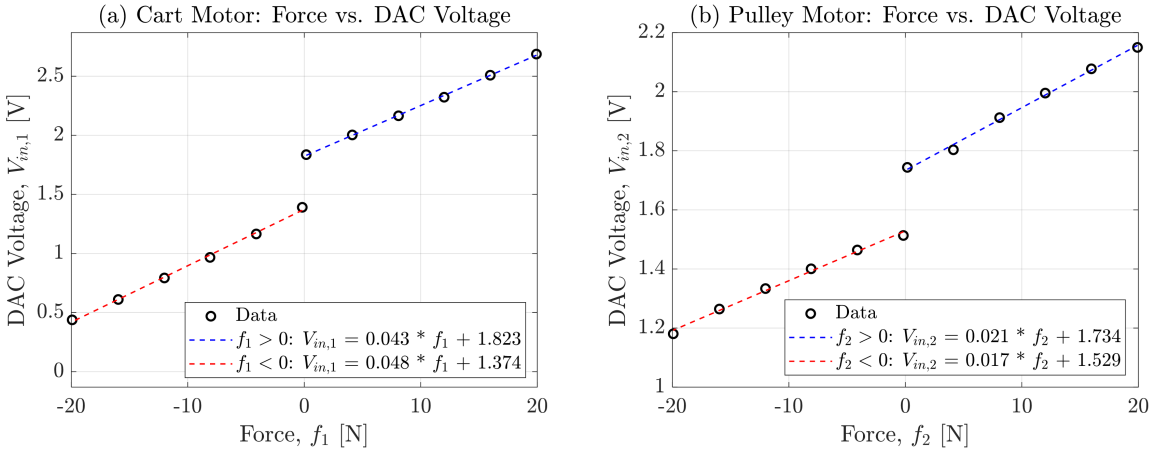


Figure 3.7: Plots relating crane robot motor forces to digital to analog converter (DAC) voltages. A desired force input for the crane robot is specified by inputting the corresponding DAC voltage according to the relationships for (a) the cart motor and (b) the pulley motor.

vertical positions of y_1 and y_2, respectively. From the measured horizontal distance to the cart marker, d_1, and a fixed measurement from the cart marker to the pulley, d_2, the cart position, x, is measured as

$$x = d_1 + d_2. \tag{3.5}$$

With the cart position, x, the swing angle, θ, can then be measured as

$$\theta = \arctan\left(\frac{y_1 - x}{y_2}\right). \tag{3.6}$$

Given the swing angle, θ, the length of the camera payload is

$$l = -\frac{y_2}{\cos(\theta)}. \tag{3.7}$$

Low frequency measurements are collected on a computer located near the operator and are wirelessly communicated to the crane robot with commands sent by the operator. Measurement noise was quantified as standard deviations when the crane robot was at rest, with the camera payload suspended at the center of the confined space as $\sigma_x = \sigma_{y_1} = 0.0007$ m, $\sigma_l = \sigma_{y_2} = 0.0059$ m, and $\sigma_\theta = 0.0026$ rad.

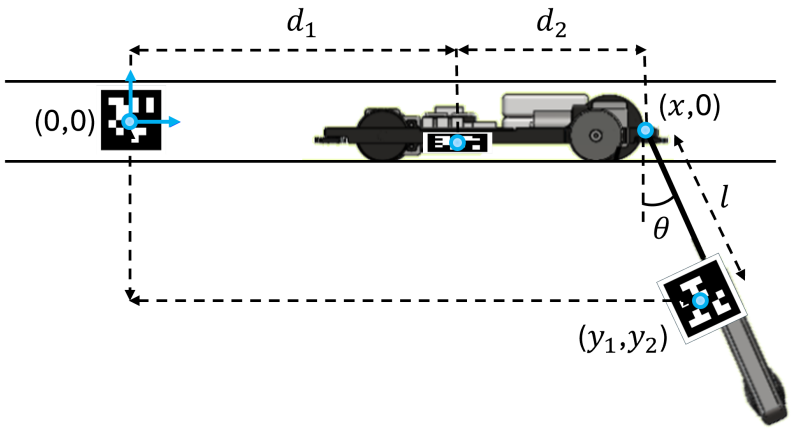


Figure 3.8: Fiducial marker configuration for global feedback. Crane robot states and outputs are computed using a combination of marker coordinate measurements and crane-robot kinematics.

Chapter 4

MC1: SEMI-AUTONOMOUS TELEOPERATION USING DIFFERENTIAL FLATNESS OF A CRANE ROBOT

This chapter presents MC1, which covers the development of semi-autonomous teleoperation using differential flatness of the crane robot. The focus is on managing swing dynamics of the crane robot for teleoperation in confined spaces of aircraft wings to perform efficient and collision-free remote inspection tasks. This chapter formulates the control problem, derives the crane robot's flatness-based control for semi-autonomous teleoperation, and experimentally validates the control through both autonomous and teleoperated crane robot experiments. This chapter is based on work in [16].

4.1 Problem formulation

The dynamics of the crane robot resembles those of a variable-length gantry crane [46], as depicted in Fig. 4.1. Under low-speed operations, it is feasible to neglect the payload swing, θ, dynamics. Without the swing dynamics, the horizontal position, y_1, of the payload (inspection camera) corresponds to the cart position, x, and the vertical camera position, y_2, corresponds to the payload length, l. As a result, the outputs (the horizontal y_1 and vertical y_2 positioning of the camera) are decoupled from each other and can be controlled independent of each other.

The decoupled approach, without compensating for the swing dynamics (i.e. assuming $x = y_1$ and $l = -y_2$), can lead to acceptable tracking during low-speed operation. To illustrate, tracking is studied for a ramp-like trajectory to move across the confined space over a pipeline obstacle, as shown in Fig. 4.2. The nominal desired trajectory, $\tilde{Y} = \begin{bmatrix} \tilde{y}_1(\tau) & \tilde{y}_2(\tau) \end{bmatrix}^T$, is defined by

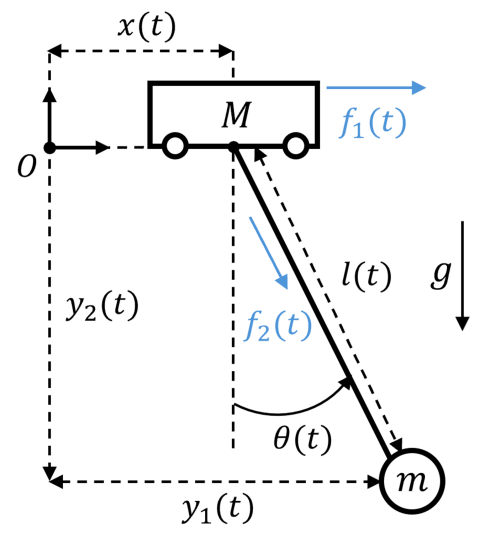


Figure 4.1: The crane-robot schematic. The system inputs are the force on the cart, f_1, and the force on the payload, f_2, and the system outputs are the horizontal and vertical positions of the payload, y_1 and y_2, respectively. The states of the system are the cart's position, x, the payload length, l, and the payload swing angle, θ, along with their time derivatives.

$$\tilde{Y} = \begin{cases} \begin{bmatrix} 0 & 0 \end{bmatrix}^T & 0 \le \tau < T_t \\ \begin{bmatrix} \frac{\Delta y_1}{T_t}(\tau - T_t) & \frac{2\Delta y_2}{T_t}(\tau - T_t) \end{bmatrix}^T & T_t \le \tau < \frac{3}{2}T_t \\ \begin{bmatrix} \frac{\Delta y_1}{T_t}(\tau - T_t) & -\frac{2\Delta y_2}{T_t}(\tau - 2T_t) \end{bmatrix}^T & \frac{3}{2}T_t \le \tau < 2T_t \\ \begin{bmatrix} \Delta y_1 & 0 \end{bmatrix}^T & 2T_t \le \tau < 4T_t \end{cases},$$

where the transition time, T_t, defines the duration of movement, Δy_1 defines the change in the camera's horizontal position over T_t, and Δy_2 defines the change in the camera's vertical position over $\frac{T_t}{2}$ before returning to the initial position within $\frac{T_t}{2}$. The nominal time trajectories $\tilde{y}_1$ and $\tilde{y}_2$ are smoothed by four cascaded first-order low-pass filters with a cutoff frequency, α, as $y_1(\tau) = F_f \tilde{y}_1$ and $y_2(\tau) = F_f \tilde{y}_2$, with

$$F_f = \left(\frac{\alpha}{\lambda + \alpha} \right)^4,$$

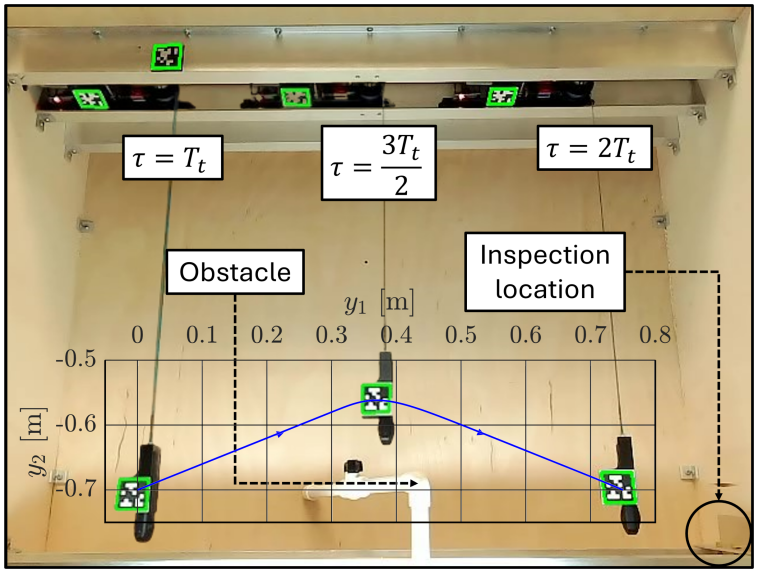


Figure 4.2: The experimental confined space with snapshots of the crane robot prototype traversing the ramp-like trajectory which transitions the camera payload over a pipeline obstacle near an inspection location.

where λ represents the Laplace variable.

The response of the crane robot with cart mass, $M = 0.815$ kg, and payload mass, $m = 0.225$ kg, when following the time trajectories $y_1(\tau)$ and $y_2(\tau)$, parameterized by $\Delta y_1 = 0.75$ m, $\Delta y_2 = 0.15$ m, and $\alpha = 10 \frac{\text{rad}}{\text{s}}$ (1.59 Hz) at a slow transition time of T_t of 40 seconds, by tracking the decoupled commands of the cart and payload length are shown in Fig. 4.3(a). Residual oscillations, present after reaching the inspection point, are small and remain below a magnitude of 1.4 degrees, as shown in Fig. 4.3(b).

However, this decoupling of the payload positioning is only valid at low speeds and accelerations, and rapid changes in the positions (e.g., needed for faster movements to speed up inspection) can excite the swing dynamics, inducing significant oscillations, which in turn, can make teleoperation challenging. The swing dynamics are excited as the same ramp-like trajectory is tracked with a smaller transition time, $T_t = 4$ seconds, with a 6.5 times increase in residual oscillation magnitude to 9.1 degrees, as shown in Fig. 4.3(a)(b). Such large residual oscillations need to be avoided to enable fast teleoperation. Therefore, the research

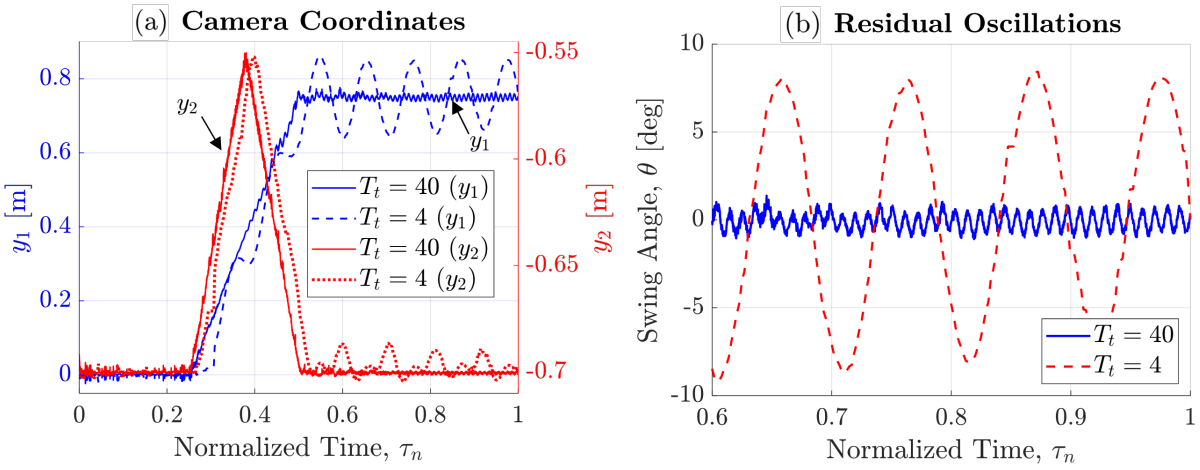


Figure 4.3: Comparison of residual oscillations for slow and fast trajectories, without swing-dynamics compensation. (a) Comparative time responses of tracking the decoupled commands for the horizontal camera coordinate, y_1, and the vertical camera coordinate, y_2, at two different transition times, $T_t = 4$ and $T_t = 40$ seconds, plotted against normalized time, $\tau_n = \frac{\tau}{4T_t}$. (b) The residual oscillations upon arriving at the inspection point for the two transition times.

problem is to compensate for the swing dynamics to reduce residual oscillations.

4.2 Crane Robot Dynamics

Equations of motion for the crane robot can be derived from its model in Fig 4.1, which can then be used to derive its controller. It is assumed that the rolling friction on the cart is negligible, or has been compensated. Generalized coordinates (i.e., system variables completely defining the state of motion) for the crane robot model can be selected as the cart position, x, camera swing angle, θ, and camera payload length, l. The kinetic energy, T, for the crane robot can be expressed the sum of cart and camera payload kinetic energy as

$$T = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m\left(\dot{y}_1^2 + \dot{y}_2^2\right), \tag{4.1}$$

where M is the mass of the cart, m is the mass of the payload, and $(\dot{y}_1^2 + \dot{y}_2^2)$ is the squared camera payload velocity. The squared camera payload velocity can be written in terms of

generalized coordinates as

$$\begin{aligned}(\dot{y}_1^2 + \dot{y}_2^2) &= \left(\dot{x} + \dot{l}s + l\dot{\theta}c\right)^2 + \left(-\dot{l}c + l\dot{\theta}s\right)^2 \\ &= \dot{x}^2 + 2\dot{x}\dot{l}s + 2\dot{x}l\dot{\theta}c + \dot{l}^2s^2 + 2l\dot{l}\dot{\theta}sc + l^2\dot{\theta}^2c^2 + \dot{l}^2c^2 - 2l\dot{l}\dot{\theta}sc + l^2\dot{\theta}^2s^2 \\ &= \dot{x}^2 + 2\dot{x}l\dot{\theta}c + 2\dot{x}\dot{l}s + l^2\dot{\theta}^2 + \dot{l}^2, \end{aligned} \tag{4.2}$$

where $s = \sin(\theta(t))$ and $c = \cos(\theta(t))$. Substituting Eq. (4.2) into Eq. (4.1) puts the kinetic energy in terms of generalized coordinates as

$$T = \frac{1}{2}(M+m)\dot{x}^2 + \frac{1}{2}m\dot{l}^2 + \frac{1}{2}ml^2\dot{\theta}^2 + m\dot{x}\dot{l}s + m\dot{x}l\dot{\theta}c. \tag{4.3}$$

The potential energy, U, for the crane robot can be expressed relative to the fixed reference of the origin at the stringer channel as

$$U = -mglc, \tag{4.4}$$

where g is the gravitational acceleration. The Laplacian, L, is defined as

$$L = T - U = \frac{1}{2}(M+m)\dot{x}^2 + \frac{1}{2}m\dot{l}^2 + \frac{1}{2}ml^2\dot{\theta}^2 + m\dot{x}\dot{l}s + m\dot{x}l\dot{\theta}c + mglc \tag{4.5}$$

The equations of motion for the crane robot can be derived from the Laplacian, L, through Lagrange's equations of

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = Q_j, \tag{4.6}$$

where q_j represents each generalized coordinate of (x, θ, l), and Q_j represents the generalized forces applied through the cart motor, f_1, and pulley motor, f_2. Substituting the generalized coordinates and generalized forces into Eq. (4.6) results in the following equations of motion

$$\begin{aligned} x:\ & (M+m)\ddot{x} + m\ddot{l}s + m\dot{l}\dot{\theta}c + m\dot{l}\dot{\theta}c + ml\ddot{\theta}c - ml\dot{\theta}^2s = f_1 \\ & (M+m)\ddot{x} + ml\ddot{\theta}c + m\ddot{l}s - ml\dot{\theta}^2s + 2m\dot{l}\dot{\theta}c = f_1 \end{aligned} \tag{4.7}$$

$$\begin{aligned} \theta:\ & 2m\dot{l}\dot{\theta} + ml^2\ddot{\theta} + m\ddot{x}lc + m\dot{x}\dot{l}c - m\dot{x}l\dot{\theta}s - m\dot{x}\dot{l}c + m\dot{x}l\dot{\theta}s + mgls = 0 \\ & m\ddot{x}lc + ml^2\ddot{\theta} + 2m\dot{l}\dot{\theta} + mgls = 0 \end{aligned} \tag{4.8}$$

$$\begin{aligned} l:\ & m\ddot{l} + m\ddot{x}s + m\dot{x}\dot{\theta}c - ml\dot{\theta}^2 - m\dot{x}\dot{\theta}c - mgc = f_2 \\ & m\ddot{x}s + m\ddot{l} - ml\dot{\theta}^2 - mgc = f_2 \end{aligned} \tag{4.9}$$

With equations of motion for the crane robot, a controller can be designed to plan camera payload trajectories.

4.3 Methods

4.3.1 Flatness-based feedforward inputs

The equations of motion for the crane robot, Eqs. (4.7)-(4.9), can be written in matrix form as

$$\begin{bmatrix} (M+m) & mlc & ms \\ mlc & ml^2 & 0 \\ ms & 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \\ \ddot{l} \end{bmatrix} = \begin{bmatrix} ml\dot{\theta}^2 s - 2m\dot{l}\dot{\theta}c + f_1 \\ -2ml\dot{l}\dot{\theta} - mgls \\ ml\dot{\theta}^2 + mgc + f_2 \end{bmatrix}, \tag{4.10}$$

Inverting the square matrix on the left hand side of Eq. (4.10), and rewriting in the state-space form, results in

$$\frac{d}{dt}X = \begin{bmatrix} \dot{x} \\ 0 \\ \dot{\theta} \\ -\frac{2}{l}\dot{l}\dot{\theta} - \frac{g}{l}s \\ \dot{l} \\ l\dot{\theta}^2 + gc \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{M} & \frac{-s}{M} \\ 0 & 0 \\ \frac{-c}{Ml} & \frac{sc}{Ml} \\ 0 & 0 \\ \frac{-s}{M} & \left(\frac{s^2}{M} + \frac{1}{m}\right) \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \tag{4.11}$$

$$= f(X) + g(X)F \tag{4.12}$$

with state vector $X = \begin{bmatrix} x & \dot{x} & \theta & \dot{\theta} & l & \dot{l} \end{bmatrix}^T$. The outputs of the system are the camera's horizontal position, y_1, and vertical position, y_2, which can be expressed in terms of the crane-robot states (x, l, θ) as

$$Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x + ls \\ -lc \end{bmatrix}. \tag{4.13}$$

To enable tracking of the outputs (y_1, y_2), an expression relating the input forces (f_1 and f_2) to the outputs is found by differentiating the outputs until the inputs appears [14].

Specifically, differentiating Eq. (4.13) twice results in

$$\dot{Y} = \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} + \dot{l}s + l\dot{\theta}c \\ -\dot{l}c + l\dot{\theta}s \end{bmatrix}, \tag{4.14}$$

$$\ddot{Y} = \begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \begin{bmatrix} \ddot{x} + \ddot{l}s + 2\dot{l}\dot{\theta}c + l\ddot{\theta}c - l\dot{\theta}^2 s \\ -\ddot{l}c + 2\dot{l}\dot{\theta}s + l\ddot{\theta}s + l\dot{\theta}^2 c \end{bmatrix}. \tag{4.15}$$

The second time derivative $\ddot{Y}$ depends on the inputs (f_1, f_2) since substituting for the second derivatives of the states from Eq. (4.11) into Eq. (4.15) results in

$$\begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -g \end{bmatrix} + \begin{bmatrix} 0 & \frac{s}{m} \\ 0 & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -g \end{bmatrix} + \tilde{\beta} F, \tag{4.16}$$

where $\tilde{\beta}$ is defined as the matrix of terms preceding the input vector $F = \begin{bmatrix} f_1 & f_2 \end{bmatrix}^T$. However, the force F cannot be found from Eq. (4.16) since the matrix $\tilde{\beta}$ is not invertible. Therefore, assuming that the input f_2 is sufficiently smooth, and redefining the new input to be $\dot{f}_2$ (with f_2 considered as an extended state), the output expression in Eq. (4.16) is differentiated again to obtain

$$Y^{(3)} = \begin{bmatrix} y_1^{(3)} \\ y_2^{(3)} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} f_2 \dot{\theta} c \\ f_2 \dot{\theta} s \end{bmatrix} + \tilde{\beta} \begin{bmatrix} f_1 \\ \dot{f}_2 \end{bmatrix}, \tag{4.17}$$

where the superscript in brackets (i) indicates the i^{th} time derivative, e.g., $y_k^{(i)}$ denotes i^{th} time derivative of y_k for $k \in \{1, 2\}$. Again, the redefined input $(f_1, \dot{f}_2)$ cannot be found from Eq. (4.17) since the matrix $\tilde{\beta}$ is not invertible. Therefore, the input is further redefined to be $\ddot{f}_2$, with $(f_2, \dot{f}_2)$ considered as extended states, and the output expression in Eq. (4.17) is differentiated again to obtain

$$Y^{(4)} = \begin{bmatrix} y_1^{(4)} \\ y_2^{(4)} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} \ddot{f}_2 s + 2\dot{f}_2 \dot{\theta} c + f_2 \ddot{\theta} c - f_2 \dot{\theta}^2 s \\ -\ddot{f}_2 c + 2\dot{f}_2 \dot{\theta} s + f_2 \ddot{\theta} s + f_2 \dot{\theta}^2 c \end{bmatrix}. \tag{4.18}$$

Substituting for the second derivatives of the states from Eq. (4.11) into Eq. (4.18), and arranging yields

$$\begin{bmatrix} y_1^{(4)} \\ y_2^{(4)} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} -\frac{1}{Ml} f_2 c^2 & s \\ -\frac{1}{Ml} f_2 cs & -c \end{bmatrix} \begin{bmatrix} f_1 \\ \ddot{f}_2 \end{bmatrix} - \frac{1}{m} \begin{bmatrix} -2\dot{f}_2\dot{\theta}c + f_2\dot{\theta}^2 s - \frac{1}{Ml} f_2^2 sc^2 + \frac{2}{l} f_2 \dot{l}\dot{\theta}c + \frac{g}{l} f_2 sc \\ -2\dot{f}_2\dot{\theta}s - f_2\dot{\theta}^2 c - \frac{1}{Ml} f_2^2 s^2 c + \frac{2}{l} f_2 \dot{l}\dot{\theta}s + \frac{g}{l} f_2 s^2 \end{bmatrix}. \tag{4.19}$$

The final redefined input $(f_1, \ddot{f}_2)$ can be found from Eq. (4.19) if the matrix preceding the input vector $\begin{bmatrix} f_1 & \ddot{f}_2 \end{bmatrix}^T$ of

$$\beta = \begin{bmatrix} -\frac{1}{Ml} f_2 c^2 & s \\ -\frac{1}{Ml} f_2 cs & -c \end{bmatrix} \tag{4.20}$$

is invertible. The determinant of β is $\frac{f_2 c}{Ml}$, making it invertible provided the payload length is nonzero, $l \neq 0$, the payload swing angle does not become horizontal, $\theta \neq \{-\frac{\pi}{2}, \frac{\pi}{2}\}$, and the payload force is nonzero, $f_2 \neq 0$. These conditions are met by the crane robot since the payload length is positive ($l > 0$) when deployed during inspection, the payload swing angle, θ, must remain within $(-\frac{\pi}{2}, \frac{\pi}{2})$ to avoid colliding with the ceiling of the confined space, and the payload force, f_2, which operates about it's equilibrium at $f_2 = -mg$, is required to be negative ($f_2 < 0$) to ensure that the cable remains taut. Given the desired outputs' fourth derivatives, $Y_d^{(4)}$, the redefined input f_1 and $\ddot{f}_2$, can be found by settling the right hand side of Eq. (4.19) to be $\begin{bmatrix} v_1 & v_2 \end{bmatrix}^T$ as

$$\begin{aligned} f_1 = \frac{Ml}{f_2 c}(&-m(cv_1 + sv_2) + 2\dot{f}_2\dot{\theta} \\ &+ \frac{1}{Ml} f_2^2 sc - \frac{2}{l} f_2 \dot{l}\dot{\theta} - \frac{g}{l} f_2 s), \end{aligned} \tag{4.21}$$

$$\ddot{f}_2 = m(sv_1 - cv_2) + f_2\dot{\theta}^2 \tag{4.22}$$

leading to the system

$$y_1^{(4)} = v_1 \tag{4.23}$$

$$y_2^{(4)} = v_2. \tag{4.24}$$

Here, the feedforward inputs $f_{1,ff}$ and $\ddot{f}_{2,ff}$ can be found from Eq. (4.21) and Eq. (4.22), respectively, by setting $v_1 = y_{1,d}^{(4)}$ and $v_2 = y_{2,d}^{(4)}$

$$f_{1,ff} = \frac{Ml}{f_2 c}(-m(cy_{1,d}^{(4)} + sy_{2,d}^{(4)}) + 2\dot{f}_2\dot{\theta} + \frac{1}{Ml}f_2^2 sc - \frac{2}{l}f_2 l\dot{\theta} - \frac{g}{l}f_2 s), \tag{4.25}$$

$$\ddot{f}_{2,ff} = m(sy_{1,d}^{(4)} - cy_{2,d}^{(4)}) + f_2\dot{\theta}^2 \tag{4.26}$$

and then integrating twice over time to find the feedforward input $f_{2,ff}$. A specified trajectory-tracking performance, i.e., a desired characteristic equation for the error dynamics, say

$$\lambda^4 + \sum_{i=0}^{3} a_{k,i}\lambda^i = 0$$

where the error is

$$e_k = y_k - y_{k,d}, \qquad k \in \{1, 2\}$$

can be achieved by selecting the controller v_k, $k \in \{1, 2\}$ in Eqs. (4.23) and (4.24) as

$$v_k = y_{k,d}^{(4)} - \sum_{i=0}^{3} a_{1,i}(y_k^{(i)} - y_{k,d}^{(i)}), \quad k \in \{1, 2\} \tag{4.27}$$

resulting in system inputs

$$f_1 = \frac{Ml}{f_2 c}\Big(- m\Big(c\big(y_{1,d}^{(4)} - \sum_{i=0}^{3} a_{1,i}(y_1^{(i)} - y_{1,d}^{(i)})\big) + s\big(y_{2,d}^{(4)} - \sum_{i=0}^{3} a_{2,i}(y_2^{(i)} - y_{2,d}^{(i)})\big)\Big) + 2\dot{f}_2\dot{\theta} + \frac{1}{Ml}f_2^2 sc - \frac{2}{l}f_2 l\dot{\theta} - \frac{g}{l}f_2 s\Big), \tag{4.28}$$

$$f_2 = \iint \Big(m\Big(s\big(y_{1,d}^{(4)} - \sum_{i=0}^{3} a_{1,i}(y_1^{(i)} - y_{1,d}^{(i)})\big) - c\big(y_{2,d}^{(4)} - \sum_{i=0}^{3} a_{2,i}(y_2^{(i)} - y_{2,d}^{(i)})\big)\Big) + f_2\dot{\theta}^2\Big)\, d^2t + \dot{f}_2(0)t + f_2(0), \tag{4.29}$$

where $f_2(0)$ and $\dot{f}_2(0)$ are initial conditions.

Remark 1. *Starting from rest, the initial conditions for Eq.* (4.29) *are* $f_2(0) = -mg$ *and* $\dot{f}_2(0) = 0$.

4.3.2 Stabilizing feedback

To avoid taking time derivatives of potentially noisy output measurements, the following provides a state feedback $F_{fb} = \begin{bmatrix} f_{1,fb} & f_{2,fb} \end{bmatrix}^T$, that achieves stable trajectory tracking without these high order, low frequency time derivatives and without full state feedback (i.e., without swing angular velocity $\dot{\theta}$), provided the tracked trajectories are sufficiently slow, resulting in a small swing angle.

The feedforward input F_{ff} is augmented with feedback F_{fb}

$$F = \begin{bmatrix} f_{1,ff} \\ f_{2,ff} \end{bmatrix} + \begin{bmatrix} f_{1,fb} \\ f_{2,fb} \end{bmatrix} \quad = F_{ff} + F_{fb} \tag{4.30}$$

to stabilize the system.

Lemma 1. *The feedback law*

$$F_{fb} = -KX = - \begin{bmatrix} k_1 & k_2 & k_3 & k_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_5 & k_6 \end{bmatrix} X, \tag{4.31}$$

stabilizes the system in Eq. (4.11) *about the equilibrium state,* $X_{eq} = \begin{bmatrix} x_0 & 0 & 0 & 0 & l_0 & 0 \end{bmatrix}^T$, *and corresponding equilibrium input,* $F_{eq} = \begin{bmatrix} 0 & -mg \end{bmatrix}^T$, *at any given cart position* x_0 *and positive payload length* l_0, *provided*

$$\begin{aligned}
&k_1 > 0, \\
&k_2 l_0 - k4 > 0, \\
&\frac{-k_4 Mg}{k_2 l_0 - k_4} + k_1 l_0 - k_3 + mg > 0, \\
&\frac{-k_2 k_4 Mg + (k_2 l_0 - k_4)(k_1 k_4 - k_2 k_3 + k_2 mg)}{-k_4 Mg + (k_2 l_0 - k_4)(k_1 l_0 - k_3 + mg)} > 0, \\
&k_5 > 0, \\
&k_6 > 0.
\end{aligned} \tag{4.32}$$

Proof. Linearization of the system model in Eq. (4.11) about the equilibrium in the lemma results in

$$\dot{X} = AX + BF, \tag{4.33}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{mg}{M} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{(M+m)g}{Ml_0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} ; \; B = \begin{bmatrix} 0 & 0 \\ \frac{1}{M} & 0 \\ 0 & 0 \\ -\frac{1}{Ml_0} & 0 \\ 0 & 0 \\ 0 & \frac{1}{m} \end{bmatrix}. \tag{4.34}$$

Applying the input as the feedback $F = F_{fb}$ in Eq. (4.31) to Eq. (4.33) results in the closed-loop dynamics

$$\begin{aligned} \dot{X} &= \left[\begin{array}{cccc|cc} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{k_1}{M} & -\frac{k_2}{M} & \frac{mg-k_3}{M} & -\frac{k_4}{M} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k_1}{Ml_0} & \frac{k_2}{Ml_0} & \frac{k_3-(M+m)g}{Ml_0} & \frac{k_4}{Ml_0} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -\frac{k_5}{m} & -\frac{k_6}{m} \end{array}\right] X \\ &= \begin{bmatrix} A_1 & 0_{4x2} \\ 0_{2x4} & A_2 \end{bmatrix} X. \end{aligned} \tag{4.35}$$

Conditions on the gains in K for stability can be derived from the blocks, A_1 and A_2, separately. The characteristic equation of A_1 is given by

$$\begin{aligned} &Ml_0\lambda^4 + (k_2l_0 - k_4)\lambda^3 + (k_1l_0 - k_3 + (M+m)g)\lambda^2 \\ &+ k_2g\lambda + k_1g = 0, \end{aligned} \tag{4.36}$$

which can be used to construct the Routh array

$$\begin{array}{c|ccc} \lambda^4 & Ml_0 & k_1l_0 - k_3 + (M+m)g & k_1g \\ \lambda^3 & k_2l_0 - k_4 & k_2g & 0 \\ \lambda^2 & c_1 & k_1g & 0 \\ \lambda^1 & c_2 & 0 & 0 \\ \lambda^0 & k_1g & 0 & 0 \end{array} \tag{4.37}$$

where

$$\begin{aligned} c_1 &= -\frac{\begin{vmatrix} Ml_0 & k_1l_0 - k_3 + (M+m)g \\ k_2l_0 - k_4 & k_2g \end{vmatrix}}{k_2l_0 - k_4} \\ &= \frac{-k_2Mgl_0 + (k_2l_0 - k_4)(k_1l_0 - k_3 + (M+m)g)}{k_2l_0 - k_4} \\ &= \frac{-k_2Mgl_0 + k_2Mgl_0 - k_4Mg + (k_2l_0 - k_4)(k_1l_0 - k_3 + mg)}{k_2l_0 - k4} \\ &= \frac{-k_4Mg}{k_2l_0 - k_4} + k_1l_0 - k_3 + mg, \end{aligned} \tag{4.38}$$

$$\begin{aligned} c_2 &= -\frac{\begin{vmatrix} k_2l_0 - k_4 & k_2g \\ \frac{-k_4Mg}{k_2l_0 - k_4} + k_1l_0 - k_3 + mg & k_1g \end{vmatrix}}{\frac{-k_4Mg}{k_2l_0 - k_4} + k_1l_0 - k_3 + mg} \\ &= \frac{-k_1k_2l_0 + k_1k_4 - \frac{k_2k_4Mg}{k_2l_0 - k_4} + k_1k_2l_0 - k_2k_3 + k_2mg}{\frac{-k_4Mg}{k_2l_0 - k_4} + k_1l_0 - k_3 + mg} g \\ &= \frac{k_1k_4 - \frac{k_2k_4Mg}{k_2l_0 - k_4} - k_2k_3 + k_2mg}{\frac{-k_4Mg}{k_2l_0 - k_4} + k_1l_0 - k_3 + mg} g \\ &= \frac{-k_2k_4Mg + (k_2l_0 - k_4)(k_1k_4 - k_2k_3 + k_2mg)}{-k_4Mg + (k_2l_0 - k_4)(k_1l_0 - k_3 + mg)} g. \end{aligned} \tag{4.39}$$

Routh-Hurwitz criteria ensures stability if all terms in the first column of the Routh array have no sign changes, or are positive since the first term Ml_0 is positive [82]. The character-

istic equation for A_2 is given by

$$m\lambda^2 + k_6\lambda + k_5 = 0, \tag{4.40}$$

which results in stability provided $k_5 > 0$ and $k_6 > 0$. The lemma follows. □

Corollary 1. *The system in Eq. (4.35) can be stabilized without feedback from the swing dynamics (i.e., with $k_3 = k_4 = 0$) by selecting positive gains $k_1 > 0$, $k_2 > 0$, $k_5 > 0$, and $k_6 > 0$.*

Proof. By considering $k_3 = k_4 = 0$, the conditions for the stability of A_1 in Eq. (4.32) simplify to

$$\begin{cases} k_1 > 0, \\ k_2 l_0 > 0, \\ k_1 l_0 + mg > 0, \\ \frac{k_2}{k_1 l_0 + mg} > 0, \end{cases} \tag{4.41}$$

which are satisfied for $k_1 > 0$, and $k_2 > 0$. □

Remark 2. *With $k_3 = k_4 = 0$, stabilizing feedback is achievable without payload angle measurements, which can ease implementation by only requiring sensors to measure the cart position and pendulum length, e.g., using encoders placed on the motors.*

Corollary 2. *The system in Eq. (4.35) is stable by selecting gains as $k_1 > 0$, $k_2 > 0$, $k_5 > 0$, $k_6 > 0$ and with the swing dynamics gains as $k_3 < mg$, $k_4 = 0$.*

Proof. By considering $k_4 = 0$, the conditions for the stability of A_1 in Eq. (4.32) simplify to

$$\begin{cases} k_1 > 0, \\ k_2 l_0 > 0, \\ k_1 l_0 + (mg - k_3) > 0, \\ \frac{k_2(mg - k_3)}{k_1 l_0 + (mg - k_3)} > 0, \end{cases} \tag{4.42}$$

which are all satisfied for $k_1 > 0$, $k_2 > 0$, and $k_3 < mg$. □

Remark 3. *While not required for stability, the addition of swing-angle feedback (i.e., $k_3 < mg$ in Corollary 2) can damp undesired oscillations faster than the case without swing-angle feedback (i.e., $k_3 = 0$). The swing-angle feedback requires the use of an external camera feedback to measure the swing angle θ.*

The desired state and feedforward satisfies the system dynamics,

$$\dot{X}_d = f(X_d) + g(X_d)F_{ff}$$

in Eq. (4.11), i.e.,

$$\frac{d}{dt}X_d = \begin{bmatrix} \dot{x_d} \\ 0 \\ \dot{\theta}_d \\ -\frac{2}{l_d}\dot{l}_d\dot{\theta}_d - \frac{g}{l_d}s_d \\ \dot{l}_d \\ l_d\dot{\theta}_d^2 + gc_d \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{M} & \frac{-s_d}{M} \\ 0 & 0 \\ \frac{-c_d}{Ml_d} & \frac{s_dc_d}{Ml_d} \\ 0 & 0 \\ \frac{-s_d}{M} & \left(\frac{s_d^2}{M} + \frac{1}{m}\right) \end{bmatrix} \begin{bmatrix} f_{1,ff} \\ f_{2,ff} \end{bmatrix}, \tag{4.43}$$

where $s_d = sin(\theta_d)$ and $c_d = cos(\theta_d)$ and the corresponding desired system states X_d can be computed in terms of the desired outputs algebraically [44]; specifically,

$$x_d = y_{1,d} - \frac{\ddot{y}_{1,d}y_{2,d}}{\ddot{y}_{2,d} + g}, \tag{4.44}$$

$$\dot{x}_d = \dot{y}_{1,d} - \frac{y^{(3)}_{1,d}y_{2,d} + \ddot{y}_{1,d}\dot{y}_{2,d}}{\ddot{y}_{2,d} + g} - \frac{\ddot{y}_{1,d}y_{2,d}y^{(3)}_{2,d}}{(\ddot{y}_{2,d} + g)^2}, \tag{4.45}$$

$$l_d = \left(\left(\frac{\ddot{y}_{1,d}y_{2,d}}{\ddot{y}_{2,d} + g}\right)^2 + y_{2,d}^2\right)^{\frac{1}{2}}, \tag{4.46}$$

$$\dot{l}_d = \left(\left(\frac{\ddot{y}_{1,d}y_{2,d}}{\ddot{y}_{2,d} + g}\right)^2 + y_{2,d}^2\right)^{-\frac{1}{2}} \left(y_{2,d}\dot{y}_{2,d} + \left(\frac{\ddot{y}_{1,d}y_{2,d}}{\ddot{y}_{2,d} + g}\right) \times \left(\frac{y^{(3)}_{1,d}y_{2,d} + \ddot{y}_{1,d}\dot{y}_{2,d}}{\ddot{y}_{2,d} + g} - \frac{\ddot{y}_{1,d}y_{2,d}y^{(3)}_{2,d}}{(\ddot{y}_{2,d} + g)^2}\right)\right), \tag{4.47}$$

$$\theta_d = \tan^{-1}\left(\frac{-\ddot{y}_1}{\ddot{y}_2 + g}\right). \tag{4.48}$$

$$\dot{\theta}_d = \frac{-y^{(3)}_{1,d}\left(\ddot{y}_{2,d} + g\right) + \ddot{y}_{1,d}y^{(3)}_{2,d}}{(\ddot{y}_{2,d} + g)^2 + \ddot{y}_{1,d}^2} \tag{4.49}$$

Eqs. (4.44) - (4.49) impose the condition of $\ddot{y}_{2,d} > -g$ to maintain positive cable tension and prevent slackening.

Remark 4. *The crane robot will follow a four times differentiable desired output trajectory Y_d by applying the feedforward input F_{ff} in Eq.* (4.43)*. Feedback is added to stabilize the desired trajectory in response to perturbations.*

Lemma 2. *The swing dynamics (θ_d and $\dot{\theta}_d$), length variation velocity ($\dot{l}_d$), and change of feedforward forces from equilibrium values ($F_{ff} - F_{eq}$) can be made arbitrarily small for sufficiently-slowly-varying desired output trajectories Y_d (i.e., for sufficiently-small time derivatives $Y_d^{(i)}$ for $i = \{1, \ldots, 4\}$) such that*

$$\lim_{\max_t \|Y_d^{(i)}(t)\| \to 0} \left(\max_t |\theta_d(t)| \right) = 0 \tag{4.50}$$

$$\lim_{\max_t \|Y_d^{(i)}(t)\| \to 0} \left(\max_t |\dot{\theta}_d(t)| \right) = 0 \tag{4.51}$$

$$\lim_{\max_t \|Y_d^{(i)}(t)\| \to 0} \left(\max_t |\dot{l}_d(t)| \right) = 0 \tag{4.52}$$

$$\lim_{\max_t \|Y_d^{(i)}(t)\| \to 0} \left(\max_t \|F_{ff}(t) - F_{eq}\| \right) = 0. \tag{4.53}$$

Proof. Assume the crane robot is commanded a desired output trajectory Y_d from equilibrium (i.e., $F = F_{eq}$). By Eqs. (4.47)-(4.49), $\dot{l}_d \to 0$, $\theta_d \to 0$, and $\dot{\theta}_d \to 0$ as $Y_d^{(i)} \to 0$. $\theta_d \to 0 \implies c_d \to 1$ and $s_d \to 0$. Therefore, by Eq. (4.25), $f_{1,ff} \to f_{1,eq} \to 0$ and by Eq. (4.26), $\ddot{f}_{2,ff} \to 0 \implies f_{2,ff} \to f_{2,eq} = -mg$. The lemma follows. □

Lemma 3. *The origin of the error $E = X - X_d$ dynamics with the feedforward input F_{ff} augmented with feedback F_{fb}*

$$F = \begin{bmatrix} f_{1,ff} \\ f_{2,ff} \end{bmatrix} + \begin{bmatrix} f_{1,fb} \\ f_{2,fb} \end{bmatrix} = F_{ff} + F_{fb} \tag{4.54}$$

is stable provided the desired output trajectories Y_d are sufficiently slowly-varying, i.e., the time derivatives $Y_d^{(i)}$ for $i = \{1, \ldots, 4\}$ are sufficiently small, with sufficiently small deviations in payload length, i.e., $\max_t(|l_d(t) - l_0|)$ is sufficiently small.

Proof. The error dynamics are given by

$$\dot{E} = \dot{X} - \dot{X}_d = f(X) + g(X)F - f(X_d) - g(X_d)F_{ff}. \tag{4.55}$$

By Taylor Series expansion,

$$f(X) \approx f(X_d) + \left(\frac{\partial f(X)}{\partial X}|_{X=X_d}\right)(X - X_d), \tag{4.56}$$

$$\begin{aligned} g(X)F \approx g(X_d)F_{ff} &+ \left(\frac{\partial g(X)F}{\partial X}|_{F=F_{ff},X=X_d}\right)(X - X_d) \\ &+ \left(\frac{\partial g(X)F}{\partial F}|_{F=F_{ff},X=X_d}\right)(F - F_{ff}). \end{aligned} \tag{4.57}$$

Therefore, the error dynamics becomes

$$\begin{aligned} \dot{E} &\approx f(X_d) + \left(\frac{\partial f(X)}{\partial X}|_{X=X_d(t)}\right)(X - X_d) \\ &\quad + g(X_d)F_{ff} + \left(\frac{\partial g(X)F}{\partial X}|_{F=F_{ff},X=X_d}\right)(X - X_d) \\ &\quad + \left(\frac{\partial g(X)F}{\partial F}|_{X=X_d(t),F=F_{ff}}\right)(F - F_{ff}) \\ &\quad - f(X_d) - g(X_d)F_{ff} \\ &= f(X_d) - f(X_d) + A(t)E + B(t)F_{fb} \\ &\quad + g(X_d)F_{ff} - g(X_d)F_{ff} \\ &= A(t)E + B(t)F_{fb} \\ &= A(t)E - B(t)KE \\ &= A(t)E - B(t)KE + AE - AE + BKE - BKE \\ &= (A - BK)E + (A(t) - A)E + (B - B(t))KE \\ &= (A - BK)E + P(t) \end{aligned} \tag{4.58}$$

where

$$A(t) = \left(\frac{\partial f(X)}{\partial X}|_{X=X_d}\right) + \left(\frac{\partial g(X)F}{\partial X}|_{X=X_d,F=F_{ff}}\right), \tag{4.59}$$

$$B(t) = \left(\frac{\partial g(X)F}{\partial F}|_{X=X_d,F=F_{ff}}\right) = g(X_d), \tag{4.60}$$

and $P(t)$ is a time varying perturbation to the exponentially stable linearized dynamics in Eq. (4.33) with terms

$$P(t) = (A(t) - A)E + (B - B(t))KE, \tag{4.61}$$

$$(A(t) - A) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{(mg+f_{2,ff})}{M} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \tilde{c}_1 & -\frac{2}{l_d}\dot{l}_d & \tilde{c}_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \tilde{c}_3 & 2l_d\dot{\theta}_d & \dot{\theta}_d^2 & 0 \end{bmatrix} E, \tag{4.62}$$

$$\tilde{c}_1 = \frac{s_d(f_{1,ff} - s_d f_{2,ff})}{Ml_d} + \frac{c_d^2 f_{2,ff} l_0 + mgl_d}{Ml_0 l_d} + \frac{g(l_d - l_0 c_d)}{l_0 l_d},$$

$$\tilde{c}_2 = \frac{c_d(f_{1,ff} - s_d f_{2,ff})}{Ml_d^2} + \frac{gs_d + 2\dot{l}_d\dot{\theta}_d}{l_d^2},$$

$$\tilde{c}_3 = -\frac{c_d f_{1,ff}}{M} + \frac{2s_d c_d f_{2,ff}}{M} - gs_d,$$

$$(B - B(t))KE = \begin{bmatrix} 0 & 0 \\ 0 & \frac{s_d}{M} \\ 0 & 0 \\ \frac{l_0 c_d - l_d}{Ml_d l_0} & -\frac{s_d c_d}{Ml_d} \\ 0 & 0 \\ \frac{s_d}{M} & -\frac{s_d^2}{M} \end{bmatrix} KE. \tag{4.63}$$

The time varying perturbation $P(t)$ in Eq. (4.61) can be made arbitrarily small for sufficiently-slowly varying desired trajectories (i.e., $\dot{l}_d$, $\theta_d \to 0$, $\dot{\theta}_d \to 0$, $f_{1,ff} \to 0$, and $f_{2,ff} \to -mg$ by Lemma 2) with sufficiently small deviations in payload length $(l_d(t) - l_0)$. As a result, the perturbation $P(t)$ is of the vanishing type as the error $E \to 0$ and satisfies the bound $\|P(t)\| \leq \gamma\|X\|$, where γ can be made arbitrarily small for sufficiently-slowly changing desired trajectories. Therefore, by Lemma 9.1 in [83], the time-varying trajectory $X = X_d$ is exponentially stable. □

Output tracking can be achieved by combining the feedforward (from Eq. (4.25) and Eq. (4.26)) and a stabilizing feedback (from Eq. (4.31) with $k_4 = 0$) as

$$\begin{aligned} f_1 = & \frac{Ml_d}{f_2 c_d}(-m(c_d y_{1,d}^{(4)} + s_d y_{2,d}^{(4)}) \\ & + 2\dot{f}_2\dot{\theta}_d + \frac{1}{Ml_d} f_2^2 s_d c_d - \frac{2}{l_d} f_2 \dot{l_d}\dot{\theta}_d - \frac{g}{l_d} f_2 s_d) \\ & - k_1(x - x_d) - k_2(\dot{x} - \dot{x}_d) - k_3(\theta - \theta_d), \end{aligned} \tag{4.64}$$

$$\begin{aligned} f_2 = & \iint \left(m\left(s_d y_{1,d}^{(4)} - c_d y_{2,d}^{(4)} \right) + f_2 \dot{\theta}_d^2 \right) d^2 t \\ & + \dot{f}_2(0)t + f_2(0) - k_5(l - l_d) - k_6(\dot{l} - \dot{l}_d). \end{aligned} \tag{4.65}$$

4.3.3 Semi-autonomous controller

In semi-autonomous control the operator's reference command (output velocity, $\dot{Y}_r$), from the joystick interface, is used to autonomously plan a snap-continuous (i.e., C^4 continuous) trajectory, $Y_d(t)$, designed (i) to avoid collisions, and (ii) to be sufficiently smooth for output tracking using the differential flatness property.

Reference specification

The reference output-velocity, $\dot{Y}_r$, is specified by the operator as

$$\dot{Y}_r \;=\; \begin{bmatrix} \dot{y}_{1,r} \\ \dot{y}_{2,r} \end{bmatrix} = \begin{bmatrix} \alpha_1 j_1 \\ \alpha_2 j_2 \end{bmatrix}, \tag{4.66}$$

where α_1 and α_2 are gains scaling the joystick inputs, j_1 and j_2, respectively. The velocity reference, $\dot{Y}_r(t)$ at time t is used to define the nominal reference point, $\tilde{Y}_r(t+T)$, to be reached within the time horizon, T, from the current time, t, i.e., $\tilde{Y}_r(t+T) = Y(t) + \dot{Y}_r T$. The nominal reference point $\tilde{Y}_r(t+T)$ acts similar to velocity commands, as larger operator-inputs (j_1, j_2) will generate higher-speed trajectories.

Collision avoidance

Collision is evaluated through projection from the crane robot [84]. Intersection between obstacles and the projected path of the crane robot, l_r, connecting the output, $Y(t)$, to the nominal operator-specified reference point, $\tilde{Y}_r(t+T)$ indicate imminent collision. If the projected path falls inside an obstacle, then a corrected reference point, $Y_r(t+T)$, is selected to be outside the obstacle. Obstacle locations in the manufacturing environment are assumed to be known, allowing obstacles to be modeled by their axis-aligned bounding box. These bounding boxes extend to the floor of the wing bay as the crane suspension prevents the camera payload from moving below obstacles, illustrated in Fig. 4.4.

An intersection between the projection and an obstacle indicates imminent collision, as the projection predicts that the planned trajectories will pass through the bounding box. The set $L_w = \{l_{w,0}, l_{w,1}, \ldots, l_{w,N}\}$ of N line segments consisting of the confined space's wall line segments and the obstacles' bounding boxes, is checked for collision with the projected reference line segment, l_r. The set of intersection points, $P = \{Y_p \mid Y_p = l_r \cap l_{w,n},\ n \in \{0, 1, \ldots, N\}\}$, is used to obtain the corrected reference point, Y_r, located at a distance offset ϵ (to account for disturbance) along the projection, l_r, towards the current state, Y, from the closest intersection point, Y_p^*, if there is an intersection (see Fig. 4.4), such that

$$Y_r = \begin{cases} Y_p^* - \epsilon \frac{Y_p^* - Y(t)}{|Y_p^* - Y(t)|}, & \text{if } P \neq \varnothing \text{ (intersection)}, \\ \tilde{Y}_r, & \text{if } P = \varnothing \text{ (otherwise)}. \end{cases} \tag{4.67}$$

where

$$Y_p^* = \underset{Y_p \in P}{\arg\min} \|Y_p - Y(t)\|. \tag{4.68}$$

Trajectory generation

From the corrected reference point, Y_r, a snap-continuous, desired output-trajectory $Y_d(\cdot)$ is planned over the time interval $t \leq \tau \leq t + T$. Five initial boundary conditions at time $\tau = t$ are found from the outputs and time derivatives of the outputs, $Y_d^{(i)}(t)$, computed from

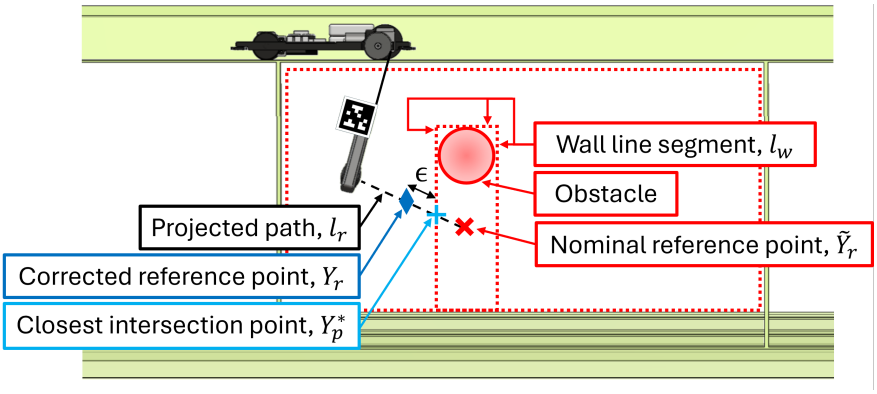


Figure 4.4: Illustration of collision prevention algorithm when the nominal reference point (cross), $\tilde{Y}_r$, specified by the operator lies within an obstacle's bounding box. The corrected reference point (diamond), Y_r, is specified at a distance ϵ towards the current position from the closest intersection point (plus), Y_p^*, of the reference line segment (dashed), l_r, and the set of bounding box walls (dotted), L_w.

(4.13),(4.14),(4.16),(4.17),(4.19). Similarly, five final boundary conditions at time $\tau = t + T$ are defined by the corrected reference point as $Y(t + T) = Y_r$, with final output derivatives set to zero (i.e. $\dot{Y}_d(t + T) = \ddot{Y}_d(t + T) = Y_d^{(3)}(t + T) = Y_d^{(4)}(t + T) = 0$) such that all desired trajectories are planned to reach a resting output state in the case of imminent collision [84]. Moreover, selecting the final time trajectory derivatives to zero (especially, $\ddot{Y}_d(t+T) = 0, Y_d^{(3)}(t+T) = 0$) results in zero final swing angle $\theta_d(t+T) = 0$ and zero swing-angle velocity $\dot{\theta}_d(t + T) = 0$, from Eqs. (4.48) and (4.49), and thereby, removes residual oscillations at time $t+T$. The minimal order polynomial is ninth-order with ten coefficients, i.e., $c_{k,0}$-$c_{k,9}$, to satisfy the ten boundary conditions on each output trajectory $Y_{k,d}^{(i)}(t)$ and $Y_{k,d}^{(i)}(t + T)$ for $0 \leq i \leq 4$, $k \in \{1, 2\}$. Therefore, the desired trajectory for each output, $y_{k,d}(\tau)$ ($k \in \{1, 2\}$), is selected independently as

$$y_{k,d}(\tau) = c_{k,9}\tau^9 + c_{k,8}\tau^8 + \ldots + c_{k,1}\tau + c_{k,0}. \tag{4.69}$$

Given the desired output trajectory, Y_d, as in Eq. (4.69) and its time derivatives found from the polynomials in Eq. (4.69), the control inputs, f_1 and f_2, can be found from Eq. (4.64) and Eq. (4.65). Fig. 3.2 shows a block diagram of the crane robot's control, and Alg. 1

summarizes the semi-autonomous controller, which generates new polynomial trajectories in real time based on joystick inputs at each control timestep, similar to [50].

4.4 Experimental results and discussion

4.4.1 Flatness-based input validation

Compensating for the swing dynamics substantially improves tracking of the desired trajectory when compared to the uncompensated case. To illustrate, the proposed flatness-based input in Eq. (4.64) and Eq. (4.65) is applied to the fast ramp-like trajectory presented in Section 4.1 with a transition time of $T_t = 4$ seconds to compensate for the undesired residual oscillations. Fig. 4.5(a) compares the time trajectories of the uncompensated response (neglecting the swing dynamics) to the compensated response (accounting for swing dynamics) for the horizontal y_1 and vertical y_2 camera positions. The maximum amplitude of the residual oscillations upon reaching the inspection location are reduced by 89%, as the oscillation magnitude remains below 1.0 degrees in the compensated case, compared to 9.1 degrees in the uncompensated case, as shown in Fig. 4.5(b).

4.4.2 Evaluating sensitivity to hyperparameters

Sensitivity of collision avoidance and potential oscillations with the proposed approach is investigated for (a) varying joystick gain α_1 that cause different speeds at which the obstacle is approached and (b) modeling errors in the cart mass M and the payload mass m. In all cases, collision is avoided and the increase in oscillations due to modeling error is removed by augmenting the differentially flat feedforward F_{ff} with feedback F_{fb}.

Simulation parameters

Collision avoidance is evaluated by investigating the horizontal payload response y_1 when starting motion from rest and exerting the maximum horizontal joystick command, $(j_1, j_2) = (1, 0)$ for 10 s, towards an obstacle, which has its nearest bounding box wall boundary at

Algorithm 1 Semi-Autonomous Controller

Given: Input gains α_1, α_2; Time horizon T; Bounding-box line segments L_w; Distance offset ϵ; $i \in \{0, \ldots, 4\}$; Cart mass M; Payload mass m; Feedback gains K

while true do

    Get joystick inputs j_1, j_2

    Get state feedback $X(t)$ from external camera

    $Y^{(i)}(t) \leftarrow$ Eqs. (4.13),(4.14),(4.16),(4.17), (4.19)

    $\dot{Y}_r \leftarrow \begin{bmatrix} \alpha_1 j_1 & \alpha_2 j_2 \end{bmatrix}^T$

    $\tilde{Y}_r \leftarrow Y(t) + \dot{Y}_r T$

    $l_r \leftarrow \overline{Y(t)\tilde{Y}_r}$

    $P \leftarrow \{Y_p \mid Y_p = l_r \cap l_{w,n},\ n \in \{0, 1, \ldots, N\}\}$

    $Y_r \leftarrow$ Eq. (4.67)

    $Y_d^{(i)}(\tau) \leftarrow$ Eq. (4.69)

    $X_d(\tau) \leftarrow$ Eqs. (4.44)-(4.48)

    $f_1, f_2 \leftarrow$ Eqs. (4.64),(4.65)

    Apply f_1 and f_2 to actuators

end while

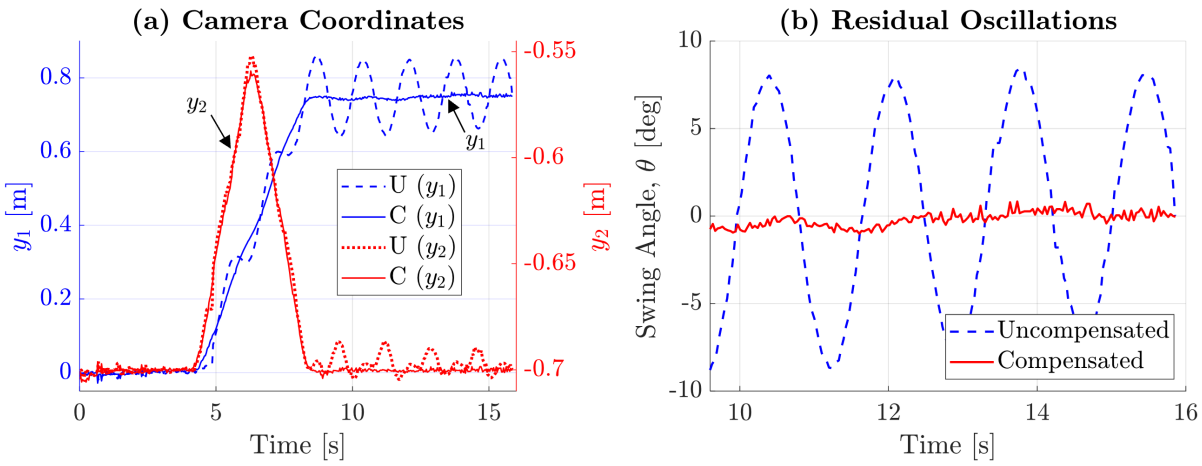


Figure 4.5: Tracking comparison of the fast ramp-like trajectory (transition time of $T_t = 4$ seconds) with compensated (C) and uncompensated (U) swing dynamics for the ramp-like trajectory described in Section 4.1. (a) Horizontal camera position, y_1, and vertical camera position y_2, and (b) residual oscillations after transitioning to the inspection location.

$y_1 = 0.75$ m, as shown in Fig.4.6(a). Nominal parameters are selected to match the human teleoperation experiments. Specifically, the cart and camera payload masses are varied from nominal values of $M = 0.815$ kg and $m = 0.225$ kg, respectively. The reference point generation uses a time horizon of $T = 1.5$ s, the scaling gain α_1 in the horizontal direction is varied around a nominal value of $\alpha_1 = 0.12$, and a distance offset of $\epsilon = 8$ cm is applied for collision avoidance, which match the nominal values in teleoperation experiments. Simulated horizontal trajectory responses y_1 for different hyperparameters are shown in Fig. 4.6, and discussed below.

Obstacle avoidance at varying speeds

Teleoperation trajectories are governed by joystick scaling gain, and the time horizon. To demonstrate the impact of speed variations, the horizontal camera coordinate, y_1, responses for five different values of horizontal-axis joystick gain α_1 are shown in Fig. 4.6(b). Note that successful collision avoidance is achieved by the algorithm when the joystick parameter

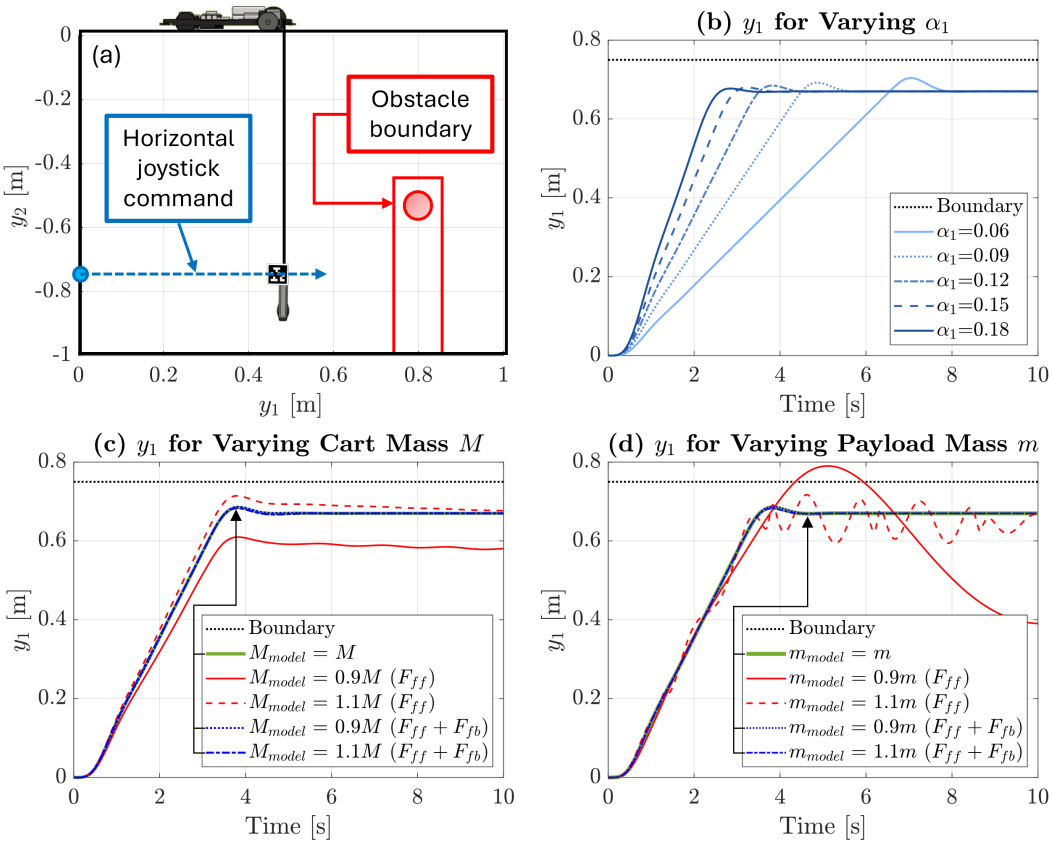


Figure 4.6: (a) Simulation setup for crane robot where the camera payload approaches an obstacle boundary horizontally. (b) Horizontal camera position, y_1, for varying joystick scaling gain, α_1, demonstrating successful collision avoidance with varying speeds due to changes in joystick gain parameter. Feedback F_{fb} reduces sensitivity of feedforward F_{ff} in the horizontal camera position, y_1 seen by comparing cases, with and without feedback F_{fb}, under (c) cart mass modeling errors of $\pm 10\%$ and (d) payload mass modeling errors of $\pm 10\%$.

is varied.

Robustness to modeling errors

The semi-autonomous controller demonstrates robustness to modeling errors due to its feedback terms. The modeled cart mass, M_{model}, is varied by $\pm 10\%$. The response to erroneous flatness-based feedforward input terms, derived from Eqs. (4.64) and (4.65), is analyzed by

setting all feedback gains to $K = 0$, leading to large errors, as shown in Fig. 4.6(c). Introducing feedback gains of $k_1 = 0.01$ N/mm, $k_2 = 0.01$ N/mm·s, $k_3 = -0.01$ N/rad, $k_4 = 0$ N/rad·s, $k_5 = 0.01$ N/mm, and $k_6 = 0.01$ N/mm·s stabilizes trajectories around the expected response. The simulation is repeated with the modeled camera payload mass, m_{model}, varied by $\pm 10\%$, with similar results shown in Fig. 4.6(d). Errors from flatness-based feedforward terms demonstrate greater sensitivity to camera payload mass variations compared to cart mass variations, but feedback gains successfully address sensitivity in both cases.

4.4.3 User study

Evaluated against the conventional industrial gantry crane control approach of decoupled velocity control (VC) [37] without swing-dynamics compensation, which relies on operator compensation of oscillations and collision avoidance, the semi-autonomous control (SC) improved efficiency and safety for 12 participants in a fastener inspection task. While the main result of this user study captures cumulative participant improvements, participant-level results are discussed in Appendix A.1. Exemption of this study was granted by the University of Washington Institutional Review Board (#STUDY00021303).

Fastener inspection task description

The fastener inspection task asks participants to move across the confined space over a pipeline obstacle to identify whether three fasteners are properly seated (i.e. no gaps under the fastener), as depicted in Fig. 4.7. Each participant performed three trials; in each trial, the participant performed the task twice in a single-blind manner, once with VC and once with SC. A pseudorandom number generator specified the order that each controller was used as well as the fastener gap configuration. The pseudo-random number generator was used to reduce learning bias and other order-related effects, ensuring that neither control method benefits disproportionately from participant practice during earlier tasks. To complete the task, participants recorded which fasteners contained gaps before capturing an image from the camera payload and confirming the task completion. In the event of collision,

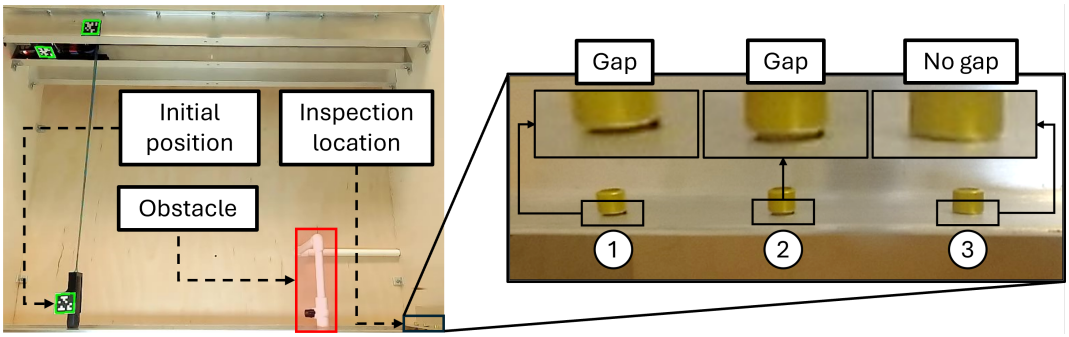


Figure 4.7: The experimental setup for the fastener inspection task. Participants move the crane robot from an initial position over the pipeline obstacle to the inspection location with three fasteners. The sample inset image shows fasteners 1 and 2 with gaps (which can be observed visually), while fastener 3 is properly seated and has no gap.

the task was recorded as a failed attempt, and the participant restarted the task. After each task, participants completed a questionnaire based on a seven point Likert scale [85, 86]. Appendix B.1 contains comprehensive procedures for conducting fastener inspection tele-operation experiments, designed to provide planned test durations and deliver consistent participant instructions.

Experiment parameters

Experiments were completed by moving the camera payload from rest at initial output coordinates, (y_1, y_2), of $(0, -0.72)$ m to inspect three fasteners seated at $(0.88, -0.72)$ m. The reference velocity, $\dot{Y}_r$ was generated by scaling the joystick inputs, j_1 and j_2 ranging from $[-1, 1]$, by gains $\alpha_1 = 0.12$ and $\alpha_2 = 0.04$, respectively, as in Eq. (4.66). The cart mass, M, and camera payload mass, m, were 0.815 kg and 0.225 kg, respectively. VC tracked $\begin{bmatrix} \dot{x} & \dot{l} \end{bmatrix}^T = \dot{Y}_r$ using feedback control, while SC applied Alg. 1 to plan trajectories over a time horizon, $T = 1.5$ s, ensuring trajectories remain in the confined space bounded virtually by $y_1 \in [0, 0.88]$ m and $y_2 \in [-0.75, -0.3]$ m and avoided collision with the pipeline obstacle modeled as a bounding box parameterized by $y_1 \in [0.575, 0.65]$ m and $y_2 \in [-0.75, -0.5]$ m using a distance offset of $\epsilon = 8$ cm for a conservative buffer. Feedback gains were experimentally

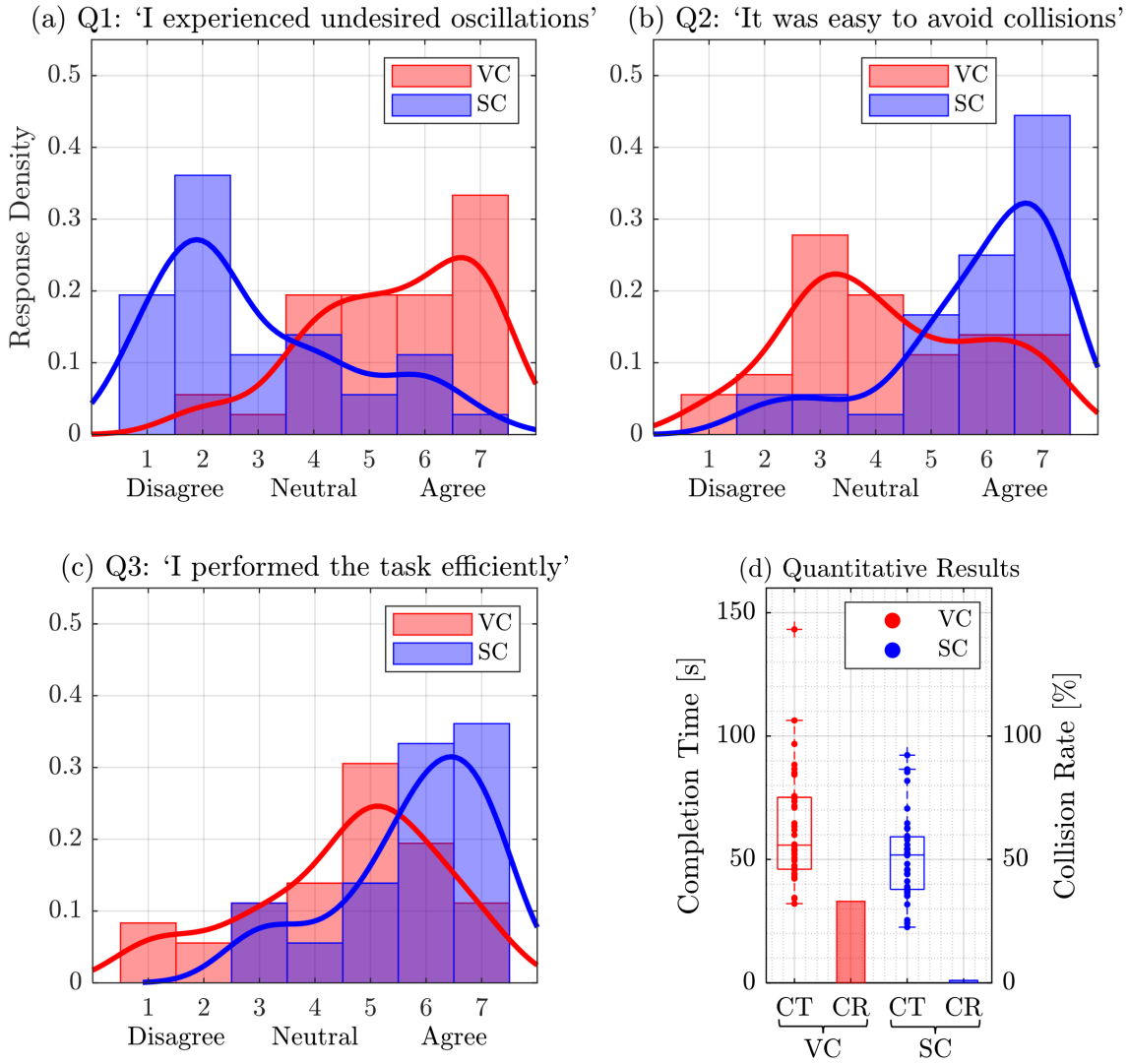


Figure 4.8: Results from 36 teleoperated fastener inspection trials. (a)-(c) Histogram of response density from the subjective questionnaire from a seven point Likert scale (1: Strongly disagree; 4: Neutral; 7: Strongly agree) comparing responses collected after using VC (without swing-dynamics compensation) and SC (with swing-dynamics compensation). The general trend of response density is represented as a line generated through a squared exponential kernel smoothing with a bandwidth of 0.7. (d) Quantitative measures of task completion time (CT) and collision rate (CR) for decoupled velocity control (VC) and semi-autonomous control (SC).

tuned to address perturbations. Gains $k_1 = 0.8\ \frac{\text{N}}{\text{mm}}$ and $k_2 = 0.8\ \frac{\text{N}}{\text{mm·s}}$ for cart position, x, and velocity, $\dot{x}$, were increased incrementally until tracking error was reduced without overshoot for cart trajectories (e.g. tracking the horizontal trajectory, $y_1(\tau)$, in Section 4.1). Similarly, gains $k_5 = 0.2\ \frac{\text{N}}{\text{mm}}$ and $k_6 = 0.2\ \frac{\text{N}}{\text{mm·s}}$ for camera payload length, l, and velocity, $\dot{l}$, were incrementally increased until error was reduced without overshoot when tracking length trajectories (e.g., tracking the vertical trajectory, $y_2(\tau)$, in Section 4.1). The gain $k_3 = -0.05\ \frac{\text{N}}{\text{rad}}$ for the swing angle, θ, was incrementally increased to suppress disturbed oscillations within a few cycles.

Reduced oscillation

Participants reported a decrease in undesired oscillations when using SC compared to VC, as shown through responses to Q1 in Fig. 4.8(a), supported by a Wilcoxon signed rank test (applied where both distributions are not normal under the Shapiro-Wilk test ($p < 0.05$)), which revealed a statistically significant difference between the two controllers, rejecting the null hypothesis ($p < 0.05$). Experiencing an increase in undesired oscillations while using VC compared with SC is consistent with the autonomous experiments of Section 4.4.1, as VC leaves unaccounted residual oscillations for participants while SC leverages the differential flatness to remove oscillations.

Safer inspection

The use of SC resulted in safer inspection for both the crane robot and surrounding aircraft structure with a collision rate of 0% compared to a collision rate of 33% under VC (Fig. 4.8(d)). The combination of reduced uncontrolled oscillation and reference point corrections led to participants reporting an ease of avoiding collisions through Q2 while using SC compared with VC (Fig. 4.8(b)). A Wilcoxon signed rank test confirmed this perception, rejecting the null hypothesis ($p < 0.05$).

Improved efficiency

Even neglecting failures due to collisions with VC, SC improved inspection efficiency by reducing mean task completion time by 18.7% (Fig. 4.8(d)) to 51.2 s compared to a mean task completion time under VC of 63.0 s. A paired-t test demonstrated a statistically significant difference between responses collected from VC and SC, rejecting the null hypothesis ($p < 0.05$). Participants also perceived improved task efficiency using SC as opposed to VC through Q3 of the subjective questionnaire (Fig. 4.8(c)), where a Wilcoxon signed rank test confirmed this perception, rejecting the null hypothesis ($p < 0.05$).

4.5 Conclusion

This chapter presented a crane robot for teleoperated in-wing confined space inspection. To remove undesired oscillations during teleoperation, the swing dynamics of the crane robot are accounted for by exploiting the differentially-flat dynamics to generate sufficiently smooth trajectories for tracking, while avoiding collision with surrounding obstacles. This enabled semi-autonomous control with reduced undesired oscillations, eliminated collisions, and enhanced inspection efficiency during teleoperation.

Chapter 5

MC2: OUTPUT-SAMPLED MODEL PREDICTIVE PATH INTEGRAL CONTROL FOR A CRANE ROBOT

This chapter presents MC2, which covers the development of an output-sample model predictive path integral control (oMPPI) for the crane robot active vision system, providing a dynamic viewpoint to operators. This chapter formulates the active vision system and MPPI control problem through experiments, develops oMPPI with inversion theory, and experimentally validates control improvements of oMPPI compared with standard MPPI through teleoperation experiments. This chapter is based on work in [17].

5.1 Problem Formulation

The confined-space active vision application is used in the following sections to illustrate challenges with the standard MPPI approach.

5.1.1 System description

The teleoperated manipulator can move between different wing bays in Fig. 1.2 through channels formed by adjacent stiffening stringer structure along the lower skin, which create a passage between adjacent wing bays under the ribs. This avoids the typical need to manually reinstall teleoperated robots in each wing bay. To traverse stringer channels, the teleoperated manipulator is stowed within a motorized cart equipped with omni wheels, which passively realign with channels during cart motion. Since in-wing systems (that can cause potential collisions) are located near the walls of confined spaces (designed to accommodate human mechanics), the cart can position the manipulator to ensure that the manipulator links remain collision-free during unstowed navigation. Once positioned, linear actuators embedded

in the cart extend, pressing against the channel walls to secure the cart in place, providing a stable base for the manipulator to unstow and perform tasks. Confined space tasks require manipulator navigation along lower confined space surfaces, where cameras mounted at access holes cannot view manipulator configuration or obstacle proximity. Furthermore, the manipulator needs to move around unknown objects, such as foreign object debris (FOD) left behind on lower surfaces [87], which necessitates the use of the active vision systems to provide a situational view of the teleoperated manipulator and obstacles during navigation.

The active vision crane-robot provides the necessary situational view by traversing adjacent wing bays through stringer channels formed along the confined space ceilings. By following the teleoperated manipulator between bays, the crane robot removes the need for extensive static camera setups, which would otherwise require planning and installation for each unique wing bay configuration. Since the crane robot's cart is constrained to a channel, it can only track the manipulator's coordinate along that channel. To overcome this limitation, the crane robot's camera payload is capable of obtaining views in all directions, similar to other physically constrained robotic systems [76]. After moving over ribs into the desired wing bay, the crane robot suspends a camera payload to a fixed length l, providing an aerial view of the teleoperated manipulator's end effector and its surrounding obstacles.

The goal of MPPI is to select optimal controls for the crane robot position $y_p(\tau)$ at time τ to track the manipulator position y_m with a time delay T_p, i.e.,

$$y_p(\tau) = y_m(\tau - T_p). \tag{5.1}$$

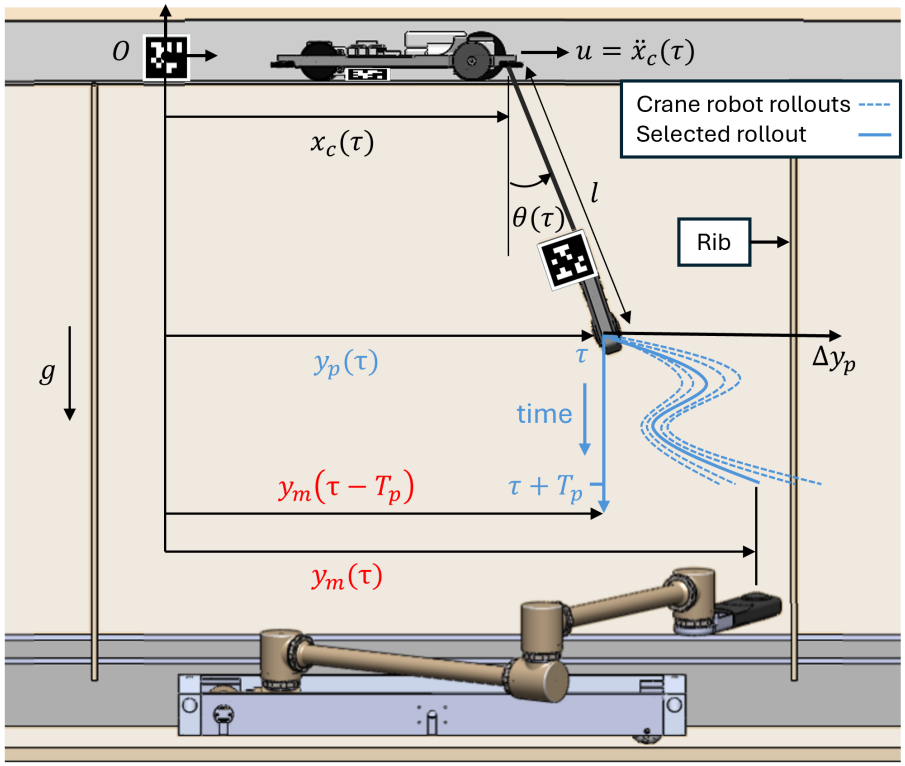


Figure 5.1: The crane-robot active vision schematic tracking a teleoperated manipulator. The crane robot has an input of cart acceleration $\ddot{x}_c$, states of cart position x_c, swing angle θ and their derivatives, and an output of the horizontal camera payload position $y_p(\tau)$ at time τ, which aims to track the teleoperated manipulator position $y_m(\tau - T_p)$ with a time delay T_p. MPPI rollouts for the future horizontal camera positions $y_p(\tau)$ over the time interval $[\tau, \tau + T_p]$ are shown in blue dashed lines.

5.1.2 Crane robot model

The crane-robot forward-dynamics is modeled as a fixed-length gantry crane, as in Fig. 5.1, with linearized dynamics (assuming small angles) as in [88]

$$\begin{aligned} \dot{x} &= F(x) + G(x)u \\ &= \begin{bmatrix} \dot{x}_c \\ 0 \\ \dot{\theta} \\ -\frac{g}{l}\theta \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ -\frac{1}{l} \end{bmatrix} \ddot{x}_c, \end{aligned} \tag{5.2}$$

where x represents the state of the system, x_c is the cart position, θ is the swing angle, $l = 0.5$ m is the fixed payload suspension length, $g = 9.81\ \frac{\text{m}}{\text{s}^2}$ is the gravitational acceleration, the system input u is the cart acceleration $\ddot{x}_c$, and the output y is the horizontal camera-payload position y_p

$$\begin{aligned} y = y_p &= H(x) \\ &= x_c + l\theta. \end{aligned} \tag{5.3}$$

Crane-robot dynamics in Eq. (5.2) are subject to constraints on the cart velocity $\dot{x}_c$ and acceleration $\ddot{x}_c$ imposed by the input motor limits of

$$|\dot{x}_c| \leq v_{max} \tag{5.4}$$

$$|\ddot{x}_c| \leq a_{max}, \tag{5.5}$$

where the experimentally-determined limits are $v_{max} = 0.3\ \frac{\text{m}}{\text{s}}$ and $a_{max} = 0.5\ \frac{\text{m}}{\text{s}^2}$ on the cart velocity and the cart acceleration, respectively. The dynamics are discretized for MPPI with Euler's method using a timestep $\Delta t = 0.05$ s to match the measurement frequency of the

manipulator

$$x_{t+1} = f(x_t, v_t) \tag{5.6}$$

$$= \begin{bmatrix} x_{c,t} + \dot{x}_{c,t}\Delta t \\ \dot{x}_{c,t} \\ \theta_t + \dot{\theta}_t \Delta t \\ \dot{\theta}_t - \frac{g}{l}\theta_t \Delta t \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta t \\ 0 \\ -\frac{1}{l}\Delta t \end{bmatrix} \ddot{x}_{c,t}. \tag{5.7}$$

where the input $v_t = \ddot{x}_{c,t} \sim \mathcal{N}(u_t, \Sigma)$ is a noisy sequence

$$V = \{v_0, \ldots, v_{T-1}\} \tag{5.8}$$

and direct control is available over its mean u_t.

5.1.3 MPPI for confined space active vision

MPPI cost function

Tracking of measured manipulator position y_m, with delay as in Eq. (5.1), and collision avoidance are achieved by optimizing the mean of the input sequences

$$U = \{u_0, u_1, \ldots, u_{T-1}\} \tag{5.9}$$

given a cost function $C(\cdot)$

$$C(u, x) = \phi(x_T) + \sum_{t=0}^{T-1} \left[c(x_t) + u_t^T \Sigma^{-1} u_t + \kappa^T u_t \right], \tag{5.10}$$

where κ enables shifting the input penalty mean from zero, across $T = T_p/\Delta t$ timesteps $t \in \{0, 1, \ldots, T-1\}$ over the MPPI receding horizon preview time T_p, where the state cost function $c(\cdot)$,

$$c(x_t) = c_c(x_t) + c_{tr}(x_t), \tag{5.11}$$

comprises of the collision cost $c_c(\cdot)$ and tracking cost $c_{tr}(\cdot)$ and the terminal cost is set equal to the state cost $\phi(x_T) = c(x_T)$. Collision costs $c_c(\cdot)$ use a weighted indicator function as in

autonomous vehicle obstacle boundary costs [89]

$$c_c(x_t) = \begin{cases} w_c, & \text{if collision} \\ 0, & \text{otherwise} \end{cases}, \tag{5.12}$$

where the collision weight $w_c = 1000$ strongly penalizes collision. Manipulator tracking costs $c_{tr}(\cdot)$ use a squared error term as in velocity and heading tracking costs used in unmanned surface vessels [90]

$$c_{tr}(x_t) = w_{tr} \left(y_m[t - T_p] - y_p[t] \right)^2, \tag{5.13}$$

where tracking weight $w_{tr} = 1$ weights squared error between the desired trajectory of the delayed manipulator trajectory $y_d = y_m[t - T_p]$ and crane robot output $y_p[t]$.

MPPI algorithm

MPPI samples N_r noisy input sequences V^i in Eq. (5.8)

$$V^i = \{v_0^i, \ldots, v_{T-1}^i\} = \{u_0^i + \epsilon_0^i, \ldots, u_{T-1}^i + \epsilon_{T-1}^i\}, \tag{5.14}$$

where ϵ^i represents sampled noise sequences

$$\mathcal{E}^i = \{\epsilon_0^i, \ldots, \epsilon_{T-1}^i\}, \quad \epsilon^i \sim \mathcal{N}(0, \Sigma), \tag{5.15}$$

with covariance matrix Σ and over the mean input U in Eq. (5.9). Each sampled input sequence V^i is simulated to find the response of the forward dynamics $f(x_t^i, v_t^i)$ in Eq. (5.6) to obtain the corresponding state X^i and output Y^i sequences resulting in the rollout triplet

$$\left(V^i, X^i, Y^i\right). \tag{5.16}$$

Then, importance sampling weights w_i are computed as

$$w_i = \frac{1}{\eta} \exp\left(-\frac{1}{\lambda} \left(S(V^i) + \gamma \sum_{t=0}^{T-1} (u_t^i)^T \Sigma^{-1} v_t^i - \delta \right) \right), \tag{5.17}$$

where $\exp(\cdot)$ denotes the exponential function, $S(V^i)$ is the state-dependent cost $C(V^i, X^i)$ in Eq. (5.10) of sample V^i that is found using the rollout triplet in Eq. (5.16), η is the

normalization constant, λ is the inverse temperature, δ is the minimum sampled cost that shifts weights to avoid numerical instability, and γ is the control weight that decouples the control costs from the inverse temperature [8]. The input U is updated as

$$u_t = u_t + \sum_{i=1}^{N_r} w(V^i)\epsilon_t^i, \tag{5.18}$$

the first weighted input u_o is applied to the system, and the input sequence U is shifted to the next time instant for the next step. This MPPI approach is summarized in Algorithm 2.

MPPI hyper-parameters

The hyper-parameters for MPPI were selected as number of rollouts $N_r = 100$, inverse temperature $\lambda = 0.1$, control weight $\gamma = 0.001$, and acceleration standard deviation $\sigma_a = 0.1\ \frac{\mathrm{m}}{\mathrm{s}^2}$ unless otherwise specified. The control weight γ was applied to promote trajectory tracking, and the number of rollouts $N_r = 100$ was selected such that adding more rollouts did not improve performance.

5.1.4 Problem: tracking degrades with reduced preview time

Reducing the preview time T_p is beneficial to decrease the time lag between the active-vision camera position $y_p(\tau)$ and the teleoperated-manipulator position $y_m(\tau)$. However, a sufficiently-large time delay T_p is needed to provide sufficient exploration time for MPPI rollouts in the crane-robot position y_p over the time interval $[\tau, \tau + T_p]$, as illustrated in Fig. 5.1. Such exploration over the preview time T_p is needed to achieve precision (delayed) tracking as in Eq. (5.1), by estimating and correcting for the system dynamics, such as the pendulum-type swing dynamics θ. The problem of performance degradation with MPPI, as the preview time T_p decreases, is clarified below using experimental results.

Fixed desired trajectory for experiments

To evaluate MPPI performance, the desired manipulator trajectory is kept constant as the preview time T_p is varied. Specifically, the desired positioning for the crane robot is defined

Algorithm 2 MPPI Algorithm

Given: Forward model f, number of rollouts N_r, number of timesteps T, initial control sequence $\{u_0, \ldots, u_{T-1}\}$, covariance Σ, state dependent cost function $c(\cdot)$, terminal cost function $\phi(\cdot)$, control weight γ, inverse temperature λ
while task not completed **do**
    Get state estimate x_0
    for rollouts $i = 1$ to N_r **do**
        $x_0^i \leftarrow x_0$
        Sample $\mathcal{E}^i = \{\epsilon_0^i, \ldots, \epsilon_{T-1}^i\}$, $\epsilon^i \sim \mathcal{N}(0, \Sigma)$
        for $t = 0$ to $T - 1$ **do**
            Apply noise $v_t^i = u_t^i + \epsilon_t^i$
            Simulate forward $x_{t+1}^i \leftarrow f(x_t^i, v_t^i)$ (Eq. (5.7))
            Find output $y_t^i \leftarrow H(x_t^i)$ (Eq. (5.3))
            Accumulate running costs (Eq. (5.10))
                $S_i(V^i)$ += $c(x_t^i) + \gamma u_t^{i^T} \Sigma^{-1} \epsilon_t^i$
        end for
        Terminal cost $S_i(V^i)$ += $\phi(x_T^i)$ (Eq. (5.10))
    end for
    $\delta = \min_i(S_i)$
    Compute weights $w(S_1, \ldots, S_{N_r})$ (Eq. (5.17))
    for $t = 0$ to $T - 1$ **do**
        Update control u_t += $\sum_{i=1}^{N_r} w_i \epsilon_t^i$
    end for
    for $t = 1$ to $T - 1$ **do**
        Shift inputs $u_{t-1} \leftarrow u_t$
    end for
    $u_{T-1} \leftarrow 0$
end while

to represent the manipulator moving back and forth within the confined space with an acceleration profile

$$\ddot{y}_m(\tau) = \begin{cases} 0 & 0 \leq \tau < \frac{T_t}{2} \\ \frac{2\pi y_{\text{step}}}{T_t^2} \sin\left(\frac{2\pi}{T_t}\tau\right) & \frac{T_t}{2} \leq \tau < \frac{3T_t}{2} \\ y_{\text{step}}\left(1 - \frac{2\pi}{T_t^2}\right) \sin\left(\frac{2\pi}{T_t}\tau\right) & \frac{3T_t}{2} \leq \tau < \frac{5T_t}{2} \\ 0 & \frac{5T_t}{2} \leq \tau < 3T_t, \end{cases}$$

where $y_{\text{step}} = 0.6$ m is the amplitude of movement in each directional over a time interval $T_t = 10$ s. The trajectory y_m obtained by integrating the above acceleration profile $\ddot{y}_m(\tau)$ is augmented with noise to represent the measured desired manipulator trajectory $\tilde{y}_m \sim \mathcal{N}(y_m, \sigma_m^2)$, with $\sigma_m = 0.01$ m, shown in Fig. 5.2. The tracking error e between the achieved position y_p of the crane robot and the desired position y_m is defined as

$$e = e_{RSME}(y_m, y_p, 0) \tag{5.19}$$

where $e_{RSME}(\cdot, \cdot, \cdot)$ is the general root mean square error between two functions over time with a delay in one function, i.e.,

$$e_{RSME}(y_1, y_2, \tau_p) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(y_1(\tau_i - \tau_p) - y_2(\tau_i)\right)^2} \tag{5.20}$$

for n samples at time instants $\{\tau_i\}_{i=1}^n$.

Need for large preview time T_p

Small tracking error e in Eq. (5.19) requires a sufficiently-large preview time T_p, as seen in Fig. 5.3(a), which shows experimental results when the noisy fixed desired trajectory $\tilde{y}_m(\tau)$ in Fig. 5.2 is tracked seven times for preview times decreasing from 2 s in increments of 0.25 s. For example the mean error decreases from 0.32 m at $T_p = 0.25$ s to 0.087 m at $T_p = 1.5$ s. With preview times $T_p \leq 0.5$ s, the responses tend to reach local minima, resulting in collisions with boundaries set for the experimental confined space at $y = -0.1$ m and

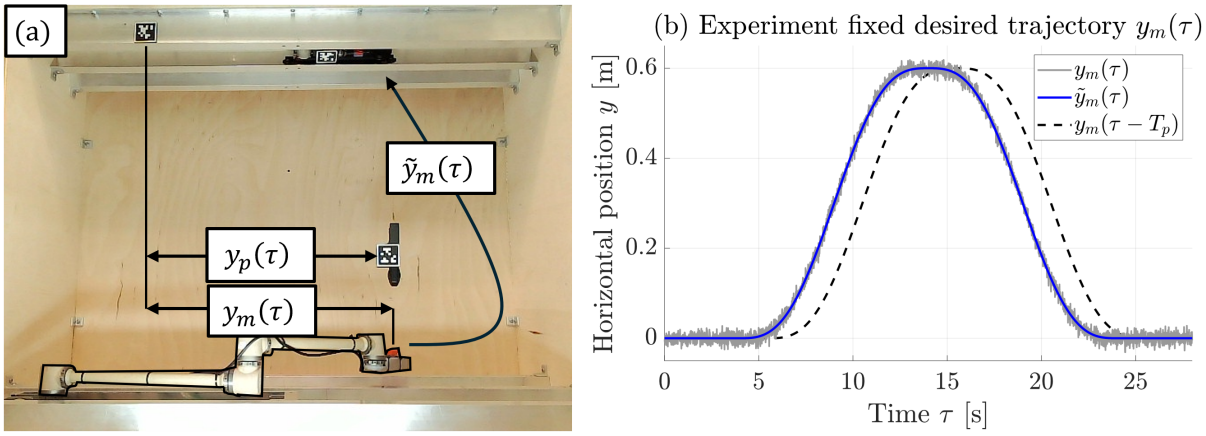


Figure 5.2: The experimental crane robot tracks a fixed trajectory representing the manipulator position. (a) The crane robot aims to track the manipulator position y_m with its horizontal camera position y_p. The crane robot controller receives noisy measurements y_m for tracking. (b) For comparative evaluations, the fixed trajectory $y_m(\tau)$ (solid, blue) moves back and forth in the confined space following a sinusoidal acceleration profile $\ddot{y}_m(\tau)$ and noise is added to this trajectory to simulate the measured trajectory $\tilde{y}_m(\tau)$ (solid, grey) to the tracked by the crane robot. Delay from a large preview time T_p introduces a significant tracking error, e_{m,T_p} in Eq. (5.21), between the manipulator trajectory $y_m(\tau)$ and its delayed version $y_m(\tau - T_p)$ (dashed, black).

$y = 0.7$ m. Additionally, as seen in the representative trajectory responses in Fig. 5.3(b), oscillations occur at the pendulum's period of oscillation when the preview time is reduced to $T_p = 1.0$ s, suggesting that swing dynamics are not fully compensated by the MPPI control. At a further reduction to preview time of $T_p = 0.75$ s, trajectories overshoot from a lack of exploration. Thus, a sufficiently-large preview time T_p is needed for MPPI to minimize the tracking error e in Eq. (5.19) between the manipulator position y_m and the crane-robot position y_p. Increasing the number of rollouts from $N_r = 100$ showed no significant improvement in results, as seen in Fig. 5.4, where seven trials were conducted for each varied number of rollouts N_r, with a fixed preview time of $T_p = 1.5$ s

The tracking error e in Eq. (5.19) is comprised of two parts:

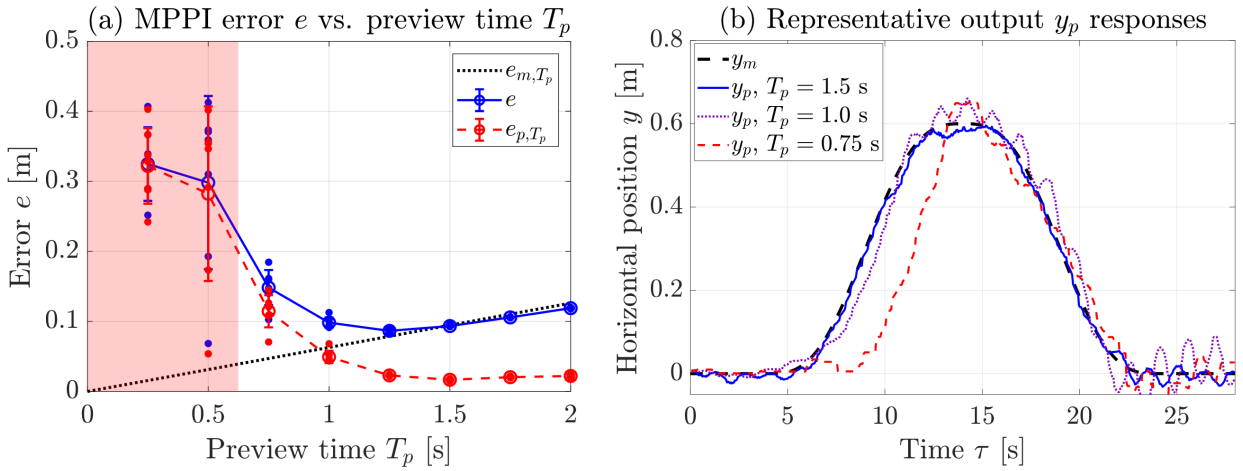


Figure 5.3: MPPI experimental responses tracking the fixed trajectory with noise $\tilde{y}_m(\tau)$ in Fig. 5.2. (a) Small tracking error e in Eq. (5.19) requires a sufficiently-large preview time T_p. Error e (blue, solid) is a combination of error e_{m,T_p} (black, dotted) due to the delay in the reference trajectory y_m that increases with preview time T_p and the error e_{p,T_p} (red, dashed) due to inability to track the delayed reference trajectory y_m that decreases with preview time T_p. Trials with small preview times of $T_p = 0.5$ s and $T_p = 0.25$ s tend to reach local minima and collide with the physical confined space as indicated by the (red) shaded region. (b) Comparison of time-shifted output responses $y_p(\tau + T_p)$ with the fixed manipulator trajectory $y_m(\tau)$ (black, thick dashed) show that decreasing the preview time with MPPI from $T_p = 1.5$ s (blue, solid) to $T_p = 1.0$ s (purple, dotted) increases oscillations, while at even smaller $T_p = 0.75$ s (red, thin dashed) responses have substantial overshoot.

1. the error e_{m,T_p}

$$e_{m,T_p} = e_{RSME}(y_m, y_m, T_p) \tag{5.21}$$

due to the delay in the reference trajectory $y_m(\tau)$, as shown in Fig 5.2(b), and

2. the error e_{p,T_p}

$$e_{p,T_p} = e_{RSME}(y_m, y_p, T_p) \tag{5.22}$$

due to inability to track the delayed reference trajectory $y_m(\tau - T_p)$.

Note that the ability to track the delayed reference trajectory $y_m(\tau - T_p)$ improves substantially with larger preview time T_p as seen in Fig. 5.3(a). For example, mean errors to the

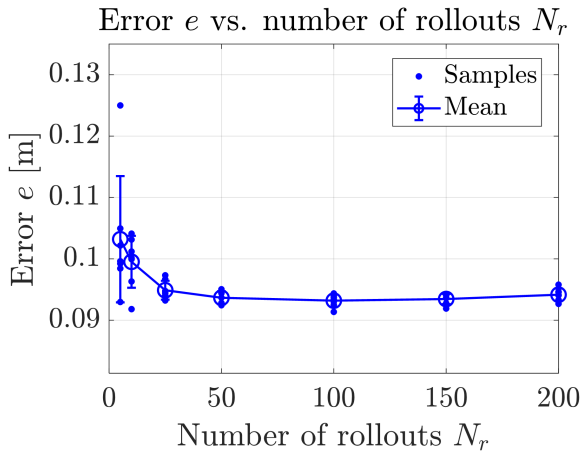


Figure 5.4: Tracking errors e do not significantly vary as the number of rollouts N_r is increased from the nominal value of $N_r = 100$ at a fixed preview time $T_p = 1.5$ s. Performance tends to degrade as the number of rollouts is reduced below $N_r < 50$.

delayed signal e_{p,T_p} decrease from 0.32 m at $T_p = 0.5$ s to 0.016 m at $T_p = 1.5$ s. However, the larger preview time T_p also implies that there is a large error e_{m,T_p} between the manipulator position $y_m(\tau)$ and the delayed reference trajectory $y_m(\tau - T_p)$, which leads to increased error e between the crane robot position $y_p(\tau)$ and the actual manipulator position $y_m(\tau)$ as seen in Fig. 5.3(a). For example, with a preview time of $T_p = 1.5$ s, the mean tracking error was $e = 0.087$ m, where the mean error e_{p,T_p} in tracking the delayed signal was only 0.023 m. Therefore, the main component of the error e is the error e_{m,T_p} due to the large time delay T_p, which is, however, necessary for precision tracking with MPPI.

5.1.5 Research problem

Improve the sampling-based optimization for active-vision systems by enabling precision tracking and reducing the error e in Eq. (5.19) between the manipulator position $y_m(\tau)$ and the crane-robot position $y_p(\tau)$, by both (i) reducing the error e_{p,T_p} in tracking the desired delayed trajectory $y_m(\tau - T_p)$ and (ii) reducing the delay-caused error e_{m,T_p} — by enabling precision tracking with a smaller required preview time T_p.

5.2 Proposed oMPPI approach

The proposed oMPPI method achieves precision tracking under decreased preview time T_p without increasing the number of rollouts N_r compared to MPPI by sampling in the output space using inverse models, as described next..

5.2.1 Inverse Models

Consider a system with nonlinear dynamics given by

$$\begin{aligned} \dot{x} &= F(x) + G(x)\,u, \\ y &= H(x), \end{aligned} \tag{5.23}$$

where input $u(t) \in \mathbb{R}^m$, state $x(t) \in \mathbb{R}^n$, output $y(t) \in \mathbb{R}^\rho$ and sufficiently smooth functions $F(x)$, $G(x)$ and $H(x)$.

Tracking-normal form

The system can be rewritten in its tracking-normal form as [91]

$$\dot{\xi}_{i,j} = \xi_{i,j+1}, \quad 1 \le i \le \rho, \quad 1 \le j \le r_i - 1, \tag{5.24}$$

$$y^{(r)} = \alpha(x) + \beta(x)u, \tag{5.25}$$

$$\dot{\eta} = \psi_1(x) + \psi_2(x)u, \tag{5.26}$$

where ξ_i represents each output y_i and its $r_i - 1$ time derivatives

$$\xi_i = \left[y_i, \dot{y}_i, \ldots, \frac{d^{r_i-1} y_i}{dt^{r_i-1}} \right]^T \in \mathbb{R}^{r_i}, \tag{5.27}$$

r_i is the number of times an output y_i needs to be differentiated before the input u directly influences it, and the vector of all r_i is r. The relative degree of the system is defined as

$$|r| = \sum_{i=1}^{\rho} r_i \le n. \tag{5.28}$$

The superscript in brackets (r) in Eq. (5.25) indicates time derivatives

$$y^{(r)} = \left[\frac{d^{r_1}y_1}{dt^{r_1}}, \frac{d^{r_2}y_2}{dt^{r_2}}, \ldots, \frac{d^{r_\rho}y_\rho}{dt^{r_\rho}}\right]^T. \tag{5.29}$$

The term η in Eq. (5.26) represents $n - |r|$ internal states, such that there is a coordinate transformation between the original state x and the output and its time derivatives ξ along with the internal state η

$$\begin{bmatrix}\xi \\ \eta\end{bmatrix} = \Phi(x), \tag{5.30}$$

where

$$\xi = \left[\xi_1^T, \xi_2^T, \ldots, \xi_\rho^T\right]^T. \tag{5.31}$$

The entries of the vector $\alpha(x) \in \mathbb{R}^\rho$ and matrix $\beta(x) \in \mathbb{R}^{\rho\times m}$ in Eq. (5.25) can be found in terms of Lie derivatives

$$\alpha_i(x) = L_F^{r_i} H_i(x), \tag{5.32}$$

$$\beta_{i,k}(x) = L_{G_k} L_F^{r_i-1} H_i(x), \tag{5.33}$$

for $i \in \{1, 2, \ldots, \rho\}$ and $k \in \{1, 2, \ldots, m\}$ where

$$L_F^0 H_i(x) = H_i(x), \tag{5.34}$$

$$L_F^k H_i(x) = \frac{\partial L_F^{k-1} H_i(x)}{\partial x} F(x), \quad k \geq 1, \tag{5.35}$$

$$L_{G_k} L_F^{r_i-1} h_i(x) = \frac{\partial L_F^{r_i-1} H_i(x)}{\partial x} G_k(x). \tag{5.36}$$

Invertibility conditions

The inverse input u can be found for a given sufficiently-smooth desired trajectory y from Eq. (5.25) provided the following two invertibility conditions are met.

Invertibility conditions

The inverse input u can be found for a given sufficiently-smooth desired trajectory y from Eq. (5.25) provided the following two invertibility conditions are met.

The first condition is that the rank of matrix $\beta(x)$ in Eq. (5.25) is ρ, $\text{rank}(\beta(x)) = \rho$, which requires that there are at-least as many inputs as outputs, i.e., $m \geq \rho$, the rows of $\beta(x)$ are linearly independent, and $\beta(x)\beta(x)^T$ is positive definite and invertible. Then, the pseudoinverse $\beta^\dagger$ of β is

$$\beta^\dagger = \beta(x)\left(\beta(x)\beta(x)^T\right)^{-1}. \tag{5.37}$$

The second condition is that the smallest eigenvalue of $\beta(x)\beta(x)^T$ needs to remain bounded away from zero to prevent the pseudo inverse $\beta^\dagger$ from becoming unbounded, i.e.,

$$\lambda_{\min}\left(\beta(x)\beta(x)^T\right) \geq \tilde{\epsilon} > 0, \tag{5.38}$$

in the region of interest.

Inverse

Inverse

For a given sufficiently smooth desired trajectory y_d the inverse is found by solving the internal dynamics η_d in Eq. (5.26)

$$\dot{\eta}_d = \psi_1(x_d) + \psi_2(x_d)\beta^\dagger(x_d)\left(-\alpha(x_d) + y_d^{(r)}\right) \tag{5.39}$$

and then finding the inverse input $u = u_{ff}$ from Eq. (5.25) as

$$u_{ff} = \beta^\dagger(x_d)\left(-\alpha(x_d) + y_d^{(r)}\right), \tag{5.40}$$

where

$$x_d = \Phi^{-1}\left(\begin{bmatrix}\xi_d \\ \eta_d\end{bmatrix}\right). \tag{5.41}$$

Thus, the sufficiently-smooth desired output $y = y_d$ leads to the state $x = x_d$ and input $u = u_{ff}$.

Remark 5. *Bounded solutions of the internal dynamics in Eq.* (5.39) *can be found, even for nonminimum-phase systems where the internal dynamics are unstable, using a preview-based-approach [92], which is well suited to sampling-based methods since preview information is available.*

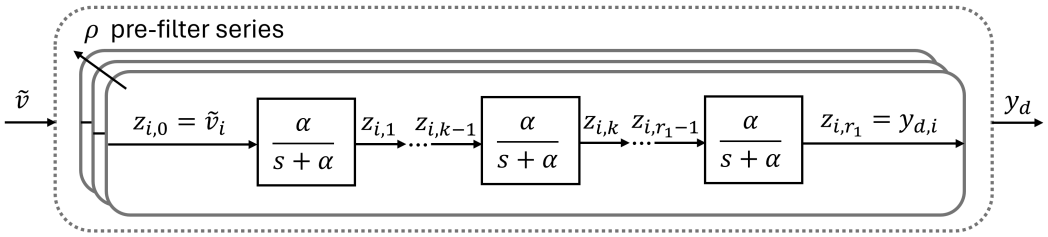


Figure 5.5: Smoothing pre-filters creating feasible trajectories y_d from, e.g., measured noisy signals $\tilde{v}$. A feasible trajectory y_d is generated by passing each noisy output measurement $\tilde{v}_i$ through a series of r_i first order filters for $i \in \{1, 2, \dots \rho\}$.

Remark 6. *The inverse model to compute the feedforward input u_{ff} for a given desired output y_d can be found using machine learning approaches from measured input-output data [72].*

5.2.2 Smoothing pre-filters for feasibility

Inverse models require sufficiently-smooth desired trajectories y_d. However, the sampled reference trajectories $\tilde{v}$ are noisy, and not smooth. Therefore, smoothing filters are applied to find the desired output y_d and its derivatives, before the inverse model is used as illustrated in Fig. 1.3. Specifically, let $z_{i,0} \triangleq \tilde{v}_i$ be the sampled reference output that needs to be r_i times differentiable for $i \in \{1, 2, \dots, \rho\}$. Each reference output $y_{d,i}$ is passed through r_i cascaded first-order filters in Fig. 5.5,

$$\dot{z}_{i,1} = -\alpha z_{i,1} + \alpha z_{i,0} \tag{5.42}$$

$$\dot{z}_{i,2} = -\alpha z_{i,2} + \alpha z_{i,1} \tag{5.43}$$

$$\vdots$$

$$\dot{z}_{i,r_i} = -\alpha z_{i,r_i} + \alpha z_{i,r_i-1}, \tag{5.44}$$

resulting in a smooth output

$$y_{d,i} = z_{i,r_i}. \tag{5.45}$$

Lemma 4. *The time derivatives of the smooth output $y_{d,i}$ can be expressed as*

$$y_{d,i}^{(k)} \;=\; z_{i,r_i}^{(k)} \;=\; \alpha^k \sum_{j=0}^{k} (-1)^j \binom{k}{j} z_{i,\,r_i-(k-j)}, \tag{5.46}$$

for $1 \le k \le r_i$.

Proof. Consider a base case of $k = 1$. Eq. (5.46) holds as

$$\dot{y}_{d,i} \;=\; \alpha \sum_{j=0}^{1} (-1)^j \binom{1}{j} z_{i,r_i-(1-j)} \;=\; \alpha \Big[z_{i,r_i-1} - z_{i,r_i} \Big]. \tag{5.47}$$

Assume Eq. (5.46) holds for some $k \ge 1$. Differentiating the expression leads to

$$z_{i,r_i}^{(k+1)} \;=\; \frac{d}{dt} z_{i,r_i}^{(k)} \;=\; \alpha^k \sum_{j=0}^{k} (-1)^j \binom{k}{j} \dot{z}_{i,r_i-(k-j)}. \tag{5.48}$$

Substituting the filter dynamics of

$$\dot{z}_{i,r_i-(k-j)} \;=\; \alpha \big(z_{i,r_i-(k-j)-1} \;-\; z_{i,r_i-(k-j)} \big) \tag{5.49}$$

results in

$$z_{i,r_i}^{(k+1)} = \alpha^{k+1} \left[\sum_{j=0}^{k} (-1)^j \binom{k}{j} z_{i,r_i-(k+1-j)} \;-\; \sum_{j=0}^{k} (-1)^j \binom{k}{j} z_{i,r_i-(k-j)} \right]. \tag{5.50}$$

Shifting the index of the second summation

$$z_{i,r_i}^{(k+1)} = \alpha^{k+1} \left[\sum_{j=0}^{k} (-1)^j \binom{k}{j} z_{i,r_i-(k+1-j)} \;+\; \sum_{j=1}^{k+1} (-1)^j \binom{k}{j-1} z_{i,r_i-(k+1-j)} \right], \tag{5.51}$$

combining summations

$$z_{i,r_i}^{(k+1)} \;=\; \alpha^{k+1} \sum_{j=0}^{k+1} (-1)^j \left(\binom{k}{j} + \binom{k}{j-1} \right) z_{i,r_i-(k+1-j)}, \tag{5.52}$$

and applying Pascal's identity

$$\binom{k+1}{j} \;=\; \binom{k}{j} \;+\; \binom{k}{j-1} \tag{5.53}$$

results in

$$z_{i,r_i}^{(k+1)} \;=\; \alpha^{k+1} \sum_{j=0}^{k+1} (-1)^j \binom{k+1}{j} z_{i,r_i-(k+1-j)}. \tag{5.54}$$

Therefore, the expression holds for $k+1$, and by induction it follows that the derivatives $z_{i,r_i}^{(k)}$ follow an alternating Pascal's triangle pattern, proving the lemma. □

Remark 7. *The concatenation of filter states for all outputs is denoted* $z \in \mathbb{R}^{\left(\sum_{i=1}^{\rho} r_i\right)}$.

Remark 8. *Filters are discretized with Euler's method for use in oMPPI as*

$$z_{i,t+1} = z_{i,t} + \alpha \left(z_{i-1,t} - z_{i,t}\right) \Delta t. \tag{5.55}$$

5.2.3 Constructing the full system dynamics with inverse

By sequentially combining the smoothing dynamics (e.g., the cascaded pre-filters of Eqs. (5.42)-(5.44)), the inverse model in Eq. (5.40), and the forward dynamics of Eq. (5.23), as in Fig. 5.6, and subsequently discretizing, the full system dynamics can be described generally as

$$\tilde{x}_{t+1} = \tilde{f}(\tilde{x}_t, \tilde{v}_t), \tag{5.56}$$

where the new input $\tilde{v}$ represents a noisy output reference with mean $\tilde{u}$, i.e., $\tilde{v} \sim \mathcal{N}(\tilde{u}, \Sigma)$, generated from noisy environment measurements. The state of the full system $\tilde{x} = \left[x^T, u^T, z^T\right]^T$ combines states x and inputs $u = u_{ff}$ of the original system with filter states z.

Remark 9. *In the full system in Eq.* (5.56) *the computed inversion-based feedforward input* $u = u_{ff}$ *that is applied to the system is noise-free. Noise* ϵ*, e.g., due to manipulator actuation and measurements, modulates the reference* $\tilde{u}$ *applied to the system, as shown in Fig. 5.7. Thus, the reference input* $\tilde{v}$ *to the full system in Eq.* (5.56) *is noisy.*

5.2.4 Optimality of oMPPI sampling

The MPPI approach can be directly applied to the full system with the inverse model by sampling the output reference, leading to the oMPPI approach. Therefore, the oMPPI

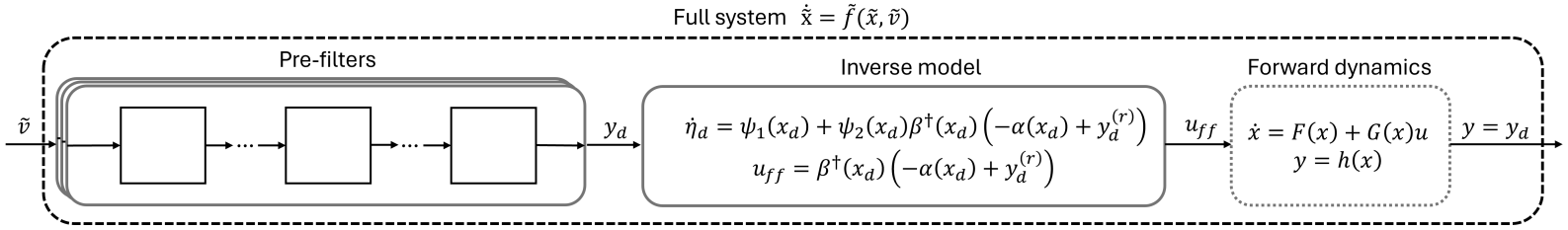


Figure 5.6: The full system $\tilde{f}(\tilde{x}, \tilde{v})$ (dashed) is constructed by applying pre-filters to an arbitrary noisy reference $\tilde{v} = y_m$ and creates a feasible reference trajectory y_d in Eq. (5.46) for the inverse model of Eqs. (5.39)-(5.40), which calculates the inverse input u_{ff} such that the forward dynamics of Eq. (5.23) exactly track the reference $y = y_d$. The oMPPI algorithm simulates the pre-filters (solid box) shown in Fig. 5.5 to generate the feasible reference y_d and the inverse model (solid box) to uncover the corresponding inputs u and states x, and since the output of the forward dynamics (dotted box) is the known feasible reference $y = y_d$. The oMPPI algorithm does not require simulation of the forward dynamics.

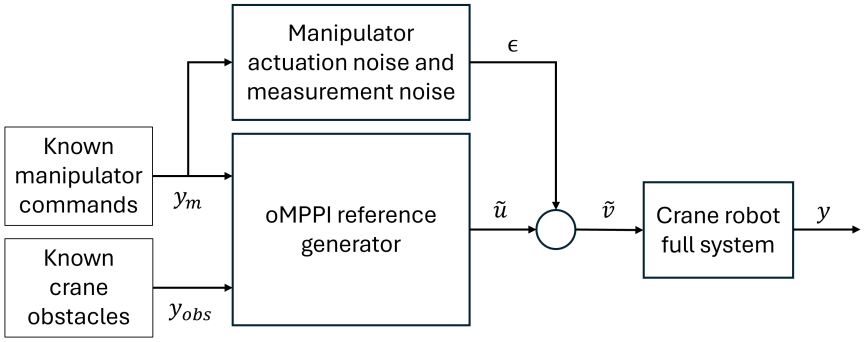


Figure 5.7: The full system in Eq. (5.56) tracks a noisy reference $\tilde{v}$, where noise ϵ is introduced through manipulator actuation (i.e., the manipulator position differs from its reference y_m) and sensing (i.e., the manipulator position differs from its noisy measurement $\tilde{y}_m$). The objective of oMPPI is to control the mean of the noisy references $\tilde{v}$ using known manipulator commands y_m and known crane-robot obstacle boundaries y_{obs} under the presence of noise ϵ.

approach, with output sampling and inversion, retains the convergence properties of the MPPI approach with input sampling, as stated formally, below.

Let the full system dynamics be given by Eq. (5.56). Similar to MPPI, the input sequence to the full system in Eq. (5.56) is the noisy output reference sequence over a receding horizon

$$\tilde{V} = \{\tilde{v}_0, \tilde{v}_1, \ldots, \tilde{v}_{T-1}\} \in \mathbb{R}^{\rho \times T} \tag{5.57}$$

and we have direct control over its mean,

$$\tilde{U} = \{\tilde{u}_0, \tilde{u}_1, \ldots, \tilde{u}_{T-1}\} \in \mathbb{R}^{\rho \times T}, \tag{5.58}$$

which can be selected, e.g., based on the measurements of the manipulator commands y_m and known crane obstacles. The sequence of noisy reference outputs is governed by the probability distribution $\mathbb{Q}$ with a probability density function of

$$q\left(\tilde{V}|\tilde{U}, \Sigma\right) = \prod_{k=0}^{T-1} D^{-1} \exp\left(-\frac{1}{2}\left(\tilde{v}_t - \tilde{u}_t\right)^{\mathrm{T}} \Sigma^{-1} \left(\tilde{v}_t - \tilde{u}_t\right)\right), \tag{5.59}$$

where $D = \left((2\pi)^{\rho}|\Sigma|\right)^{\frac{1}{2}}$ and $\exp(\cdot)$ denotes the exponential function. The discrete time

optimal control problem is defined by

$$\tilde{U}^* = \underset{\tilde{U}\in\tilde{\mathcal{U}}}{\arg\min}\, \mathbb{E}_{\mathbb{Q}_{\tilde{U},\Sigma}} \left[\phi(\tilde{x}_T) + \sum_{t=0}^{T-1} \mathcal{L}\left(\tilde{x}_t, \tilde{u}_t\right) \right], \tag{5.60}$$

where $\phi(\tilde{x}_T)$ is the terminal cost and $\mathcal{L}(\tilde{x}_t, \tilde{u}_t)$ is the running cost function in the form of

$$\mathcal{L}(\tilde{x}_t, \tilde{u}_t) = c(\tilde{x}_t) + \frac{\lambda}{2}\left(\tilde{u}_t^T \Sigma^{-1} \tilde{u}_t + \kappa^T u_t\right), \tag{5.61}$$

where κ enables the reference-input mean to be shifted from the origin, e.g., to a nominal reference $\bar{u}$.

The preceding formulation of solving for the optimal reference sequence $\tilde{U}^*$ for tracking has the same form of the MPPI problem and can therefore leverage its sample-based algorithm. While MPPI samples input sequences U^i and simulates the forward model to find corresponding state X^i and output Y^i sequences, oMPPI samples output sequences Y^i and simulates the inverse model to find corresponding state X^i and input U^i sequences, finding a rollout triplet similar to Eq. (5.16)

$$(U^i, X^i, \tilde{V}^i). \tag{5.62}$$

oMPPI achieves this by creating feasible trajectories y_d with pre-filters and uses the inverse model to uncover the corresponding inverse input $u = u_{ff}$ and state x_d. Algorithm 3 summarizes the oMPPI algorithm, highlighting differences from MPPI in blue.

Theorem 1. *The solution $\tilde{U}^*$ to the optimal control problem in Eq.* (5.60) *can be approximated with the oMPPI algorithm, converging to the optimal sequence as the number of rollouts $N_r \to \infty$.*

Proof. The proof follows directly from [8], since the approach is the same as MPPI for the full system $\tilde{f}$ in Eq. (5.56) with the revised input $\tilde{u}$ — the proof is outlined below. The free energy of a control system is

$$\mathcal{F}(\tilde{V}) = -\lambda \log\left(\mathbb{E}_{\mathbb{P}}\left[\exp\left(-\frac{1}{\lambda} S(\tilde{V})\right)\right]\right), \tag{5.63}$$

Algorithm 3 oMPPI Algorithm (difference from MPPI Algorithm in blue)

Given: Inverse model Eqs. (5.39)-(5.40), number of rollouts N_r, number of timesteps T, initial reference sequence $\{\tilde{u}_0, \ldots, \tilde{u}_{T-1}\}$, covariance Σ, state dependent cost function $c(\cdot)$, terminal cost function $\phi(\cdot)$, control weight γ, inverse temperature λ, filter gain α
while task not completed **do**
    Get filter states z_0
    for $i = 1$ to N_r **do**
        $z_0^i \leftarrow z_0$
        Sample $\mathcal{E}^i = \{\epsilon_0^i, \ldots, \epsilon_{T-1}^i\}$, $\epsilon^i \sim \mathcal{N}(0, \Sigma)$
        for $t = 0$ to $T - 1$ **do**
            Apply noise $\tilde{v}_t^i = \tilde{u}_t^i + \epsilon_t^i$
            Simulate pre-filters $y_d \leftarrow$ Eq. (5.46)
            Simulate inverse $u_t^i, x_t^i \leftarrow$ Eqs. (5.39)-(5.40)
            Construct full system state $\tilde{x}_t \leftarrow u_t, x_t, z_t$
            Accumulate running costs (Eq. (5.61))
                $S_i(\tilde{V}^i)$ += $c(\tilde{x}_t^i) + \gamma \tilde{u}_t^{iT} \Sigma^{-1} \epsilon_t^i$
        end for
        Terminal cost $S_i(\tilde{V}^i)$ += $\phi(\tilde{x}_T^i)$ (Eq. (5.10))
    end for
    $\delta = \min_i(S_i)$
    Compute weights $w(S_1, \ldots, S_{N_r})$ (Eq. (5.72))
    for $t = 0$ to $T - 1$ **do**
        Update control u_t += $\sum_{i=1}^{N_r} w_i \epsilon_t^i$
    end for
    for $t = 1$ to $T - 1$ **do**
        Shift references $\tilde{u}_{t-1} \leftarrow \tilde{u}_t$
    end for
    $\tilde{u}_{T-1} \leftarrow 0$
end while

where $\lambda > 0$ is the inverse temperature parameter, $S(\tilde{V})$ is the state cost of an output reference sequence $\tilde{V}$, $\mathbb{P}$ is a base distribution with a corresponding probability density function $p(V)$, and $\mathbb{E}_{\mathbb{P}}[X]$ represents the expected value of a random variable X under the probability distribution $\mathbb{P}$. The free energy can be shown to satisfy

$$\mathcal{F}(\tilde{V}) \leq \mathbb{E}_{\mathbb{Q}_{\tilde{U},\Sigma}}\left[S(\tilde{V})\right] + \lambda \mathbb{D}_{KL}(\mathbb{Q}_{\tilde{U},\Sigma}||\mathbb{P}), \tag{5.64}$$

where $\mathbb{D}_{KL}$ is the Kullback–Leibler (KL) divergence measuring distance between probability distributions. With a base distribution given by

$$p(V) = q(V|\bar{U}, \Sigma), \tag{5.65}$$

the KL divergence represents the cost penalizing large output variations

$$\mathbb{D}_{KL}(\mathbb{Q}_{\tilde{U},\Sigma}||\mathbb{Q}_{\bar{U},\Sigma}) = \sum_{t=0}^{T-1} \frac{1}{2}(\tilde{u}_t - \bar{u}_t)^T \Sigma^{-1}(\tilde{u}_t - \bar{u}_t). \tag{5.66}$$

Therefore, Eq. (5.64) lower bounds the optimal control problem as

$$\mathcal{F}(\tilde{V}) \leq \mathbb{E}_{\mathbb{Q}_{\tilde{U},\Sigma}}\left[\phi(\tilde{x}_T) + \sum_{t=0}^{T-1} \mathcal{L}\left(\tilde{x}_t, \tilde{u}_t\right)\right]. \tag{5.67}$$

It can be shown that the lower bound of the free energy is achieved by sampling an optimal distribution $\mathbb{Q}^*$ with density function

$$\begin{aligned} q^*(\tilde{V}) &= \frac{1}{\eta} \exp\left(-\frac{1}{\lambda} S(\tilde{V})\right) p(\tilde{V}) \\ \eta &= \int \exp\left(-\frac{1}{\lambda} S(\tilde{V})\right) p(\tilde{V})\, d\tilde{V}, \end{aligned} \tag{5.68}$$

and the optimal output reference at time t is

$$\tilde{u}^* = \int q^*(\tilde{V}) \tilde{v}_t \, d\tilde{V} = \mathbb{E}_{\mathbb{Q}^*}[\tilde{v}_t]. \tag{5.69}$$

However, samples cannot be drawn directly from the optimal distribution, so importance sampling is applied using a distribution with an initialized estimate of the output reference, $\hat{U}$,

$$\mathbb{E}_{\mathbb{Q}^*}[\tilde{v}_t] = \int \frac{q^*(\tilde{V})}{q(\tilde{V}|\hat{U})} q(\tilde{V}|\hat{U}) \tilde{v}_t \, d\tilde{V}, \tag{5.70}$$

where

$$w(\tilde{V}) = \frac{q^*(\tilde{V})}{q(\tilde{V}|\hat{U})} \tag{5.71}$$

defines importance sampling weights, which simplify to

$$w(\tilde{V}) = \frac{1}{\eta} \exp\left(-\frac{1}{\lambda}\left(S(\tilde{V}) + \lambda \sum_{t=0}^{T-1} (\hat{u}_t - \bar{u}_t)^T \Sigma^{-1} \tilde{v}_t\right)\right) \tag{5.72}$$

and the optimal references are

$$\tilde{u}_t^* = \mathbb{E}_{\mathbb{Q}_{\hat{U},\Sigma}}[w(\tilde{V})\tilde{v}_t]. \tag{5.73}$$

As in standard MPPI, the optimal references of Eq. (5.73) use a Monte Carlo approximation of N_r reference sequences to reach a local optimum

$$\tilde{u}_t^* = \mathbb{E}_{\mathbb{Q}^*}[\tilde{v}_t] \approx \frac{\sum_{i=1}^{N_r} w(\tilde{V}_i)\, \tilde{v}_{t,i}}{\sum_{i=1}^{N_r} w(\tilde{V}_i)}, \tag{5.74}$$

and the global optimum is achieved as $N_r \to \infty$.

□

Remark 10. *In the oMPPI framework, the feedforward control input $u = u_{ff}$ is included as part of the full system state $\tilde{x}_t$ and, therefore, a weighting to avoid large inputs can be directly included e.g., with standard quadratic cost terms such as $u_{ff}^T \Sigma^{-1} u_{ff}$ in the state cost function $c(\tilde{x}_t)$. Additionally, if needed, saturation constraints on the system input u can be included by simulating the full system through the forward model in Fig. 5.6 as in standard MPPI.*

5.3 Results and Discussion

5.3.1 Inverse of crane-robot dynamics

The inverse of the crane-robot dynamics in Eq. (5.2) is found after it is first stabilized about the origin using a state feedback of the form [15]

$$\ddot{x}_c = -Kx + \ddot{x}_{c,r} = \begin{bmatrix} k_1 & k_2 & k_3 & 0 \end{bmatrix} x + \ddot{x}_{c,r}, \tag{5.75}$$

which stabilizes the dynamics in Eq. (5.2) if $k_1 > 0$, $k_2 > 0$, and $k_3 < 0$, where $\ddot{x}_{c,r}$ is the input, i.e., the reference acceleration of the crane robot. The resulting closed-loop dynamics

$$\begin{aligned}\dot{x} &= F_{fb}(x) + G(x)u \\ &= \begin{bmatrix} \dot{x}_c \\ -k_1 x_c - k_2 \dot{x}_c - k_3 \theta \\ \dot{\theta} \\ \frac{k_1}{l} x_c + \frac{k_2}{l} \dot{x}_c + \frac{k_3 - g}{l} \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ \frac{1}{l} \end{bmatrix} \ddot{x}_{c,r},\end{aligned} \tag{5.76}$$

is inverted to find the reference acceleration $\ddot{x}_{c,r}$ to track the output y_p. Specifically, the system has relative degree $r = 4$, with ξ representing the output y_p and its $r - 1 = 3$ time derivatives

$$\xi = \left[y_p, y_p^{(1)}, y_p^{(2)}, y_p^{(3)}\right]^T \in \mathbb{R}^4, \tag{5.77}$$

$$y_p = x_c + l\theta, \tag{5.78}$$

with tracking-normal form in Eqs. (5.24)-(5.26)

$$y_p^{(1)} = \dot{\xi}_1 = \xi_2 = \dot{x}_c + l\dot{\theta}, \tag{5.79}$$

$$y_p^{(2)} = \dot{\xi}_2 = \xi_3 = \ddot{x}_c + l\ddot{\theta} = -g\theta, \tag{5.80}$$

$$y_p^{(3)} = \dot{\xi}_3 = \xi_4 = -g\dot{\theta}, \tag{5.81}$$

$$y_p^{(4)} = \alpha(x) + \beta(x)u \tag{5.82}$$

$$= L_{F_{fb}}^4 H(x) + \left(L_G L_{F_{fb}}^3 H(x)\right) u \tag{5.83}$$

$$= -g\left(\frac{k_1}{l} x_c + \frac{k_2}{l} \dot{x}_c + \frac{k_3 - g}{l} \theta\right) - \frac{g}{l} \ddot{x}_{c,r}. \tag{5.84}$$

Since $L_g L_f^3 H(x) \neq 0$, the relative degree of the system is $r = 4$, which matches the number of states, and therefore, there are no internal dynamics. The inverse system is given by Eq. (5.40) as

$$\ddot{x}_{c,ff} = \frac{l}{g} y_d^{(4)} + k_1 x_{c,d} + k_2 \dot{x}_{c,d} + (k_3 - g)\theta_d. \tag{5.85}$$

Furthermore, the system is differential flat [44], as all states and inputs can be written in terms of a desired output and its time derivatives as

$$x_{c,d} = y_d + \frac{l}{g}\ddot{y}_d, \tag{5.86}$$

$$\dot{x}_{c,d} = \dot{y}_d + \frac{l}{g}y_d^{(3)}, \tag{5.87}$$

$$\theta_d = -\frac{\ddot{y}_d}{g}, \tag{5.88}$$

$$\dot{\theta}_d = -\frac{y_d^{(3)}}{g}. \tag{5.89}$$

Thus, the feedforward input $\ddot{x}_{c,ff}$ for the crane robot can be written in terms of the desired output by substituting Eqs. (5.86)-(5.89) into Eq. (5.85)

$$\begin{aligned}\ddot{x}_{c,ff} = \frac{l}{g}y_d^{(4)} + k_1\left(y_d + \frac{l}{g}\ddot{y}_d\right) + k_2\left(\dot{y}_d + \frac{l}{g}y_d^{(3)}\right) \\ + \left(1 - \frac{k_3}{g}\right)\ddot{y}_d.\end{aligned} \tag{5.90}$$

Gains were experimentally selected following methods in [15] as $k_1 = 25\ \mathrm{s}^{-2}$, $k_2 = 15\ \mathrm{s}^{-1}$, and $k_3 = -5\ \frac{\mathrm{m}}{\mathrm{s}^2}$.

5.3.2 Crane robot pre-filters

Pre-filters precede the inverse model to create feasible trajectories. While $r = 4$ pre-filters are required for tracking, $r + 1 = 5$ pre-filters were implemented for the crane-robot for additional smoothness

$$y_d = \left(\frac{\alpha}{s + \alpha}\right)^5 \tilde{y}_m, \tag{5.91}$$

and filter cutoff frequency was experimentally selected to 1.0 Hz ($\alpha = 6.3\ \frac{\mathrm{rad}}{\mathrm{s}}$) by incrementally decreasing the cutoff frequency until vibrations were not present in the control.

5.3.3 oMPPI design

State-dependent costs

oMPPI achieves trajectory tracking and collision avoidance through its state-dependent cost terms of

$$\tilde{c}(\tilde{x}_t) = c_c(\tilde{x}_t) + c_{tr}(\tilde{x}_t) + c_s(\tilde{x}_t), \tag{5.92}$$

where $c_c(\cdot)$ is the collision cost in Eq. (5.12), $c_{tr}(\cdot)$ is the tracking cost in Eq. 5.13, and $c_s(\cdot)$ is a saturation cost of

$$c_s(x_t) = \begin{cases} w_s, & \text{if abs}(\dot{x}_c) > v_{max} \text{ or } (\ddot{x}_c) > a_{max} \\ 0, & \text{otherwise} \end{cases}, \tag{5.93}$$

with a weight of $w_s = 0.001$, where each saturated timestep is penalized equivalent to a tracking error of 0.03 m, enabling saturation penalty without simulating the forward dynamics.

Output-informed mean initialization

Since samples are drawn in the output space, the mean can directly encode estimates of the optimal solution to guide sampling. The crane-robot's optimal responses seek to track the manipulator commands y_m when there are no obstacles and need to stop from hitting obstacles. Therefore, the mean is initialized at each timestep t as

$$\tilde{u}_t = \begin{cases} y_{obs,r}, & \text{if } y_{m,t} < y_{obs,r} \\ y_{obs,l} & \text{if } y_{m,t} < y_{obs,l} \\ \tilde{y}_{m,t} & \text{otherwise} \end{cases}, \tag{5.94}$$

where the reference sequence is initialized as the commands $y_{p,t} = y_{m,t}$ unless it is close to an obstacle, i.e., for a left obstacle boundary $y_{m,t} < y_{obs,l}$ or for a right obstacle boundary $y_{m,t} > y_{obs,r}$.

5.3.4 Comparative Evaluation of Autonomous Experiments

oMPPI improves tracking with smaller preview time

As shown experimentally below, oMPPI improves tracking even with smaller preview time T_p. Similar to experiments from Section 5.1.4 for MPPI, the crane-robot output y_p aims to track the manipulator trajectory $y_m(\tau)$ with a delay as in Eq. (5.1). As seen in Fig. 5.8(a), oMPPI tracking error e decreases from $e = 0.16$ m for a preview time of $T_p = 2$ s to tracking error $e = 0.063$ m for a preview time of $T_p = 0.25$ s. Overall, the oMPPI tracking error e_{p,T_p} of the delayed manipulator trajectory remains nearly constant with preview time T_p as shown in Fig. 5.8(b), and ranges from values of 0.042 m to 0.046 m. This near-constant error is also illustrated in Fig. 5.8(b) by plotting output $y_p(\tau)$ responses taken at different preview times, $T_p = 1.5$ s and $T_p = 0.5$ s. This remaining (near-constant) error e_{p,T_p} corresponds to the error e_d induced by the pre-filters

$$e_d = e_{RSME}(y_m, y_d, 0) \tag{5.95}$$

of $e_d = 0.049$ m, verifying that the inverse model enables precision output y_p tracking of the filtered desired output y_d.

Some preview time is needed for collision avoidance

Providing adequate preview time T_p enables collision avoidance since the crane robot cannot instantaneously stop. Collision avoidance is evaluated by adding an obstacle boundary to the experiments from Section 5.3.4 at $y_{obs,r} = 0.5$ m, as depicted in Fig. 5.9(a). This scenario represents the case where the manipulator passes underneath an in-wing system (e.g., a pipeline), typically near the confined space walls. When the preview time T_p was decreased to $T_p = 0.25$ s all seven trajectories violated the obstacle boundary, while larger preview times of $T_p = 1.5$ s, $T_p = 0.5$ s, enables collision avoidance with oMPPI as shown in Fig. 5.9(b).

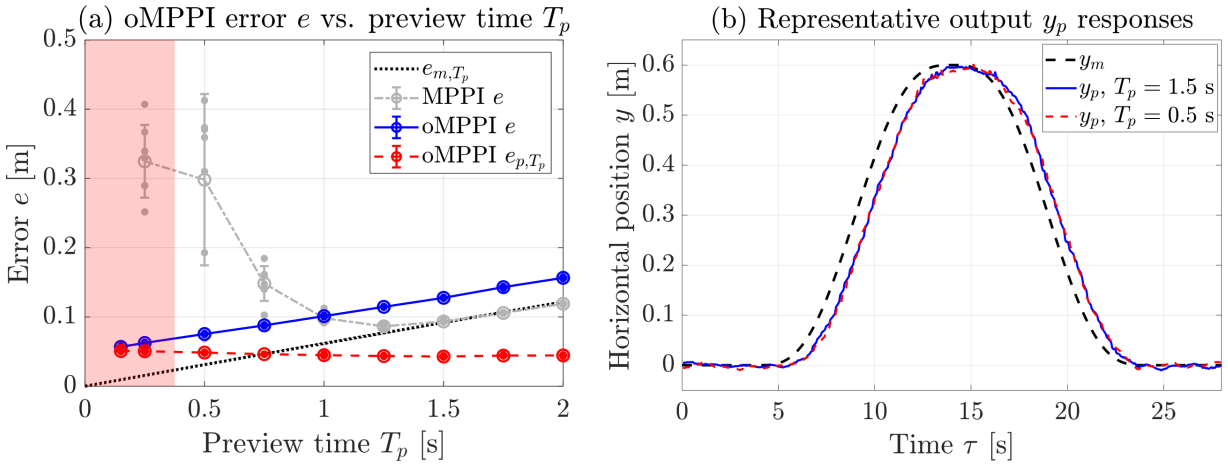


Figure 5.8: oMPPI experimental responses tracking the fixed trajectory with noise $\tilde{y}_m(\tau)$ in Fig. 5.2. (a) Tracking error e in Eq. (5.19) with oMPPI (blue, solid) decreases as preview time T_p decreases, as opposed to increasing tracking error e observed with MPPI (grey, thin dashed). Error e_{p,T_p} to the delayed trajectory is nearly-constant, corresponding to delay error e_d in Eq. (5.95) from the pre-filters. (b) Comparison of time-shifted output responses $y_p(\tau + T_p)$ with the fixed manipulator trajectory $y_m(\tau)$ (black, thick dashed) show that decreasing the preview time with oMPPI from $T_p = 1.5$ s (blue, solid) to $T_p = 0.5$ s (red, dashed) does not significantly change the error in the response. Preview time cannot be made arbitrarily small (e.g., $T_p = 0.25$ s), since preview time enables collision avoidance through sampling in cases with obstacles as indicated by the red highlighted region.

Performance comparison

Overall, oMPPI achieves precision tracking and reduces oscillations without the need to increase the preview time T_p. Fig. 5.8(a) shows that MPPI achieves its best tracking error of $e = 0.087$ m at preview time $T_p = 1.25$ s while oMPPI achieves its best tracking error without collision of $e = 0.075$ m at a smaller preview time of $T_p = 0.5$ s. Therefore, oMPPI achieves better tracking (with 13% reduction tracking error e) while substantially reducing the preview time needed by a factor of 2.5 with a lower (13% reduction) tracking error for the sample trajectory $y_m(\tau)$. Moreover, oMPPI reduces oscillations in the output in Fig. 5.8(b), caused by the pendulum dynamics, when compared to the MPPI responses in Fig. 5.3(b).

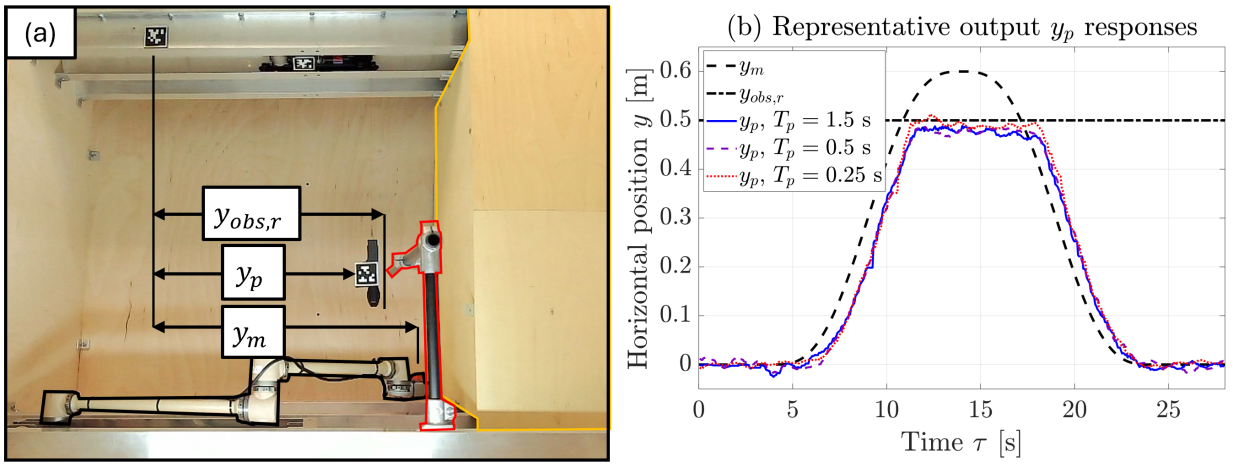


Figure 5.9: The experimental crane robot tracks a fixed trajectory representing the manipulator position with an obstacle boundary at $y_{obs,r}$. (a) The crane robot aims to track the manipulator until the obstacle boundary is reached. At the obstacle boundary, oMPPI samples seek to avoid collisions. (b) Comparison of time-shifted output responses $y_p(\tau + T_p)$ with the fixed manipulator trajectory $y_m(\tau)$ (black, thick dashed) and obstacle boundary at $y_{obs,r}$ (black, dot-dashed) show sample trajectories avoiding collision at $T_p = 1.5$ s (blue, solid) and $T_p = 0.5$ s (purple, thin dash) and colliding with the obstacle boundary at $T_p = 0.25$ s (red, dotted), demonstrating that some amount of preview time T_p is required for collision avoidance.

5.3.5 Comparative Evaluation of Teleoperated Experiments

Teleoperation experiments verify the effectiveness of the active vision crane-robot for teleoperated manipulator navigation and show improvements in user experience with oMPPI compared to MPPI when eight participants perform a fastener-inspection task within an experimental confined space.

Fastener-inspection task

The fastener inspection task requires participants to navigate the teleoperated manipulator across the confined space to inspect whether fasteners are properly seated (i.e., no gap) at two brackets located between obstacles, as depicted in in Fig. 5.10. Participants relied on the active vision (crane-robot) camera view to visualize obstacles and navigate the manip-

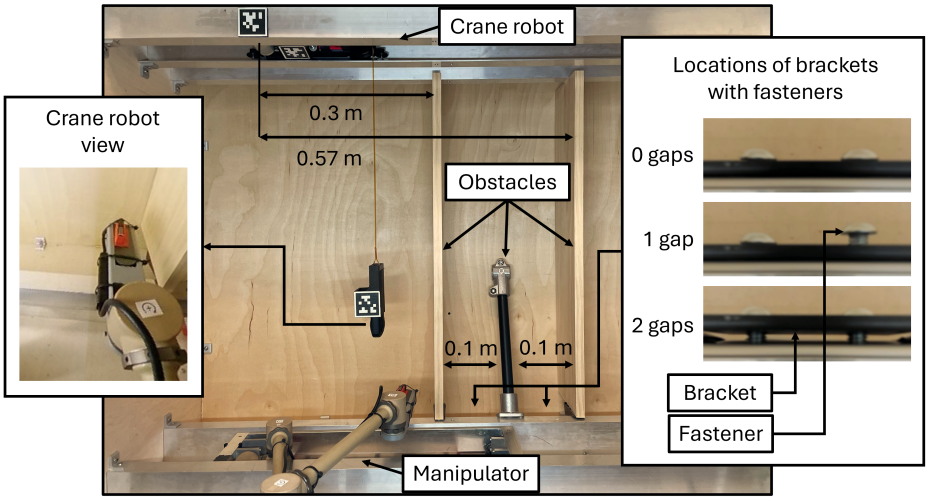


Figure 5.10: The confined space for the teleoperation experiments. Participants navigated across the confined space to inspect whether fasteners, located between obstacles, were properly installed or had gaps using the teleoperated manipulator camera – example fastener patterns are shown in the right inset. Participants relied on the view from the active vision (crane-robot) camera to move around obstacles – an example view is shown in the left inset.

ulator with a joystick around the obstacles to avoid collision. The joystick input by the operator specified manipulator camera velocities, and inverse kinematics was used to control the manipulator arm motion with a maximum speed of 0.2 $\frac{\text{m}}{\text{s}}$ while the camera at the end of the manipulator maintained a fixed orientation relative to the confined space. Participants inspected and recorded whether each fastener had a visible gap using the downward-facing monocular camera at the end of the manipulator. After inspecting all the fasteners, participants pressed a button to confirm task completion, and then answered a seven point Likert scale questionnaire (Q1-Q5, Fig. 5.11). Each participant completed the task four times, twice with MPPI and twice with oMPPI, where a psuedo-random number generator randomized the initial controller (removing sequence bias) and randomized fastener configurations. The participants did not have apriori knowledge of which controller they were using. The crane robot was controlled to autonomously follow the teleoperated manipulator using MPPI with the estimated minimal-tracking-error preview time of 1.25 s and oMPPI with a preview time

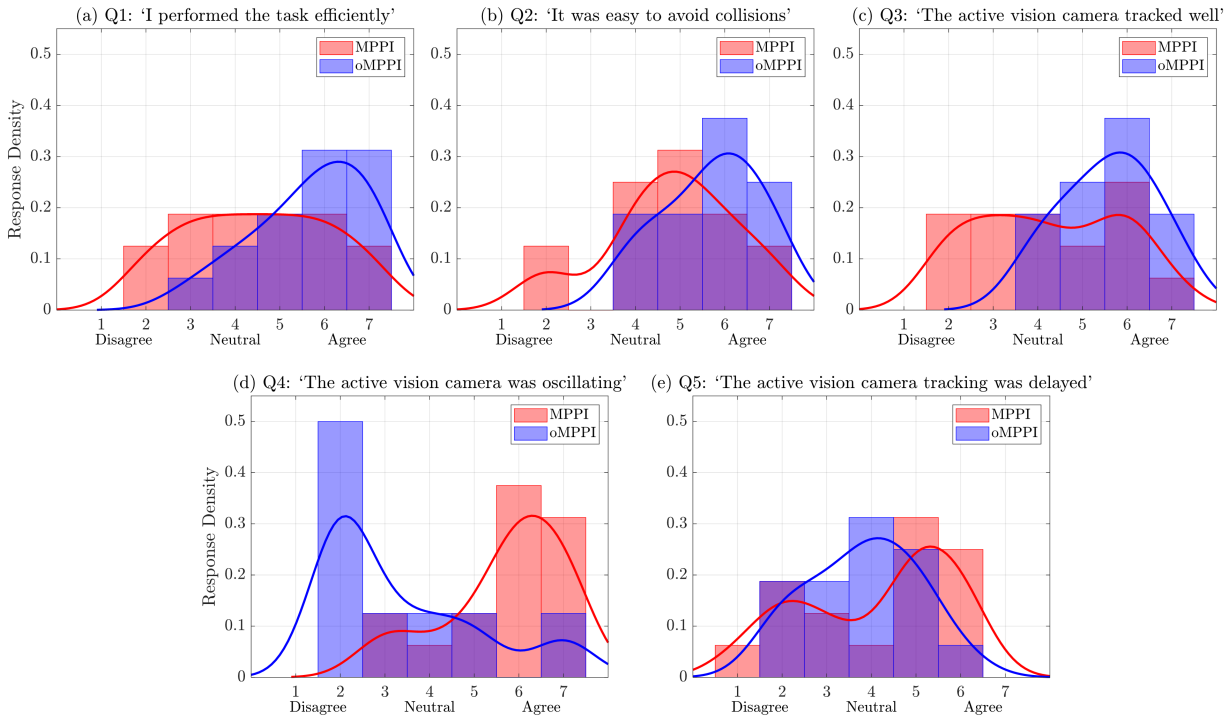


Figure 5.11: Results from 32 teleoperated fastener inspection trials. (a)-(e) Histogram of response density from subjective questionnaire on a 7-point Likert scale (1: Strongly disagree; 4: Neutral; 7: Strongly agree) comparing participant perceptions after completing the fastener inspection task using MPPI and oMPPI. Trends are clarified with smoothed response densities (solid lines) using a squared exponential kernel with a bandwidth of 0.7.

of 0.5 s to minimize tracking error e, as discussed earlier.

Active vision system facilitates inspection task

The teleoperated experiments show that the active vision crane-robot enables confined space navigation and inspection with the teleoperated manipulator. With both MPPI and oMPPI controllers, tasks were successfully completed with mean completion times of 75 s and 67 s, respectively, as seen in Table 5.1. Although all participants successfully avoided manipulator collisions with obstacles while navigating to the inspection sites with both MPPI and oMPPI, the task was perceived as being completed more efficiently with oMPPI compared with

Table 5.1: Experimental results from 32 teleoperated fastener inspection trials

Metric	MPPI Mean (SD)	oMPPI Mean (SD)
Completion time [s]	75.2 (35.3)	67.4 (37.9)
Tracking error e [mm]	30.6 (4.3)	23.7 (6.0)
Angle θ RMS [deg]	1.3 (0.3)	0.5 (0.1)

MPPI, as seen through Q1 in Fig. 5.11(a). Additionally, collision avoidance was perceived as being easier with oMPPI compared with MPPI, as seen through Q2 in Fig. 5.11(b). Both perceptions are supported by a Wilcoxon signed rank test with statistical significance, rejecting the null hypothesis ($p < 0.05$).

Quantitative Tracking and Oscillation Comparisons

oMPPI reduced tracking error e and oscillation with a lower preview time compared with MPPI. Mean tracking error e across all teleoperated trajectories for oMPPI was 23.7 mm, a 22% improvement compared with mean tracking error e of 30.6 mm for MPPI. Participants generated overall slower teleoperation trajectories with significant resting periods compared with the fixed trajectory in Fig. 5.2, and therefore, errors due to delay are less pronounced. However, participant teleoperation trajectories typically include stop-and-go-type movements (see example trajectories in Fig. 5.12), resulting in an increase of oscillations that lead to an increase in the overall tracking error. The observed oscillations are at the crane-robot's natural frequency (of the pendulum-type camera payload), as seen in the mean power spectral density (PSD) of all teleoperated trajectories shown in Fig. 5.13(a). The PSD contains a peak of magnitude -23.5 dB/Hz at 0.7 Hz for trajectories with MPPI, corresponding to the natural frequency of the pendulum payload. oMPPI achieves better tracking and effectively removes these oscillations, with a peak magnitude of -43.8 dB/Hz at 0.7 Hz. Furthermore, the root mean square (RMS) of the angle measurements θ for each participant

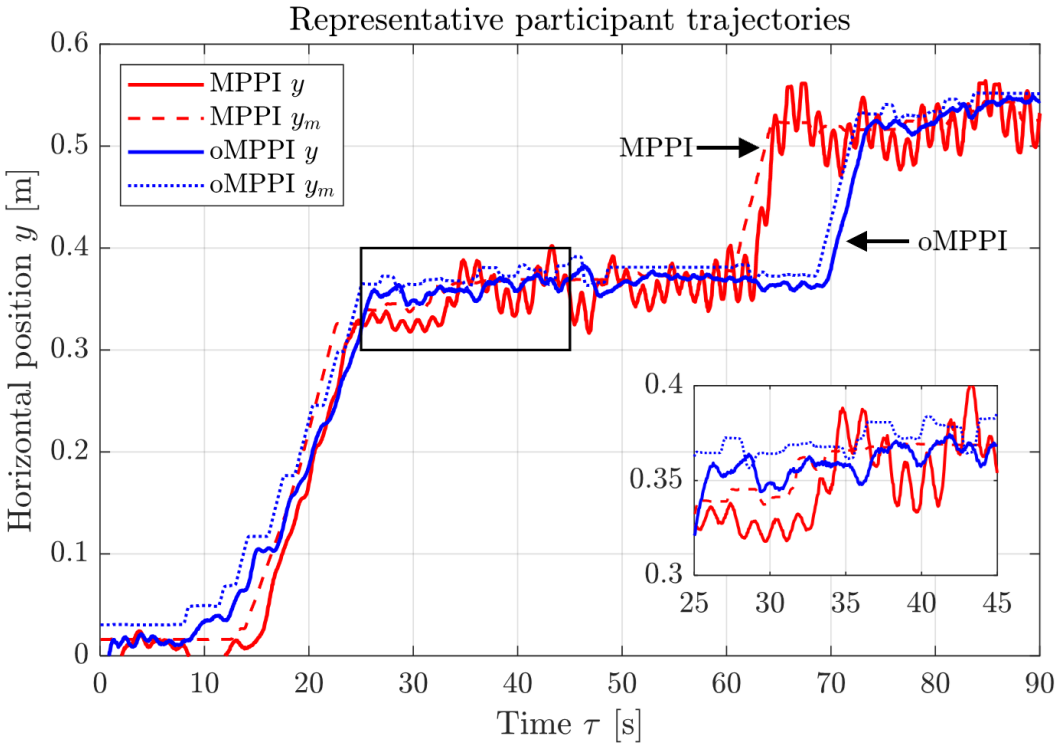


Figure 5.12: Comparison of an example participant's trajectories with MPPI and oMPPI. Participant references y_m typically include stop-and-go-type movements that induce oscillations in the output y response. MPPI (red, dashed reference y_m) does not fully remove these oscillations, while oMPPI (blue, dotted reference y_m) reduces these oscillations leading to improved tracking.

trajectory is shown in Fig. 5.13(b), and the mean RMS value across all MPPI trajectories was 0.023 rad (1.3 deg), compared to 0.008 rad (0.5 deg) for oMPPI, as reported in Table 5.1. Thus, oMPPI reduced the mean angle θ RMS by 65% across all trials. These results show that oMPPI can effectively reduce oscillations and precisely track trajectories by sampling the output directly, whereas MPPI relies on exploration through the input space for tracking, leading to sustained oscillation with increased tracking errors.

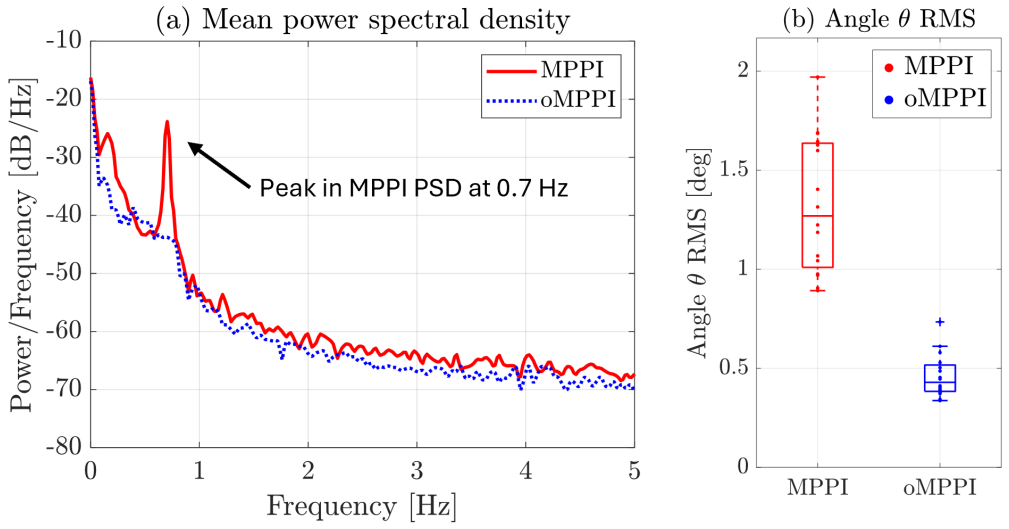


Figure 5.13: Comparison of oscillations in MPPI and oMPPI. (a) Mean power spectral density of all teleoperated trajectories for MPPI (red, solid) and oMPPI (blue, dotted) show that MPPI excites the natural frequency of the payload swing dynamics at 0.7 Hz, whereas oMPPI corrects for the swing dynamics and reduces these oscillations. (b) RMS values of the angle θ, over all participant trajectories, show that oMPPI substantially reduces the RMS value compared with MPPI.

Qualitative Tracking and Oscillation Comparisons

Participants perceived reduced tracking errors and oscillations when using oMPPI compared to MPPI, supporting the preceding quantitative findings. Specifically, responses to Q3 in Fig. 5.11(c) indicate improved active vision camera tracking with oMPPI, while responses to Q4 show reduced oscillations with oMPPI. Both perceptions were validated by a Wilcoxon signed rank test, which showed statistical significance, rejecting the null hypothesis ($p < 0.05$). Participant camera-tracking perception is likely attributable to oscillation reduction discussed in Section 5.3.5 rather than the 0.75 s preview time delay since Q5 in Fig. 5.11(e) shows no statistically significant difference in perceived delay between MPPI and oMPPI, as indicated by a Wilcoxon signed rank test ($p > 0.05$).

While the teleoperated experiments demonstrated the effectiveness of the crane-robot

active vision system with oMPPI for a confined space fastener inspection task, additional work is needed to investigate the longer-term impacts of improved tracking and reduced oscillations of the active vision system on human operations, including fatigue and learning, which were not studied here. The results of this study can be system-dependent, and improvements with oMPPI may differ for various tasks and across different teleoperated robotic systems. For example, the manipulator joints had noticeable backlash, which may contribute to the observed stop-and-go positioning near obstacles. Similarly, the joystick contained a 10% deadzone on the input, rejecting unintentional commands, which may have influenced control. A future study on the impacts of different input devices may lead to improved teleoperation performance for both MPPI and oMPPI. Furthermore, the results are task-dependent. Future work can explore how tracking delay from preview time influences participant experience in higher-speed applications or dynamic inspection scenarios.

5.4 Conclusion

This article presented an output-sampled model-predictive path integral (oMPPI) control approach that used reference-output-sampling and inversion-based methods for improving the tracking performance of standard input-sampled MPPI. Methods were presented to ensure feasibility of the inversion-based oMPPI approach, and to ensure that optimality of MPPI methods are preserved. The improved performance of oMPPI (reduced tracking error and smaller oscillations with smaller preview time) was demonstrated through comparative experiments. Moreover, teleoperation experiments with oMPPI showed significant reductions in tracking error and oscillations as participants teleoperated a manipulator to inspect fasteners using an active-vision crane robot in a confined space (aircraft in-wing) application.

Chapter 6

CONCLUSIONS

6.1 Summary

The objective of this dissertation was to improve in-wing robotic teleoperation enabling mechanics to perform their work safely from outside of unergonomic and hazardous confined spaces. This was achieved by improving the control of the crane robot presented in Chapter 3. Control was improved through (MC1) direct teleoperation of the crane robot capable of completing inspection tasks in Chapter 4 and (MC2) an output-sampled model predictive path integral control (oMPPI) for the crane robot active vision system, capable of autonomously tracking teleoperated manipulators to provide dynamic confined space views in Chapter 5.

6.1.1 Semi-autonomous control using differential flatness

The semi-autonomous control in Chapter 4 leveraged the differential flatness property of the crane robot to generate smooth trajectories from operator inputs with corrections to assist collision avoidance, thereby reducing undesired oscillations, mitigating collisions, and improving task efficiency compared with direct velocity control. Autonomous experiments demonstrated an 89% reduction in residual oscillations when compensating for swing dynamics using flatness-based control. Simulations demonstrated the robustness of the proposed approach. Teleoperation experiments further validated the method, achieving safer inspections with a 0% collision rate compared to 33% under conventional control and improving task efficiency with an 18.7% reduction in completion time for collision free trials, with improvements perceived by participants. These results confirm that the crane robot, with semi-autonomous control, is a viable solution for in-wing teleoperated inspection tasks.

6.1.2 oMPPI for active vision systems

oMPPI presented in Chapter 5 achieves precision tracking for the control of the crane robot active vision system for tracking teleoperated manipulators using pre-filters to create feasible trajectories for exact tracking with inverse models. oMPPI enables direct sampling from the output space with the ability to initialize sample means based on known output criteria (e.g, tracking and collision avoidance). By reducing the required preview time, delay errors from the optimization horizon and required computation are reduced. Experiments show that preview time cannot be reduced indefinitely, as it relies on sampling to ensure collision avoidance through exploration of the output space. Autonomous experiments verify that oMPPI reduced the necessary preview time by a factor of 2.5 and with an improved tracking accuracy of 13% on the sample trajectory compared with MPPI. Teleoperation experiments show that oMPPI (with inversion) increases tracking precision of a teleoperated manipulator by 22% and reduces camera oscillations by 65% when compared to MPPI without the inversion. The crane robot active vision system, with improved oMPPI control, enables inwing teleoperation of manipulators by providing precise and efficient active vision tracking in confined spaces.

6.2 Future work

6.2.1 Semi-autonomous control teleoperation experiments

While the semi-autonomous control (SC) successfully mitigated collision during teleoperated experiments in Section 4.4.3, additional work is needed to decouple the influence of reduced oscillations from the flatness-based control and the reference point modulation on collision avoidance. Furthermore, some reports of undesired oscillation under the use of SC may be caused by the camera payload's tendency to rotate axially about its belt. Future crane robot designs can incorporate physical constraints to limit axial rotations.

6.2.2 oMPPI teleoperation experiments

While the teleoperated experiments demonstrated the effectiveness of the crane-robot active vision system with oMPPI for a confined space fastener inspection task in Section 5.3.5, additional work is needed to investigate the longer-term impacts of improved tracking and reduced oscillations of the active vision system on human operations, including fatigue and learning, which were not studied here. The results of this study can be system-dependent, and improvements with oMPPI may differ for various tasks and across different teleoperated robotic systems. For example, the manipulator joints had noticeable backlash, which may contribute to the observed stop-and-go positioning near obstacles. Similarly, the joystick contained a 10% deadzone on the input, rejecting unintentional commands, which may have influenced control. A future study on the impacts of different input devices may lead to improved teleoperation performance for both MPPI and oMPPI. Furthermore, the results are task-dependent. Future work can explore how tracking delay from preview time influences participant experience in higher-speed applications or dynamic inspection scenarios.

6.2.3 oMPPI pre-filter optimization

Pre-filter gain selection for the crane robot oMPPI was experimentally determined in Section 5.3.2. Future work can investigate optimized filter gain selection. Increasing the oMPPI control timestep could generate smoother rollout trajectories, enabling higher filter cutoff frequencies without introducing vibrations. Additionally, optimizing the sample covariance could further constrain the shaping of resultant trajectories.

6.2.4 Crane robot applications beyond aircraft wings

Crane robot applications extend beyond aircraft manufacturing. Hazardous confined spaces containing structural beams are common in other industries, such as naval ship ballast tanks, which can directly deploy the crane robot [87, 93]. The crane robot can also enter hazardous spaces such as ceilings of nuclear hazard sites, enabling full room inspection [94]. Further-

more, a modified crane robot can simplify bridge inspections by eliminating the need for complex climbing robots [95–97]. By traversing overhead beams, crane robots can enable efficient top-down inspections, making them ideal for large structures. Future research can focus on modified crane robots for extended applications in other hazardous environments.

6.2.5 Semi-autonomous control for cranes

Semi-autonomous crane robot control demonstrated safe operations while preserving human expertise for guidance. This approach extends to crane applications across various industries. Differential flatness properties are observed in other crane systems, e.g. three-dimensional gantry cranes used in factories [98], which can naturally adapt the semi-autonomous control to assist with oscillation reduction and collision avoidance. Additionally, crane accidents are a persistent problem in the construction industry, where power line collision and uncontrolled payload sway are common causes of accidents [99–101]. Both hazards can be directly improved by adopting the proposed semi-autonomous control. Future research efforts include adapting the semi-autonomous control to other industries to create safer operations by accounting for open challenges of sensing and disturbance rejection (e.g., to counteract wind) in unstructured environments.

6.2.6 oMPPI for multi-robot optimization

oMPPI can be extended to multi-agent optimization, enabling collaborative and efficient task execution. In the crane robot active vision system, optimization problems can include manipulator control for collaborative task completion. In multi-agent systems already applying MPPI, reducing preview time with oMPPI not only enhances computational efficiency but also improves responsiveness to collisions, particularly in decentralized settings where less optimization preview is advantageous [102, 103]. Therefore, it is valuable to explore future research extending the applications of oMPPI to multi-agent optimization.

BIBLIOGRAPHY

[1] Manpreet Haur Dhoot, Ip-Shing Fan, and Nico Avdelidis. Requirements for designing a robotic system for aircraft wing fuel tank inspection. In *PHM Society European Conference*, volume 6, pages 12–12, 2021.

[2] Florian Heilemann, Alireza Dadashi, and Kai Wicke. Eeloscope—towards a novel endoscopic system enabling digital aircraft fuel tank maintenance. *Aerospace*, 8(5), 2021.

[3] Maria-Giorgiana Gaina. Dangerous entry into the aircraft fuel tank–introduction of mobil robot. *INCAS Bulletin*, 11(2):97–110, 2019.

[4] Parker Owan, Joseph Garbini, and Santosh Devasia. Addressing agent disagreement in mixed-initiative traded control for confined-space manufacturing. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 227–234. IEEE, 2017.

[5] Ryuya Sato, Mitsuhiro Kamezaki, Satoshi Niuchi, Shigeki Sugano, and Hiroyasu Iwata. Cognitive untunneling multi-view system for teleoperators of heavy machines based on visual momentum and saliency. *Automation in Construction*, 110:103047, 2020.

[6] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. An autonomous dynamic camera method for effective remote teleoperation. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 325–333, 2018.

[7] Emmanuel Senft, Michael Hagenow, Pragathi Praveena, Robert Radwin, Michael Zinn, Michael Gleicher, and Bilge Mutlu. A method for automated drone viewpoints to support remote robot manipulation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7704–7711, 2022.

[8] Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Information-theoretic model predictive control: Theory and applications to autonomous driving. *IEEE Transactions on Robotics*, 34(6):1603–1622, 2018.

[9] Grady Williams, Brian Goldfain, Paul Drews, Kamil Saigol, James M Rehg, and Evangelos A Theodorou. Robust sampling based model predictive control with sparse objective information. In *Robotics: Science and Systems*, volume 14, page 2018, 2018.

[10] Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357, 2017.

[11] Thomas Power and Dmitry Berenson. Learning a generalizable trajectory sampling distribution for model predictive control. *IEEE Transactions on Robotics*, 2024.

[12] Zishen Wan, Ashwin Sanjay Lele, and Arijit Raychowdhury. Circuit and system technologies for energy-efficient edge robotics. In *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 275–280. IEEE, 2022.

[13] Jongmin Jo, Sucheol Jeong, and Pilsung Kang. Benchmarking gpu-accelerated edge devices. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 117–120, 2020.

[14] Bernd Kolar, Hubert Rams, and Kurt Schlacher. Time-optimal flatness based control of a gantry crane. *Control Engineering Practice*, 60:18–27, 2017.

[15] Wade Marquette, Kyle Schultz, Vamsi Jonnalagadda, Benjamin Wong, Joseph Garbini, and Santosh Devasia. Semi-autonomous teleoperation using differential flatness of a crane robot for aircraft in-wing inspection: Extended version. *arXiv:2412.10973*, 2024.

[16] Wade Marquette, Kyle Schultz, Vamsi Jonnalagadda, Benjamin Wong, Joseph Garbini, and Santosh Devasia. Semi-autonomous teleoperation using differential flatness of a crane robot for aircraft in-wing inspection. *IEEE Robotics and Automation Letters*, Accepted, to appear.

[17] Wade Marquette, Kyle Schultz, Leon Yan, Michael Bi, Joseph Garbini, and Santosh Devasia. Output-sampled model predictive path integral control (omppi) for precision tracking with active vision systems. *Submitted to journal*.

[18] Niu Guochen, Wang Li, Gao Qingji, and Hu Dandan. Path-tracking algorithm for aircraft fuel tank inspection robots. *International Journal of Advanced Robotic Systems*, 11(5):82, 2014.

[19] Qing Ji Gao, Wei Juan Wang, and Guo Chen Niu. Design bionic structure and analysis of kinematics for aircraft fuel tank inspection robot. *Applied Mechanics and Materials*, 278:594–598, 2013.

[20] Wilhelm Johan Marais and Ali Haydar Göktogan. Design and control of cram: A highly articulated cable-driven remote access manipulator for confined space inspection. In *Proc. Australas. Conf. Robot. Autom.(ACRA)*, pages 1–9, 2017.

[21] Guochen Niu, Zunchao Zheng, Qingji Gao, Weijuan Wang, and Lei Wang. A novel design of aircraft fuel tank inspection robot. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 11(7):3684–3692, 2013.

[22] Guochen Niu, Jinkun Wang, and Kailu Xu. Model analysis for a continuum aircraft fuel tank inspection robot based on the rzeppa universal joint. *Advances in Mechanical Engineering*, 10(5):1687814018778229, 2018.

[23] Jiefeng Jiang, Jingjing You, and Yunbo Bi. Kinematic modeling and simulation of a new robot for wingbox internal fastening application. *Machines*, 11(7), 2023.

[24] Parker Owan, Joseph Garbini, and Santosh Devasia. Faster confined space manufacturing teleoperation through dynamic autonomy with task dynamics imitation learning. *IEEE Robotics and Automation Letters*, 5(2):2357–2364, 2020.

[25] Rob Buckingham, Vilas Chitrakaran, Rosalind Conkie, Geoff Ferguson, Andrew Graham, Alex Lazell, Mariusz Lichon, Nick Parry, Fred Pollard, Amir Kayani, et al. Snake-arm robots: a new approach to aircraft assembly. Technical report, SAE Technical Paper, 2007.

[26] Jerry D. Chungbin, Shuonan Dong, John W. Fuller, Samuel F. Pedigo, Santosh Devasia, Benjamin Yat-chun Wong, Kyle William Schultz, Wade Marquette, Lucky Singh, Derek Keith Loy, and Joseph L. Garbini. Mechanical avatar assembly and system for use in a confined space in a structure and method of using the same, Sep 2022. United States Patent 0281102.

[27] Manpreet Kaur Dhoot and Ip-Shing Fan. Design and development of a mobile robotic system for aircraft wing fuel tank inspection. *SAE International Journal of Advances and Current Practices in Mobility*, 4(2022-01-0042):1126–1137, 2022.

[28] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. Remote telemanipulation with adapting viewpoints in visually complex environments. *Robotics: Science and Systems XV*, 2019.

[29] Ryotaro Temma, Kazuki Takashima, Kazuyuki Fujita, Koh Sueda, and Yoshifumi Kitamura. Third-person piloting: Increasing situational awareness using a spatially coupled second drone. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 507–519, 2019.

[30] Zichao Ji, Guangming Song, Fei Wang, Yawen Li, and Aiguo Song. Design and control of a snake robot with a gripper for inspection and maintenance in narrow spaces. *IEEE Robotics and Automation Letters*, 8(5):3086–3093, 2023.

[31] David Rollinson and Howie Choset. Pipe network locomotion with a snake robot. *Journal of Field Robotics*, 33(3):322–336, 2016.

[32] Tomonari Yamamoto, Sayako Sakama, and Akiya Kamimura. Pneumatic duplex-chambered inchworm mechanism for narrow pipes driven by only two air supply lines. *IEEE Robotics and Automation Letters*, 5(4):5034–5042, 2020.

[33] Jianlin Wang, Yixiang Wang, Lining Peng, Haixiang Zhang, Hang Gao, Chengjiang Wang, Yuan Gao, Huanliang Luo, and Yongquan Chen. Transformable inspection robot design and implementation for complex pipeline environment. *IEEE Robotics and Automation Letters*, 9(6):5815–5822, 2024.

[34] Atsushi Kakogawa and Shugen Ma. A multi-link in-pipe inspection robot composed of active and passive compliant joints. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6472–6478, 2020.

[35] Motoyasu Tanaka Kazuo Tanaka Hidemasa Sawabe, Mizuki Nakajima and Fumitoshi Matsuno. Control of an articulated wheeled mobile robot in pipes. *Advanced Robotics*, 33(20):1072–1086, 2019.

[36] Atsushi Kakogawa and Shugen Ma. Design of a multilink-articulated wheeled pipeline inspection robot using only passive elastic joints. *Advanced Robotics*, 32(1):37–50, 2018.

[37] Silvere Bonnabel and Xavier Claeys. The industrial control of tower cranes: An operator-in-the-loop approach [applications in control]. *IEEE Control Systems Magazine*, 40(5):27–39, 2020.

[38] Héctor Azpúrua, Adriano Rezende, Guilherme Potje, Gilmar Pereira da Cruz Júnior, Rafael Fernandes, Victor Miranda, Levi Welington de Resende Filho, Jacó Domingues, Filipe Rocha, Frederico Luiz Martins de Sousa, et al. Towards semi-autonomous robotic inspection and mapping in confined spaces with the espeleorobô. *Journal of Intelligent & Robotic Systems*, 101:1–27, 2021.

[39] Matteo Rubagotti, Tasbolat Taunyazov, Bukeikhan Omarali, and Almas Shintemirov. Semi-autonomous robot teleoperation with obstacle avoidance via model predictive control. *IEEE Robotics and Automation Letters*, 4(3):2746–2753, 2019.

[40] Tarunraj Singh and William Singhose. Input shaping/time delay control of maneuvering flexible structures. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, volume 3, pages 1717–1731. IEEE, 2002.

[41] Neil Singer, William Singhose, and Eric Kriikku. An input shaping controller enabling cranes to move without sway. Technical report, Savannah River Site (SRS), Aiken, SC (United States), 1997.

[42] William Singhose, Lisa Porter, Michael Kenison, and Eric Kriikku. Effects of hoisting on the input shaping control of gantry cranes. *Control engineering practice*, 8(10):1159–1165, 2000.

[43] Michel Fliess, Jean Lévine, Philippe Martin, and Pierre Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *International journal of control*, 61(6):1327–1361, 1995.

[44] Bernd Kolar and Kurt Schlacher. Flatness based control of a gantry crane. *IFAC Proceedings Volumes*, 46(23):487–492, 2013.

[45] Johannes Diwold, Bernd Kolar, and Markus Schöberl. Discrete-time flatness-based control of a gantry crane. *Control Engineering Practice*, 119:104980, 2022.

[46] Zian Yu and Wangqiang Niu. Flatness-based backstepping antisway control of underactuated crane systems under wind disturbance. *Electronics*, 12(1):244, 2023.

[47] Jun Zeng, Prasanth Kotaru, Mark W Mueller, and Koushil Sreenath. Differential flatness based path planning with direct collocation on hybrid modes for a quadrotor with a cable-suspended payload. *IEEE Robotics and Automation Letters*, 5(2):3074–3081, 2020.

[48] Christoph Hebisch, Sven Jackisch, Dieter Moormann, and Dirk Abel. Flatness-based model predictive trajectory planning for cooperative landing on ground vehicles. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 1031–1036. IEEE, 2021.

[49] Ezra Tal, Gilhyun Ryou, and Sertac Karaman. Aerobatic trajectory generation for a vtol fixed-wing aircraft using differential flatness. *IEEE Transactions on Robotics*, 2023.

[50] Matthias Thomas, Timothy Werner, and Oliver Sawodny. Online trajectory generation and feedforward control for manually-driven cranes with input constraints. In *2021 IEEE Conference on Control Technology and Applications (CCTA)*, pages 654–659, 2021.

[51] Haibo Yu, Liao Wu, Keyu Wu, and Hongliang Ren. Development of a multi-channel concentric tube robotic system with active vision for transnasal nasopharyngeal carcinoma procedures. *IEEE Robotics and Automation Letters*, 1(2):1172–1178, 2016.

[52] Teppei Okuno Shoichi Maeyama and Keigo Watanabe. View point decision algorithm for an autonomous robot to provide support images in the operability of a teleoperated robot. *SICE Journal of Control, Measurement, and System Integration*, 9(1):33–41, 2016.

[53] Emanuele Venzano, Hugo Pousseur, Alessandro Correa Victorino, and Pedro Castillo Garcia. Motion control for aerial and ground vehicle autonomous platooning. In *2022 IEEE 17th International Conference on Advanced Motion Control (AMC)*, pages 213–218, 2022.

[54] Xuesu Xiao, Jan Dufek, and Robin R Murphy. Autonomous visual assistance for robot operations using a tethered uav. In *Field and Service Robotics: Results of the 12th International Conference*, pages 15–29. Springer, 2021.

[55] Seiga Kiribayashi, Kaede Yakushigawa, and Keiji Nagatani. Design and development of tether-powered multirotor micro unmanned aerial vehicle system for remote-controlled construction machine. In *Field and Service Robotics: Results of the 11th International Conference*, pages 637–648. Springer, 2018.

[56] Saman Rahnamaei and Shahin Sirouspour. Automatic viewpoint planning in teleoperation of a mobile robot. *Journal of Intelligent & Robotic Systems*, 76:443–460, 2014.

[57] Zhaoming Xie, C Karen Liu, and Kris Hauser. Differential dynamic programming with nonlinear constraints. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 695–702. IEEE, 2017.

[58] J. Morimoto, G. Zeglin, and C.G. Atkeson. Minimax differential dynamic programming: application to a biped walking robot. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 2, pages 1927–1932 vol.2, 2003.

[59] Yuanfei Xu and Yong Wang. Online iterative lqr controller for wheeled bipedal robot trajectory tracking. In *2024 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1334–1339, 2024.

[60] Vikash Kumar, Emanuel Todorov, and Sergey Levine. Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 378–383, 2016.

[61] Ihab S Mohamed, Kai Yin, and Lantao Liu. Autonomous navigation of agvs in unknown cluttered environments: log-mppi control strategy. *IEEE Robotics and Automation Letters*, 7(4):10240–10247, 2022.

[62] Isin M Balci, Efstathios Bakolas, Bogdan Vlahov, and Evangelos A Theodorou. Constrained covariance steering based tube-mppi. In *2022 American Control Conference (ACC)*, pages 4197–4202. IEEE, 2022.

[63] Chuyuan Tao, Hunmin Kim, and Naira Hovakimyan. Rrt guided model predictive path integral method. In *2023 American Control Conference (ACC)*, pages 776–781. IEEE, 2023.

[64] Ihab S Mohamed, Guillaume Allibert, and Philippe Martinet. Model predictive path integral control framework for partially observable navigation: A quadrotor case study. In *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 196–203. IEEE, 2020.

[65] Manan Gandhi, Hassan Almubarak, and Evangelos Theodorou. Safe importance sampling in model predictive path integral control. *arXiv preprint arXiv:2303.03441*, 2023.

[66] Raphael Kusumoto, Luigi Palmieri, Markus Spies, Akos Csiszar, and Kai O Arras. Informed information theoretic model predictive control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2047–2053. IEEE, 2019.

[67] Elia Trevisan and Javier Alonso-Mora. Biased-mppi: Informing sampling-based model predictive control by fusing ancillary controllers. *IEEE Robotics and Automation Letters*, 2024.

[68] Manan S Gandhi, Bogdan Vlahov, Jason Gibson, Grady Williams, and Evangelos A Theodorou. Robust model predictive path integral control: Analysis and performance guarantees. *IEEE Robotics and Automation Letters*, 6(2):1423–1430, 2021.

[69] G. M. Clayton, S. Tien, K. K. Leang, Q. Zou, and S. Devasia. A review of feedforward control approaches in nanopositioning for high speed spm. *SME Journal of Dynamic Systems, Measurement and Control*, 131(6):061001, 1–19, 2009.

[70] Leon Liangwu Yan and Santosh Devasia. Output-sampled model predictive path integral control (o-mppi) for increased efficiency. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14279–14285. IEEE, 2024.

[71] S. Devasia, Degang Chen, and B. Paden. Nonlinear inversion-based output tracking. *IEEE Transactions on Automatic Control*, 41(7):930–942, 1996.

[72] Liangwu Yan and Santosh Devasia. What observables are needed for precision data-enabled learning of inverse operators? *Journal of Dynamic Systems, Measurement, and Control*, 146(3):031008, 2024.

[73] Diego Romeres, Mattia Zorzi, Raffaello Camoriano, Silvio Traversaro, and Alessandro Chiuso. Derivative-free online learning of inverse dynamics models. *IEEE Transactions on Control Systems Technology*, 28(3):816–830, 2020.

[74] Franziska Meier, Daniel Kappler, Nathan Ratliff, and Stefan Schaal. Towards robust online inverse dynamics learning. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4034–4039. IEEE, 2016.

[75] Thomas Beckers, Jonas Umlauft, and Sandra Hirche. Stable model-based control with gaussian process regression for robot manipulators. *IFAC-PapersOnLine*, 50(1):3877–3884, 2017.

[76] Yuan Gao, Guangming Song, Songtao Li, Fushuai Zhen, Dabing Chen, and Aiguo Song. Linespyx: A power line inspection robot based on digital radiography. *IEEE Robotics and Automation Letters*, 5(3):4759–4765, 2020.

[77] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE international conference on robotics and automation*, pages 3400–3407. IEEE, 2011.

[78] Yuki Nishimura, Shuki Takahashi, Hiromi Mochiyama, and Tomoyuki Yamaguchi. Automated hammering inspection system with multi-copter type mobile robot for concrete structures. *IEEE Robotics and Automation Letters*, 7(4):9993–10000, 2022.

[79] Wenfu Xu, Panhui Yan, Fengxu Wang, Han Yuan, and Bin Liang. Vision-based simultaneous measurement of manipulator configuration and target pose for an intelligent cable-driven robot. *Mechanical Systems and Signal Processing*, 165:108347, 2022.

[80] Zhefan Yu, Jianping Luo, Han Zhang, Eiji Onchi, and Seung-Hee Lee. Approaches for motion control interface and tele-operated overhead crane handling tasks. *Processes*, 9(12):2148, 2021.

[81] Dengqing Tang, Tianjiang Hu, Lincheng Shen, Zhaowei Ma, and Congyu Pan. Apriltag array-aided extrinsic calibration of camera–laser multi-sensor system. *Robotics and biomimetics*, 3:1–9, 2016.

[82] Norman S. Nise. *Control Systems Engineering*, chapter Routh-Hurwitz Criterion, pages 303–305. John Wiley & Sons, Hoboken, NJ, 7th edition, 2015.

[83] H.K. Khalil. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002.

[84] Xuning Yang, Jasmine Cheng, and Nathan Michael. An intention guided hierarchical framework for trajectory-based teleoperation of mobile robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 482–488, 2021.

[85] Mariah L. Schrum, Michael Johnson, Muyleng Ghuy, and Matthew C. Gombolay. Four years in review: Statistical practices of likert scales in human-robot interaction studies. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '20, page 43–52, New York, NY, USA, 2020. Association for Computing Machinery.

[86] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[87] Benjamin Wong, Wade Marquette, Nikolay Bykov, Tyler M. Paine, and Ashis G. Banerjee. Human-assisted robotic detection of foreign object debris inside confined spaces of marine vessels using probabilistic mapping. *Robotics and Autonomous Systems*, 161:104349, 2023.

[88] Ayhan Aktas, Koert Bruggeman, Hakan Yazici, and Mert Sever. Anti-sway control of a gantry crane with lmi based robust pole placement: Experimental verification for acceleration control approach. In *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, pages 1–6. IEEE, 2018.

[89] Ji Yin, Zhiyuan Zhang, Evangelos Theodorou, and Panagiotis Tsiotras. Trajectory distribution control for model predictive path integral control using covariance steering. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1478–1484. IEEE, 2022.

[90] Akash M Patel, Matthew J Bays, Ethan N Evans, Jonathan R Eastridge, and Evangelos A Theodorou. Model-predictive path-integral control of an unmanned surface vessel with wave disturbance. In *OCEANS 2023-MTS/IEEE US Gulf Coast*, pages 1–7. IEEE, 2023.

[91] Alberto Isidori. Elementary theory of nonlinear feedback for multi-input multi-output systems. In *Nonlinear Control Systems*, Communications and Control Engineering, pages 219–227. Springer-Verlag London, 3rd edition, 1995.

[92] Q. Zou and S. Devasia. Preview-based stable-inversion for output tracking. In *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, volume 5, pages 3544–3548 vol.5, 1999.

[93] Mihir Dharmadhikari, Paolo De Petris, Mihir Kulkarni, Nikhil Khedekar, Huan Nguyen, Arnt Erik Stene, Eivind Sjøvold, Kristian Solheim, Bente Gussiaas, and Kostas Alexis. Autonomous exploration and general visual inspection of ship ballast water tanks using aerial robots. In *2023 21st International Conference on Advanced Robotics (ICAR)*, pages 409–416. IEEE, 2023.

[94] Wei Cheah, Keir Groves, Horatio Martin, Harriet Peel, Simon Watson, Ognjen Marjanovic, and Barry Lennox. Mirrax: a reconfigurable robot for limited access environments. *IEEE Transactions on Robotics*, 39(2):1341–1352, 2022.

[95] PK Ward, Palitha Manamperi, Philip Brooks, Peter Mann, Waruna Kaluarachchi, Laurent Matkovic, Gavin Paul, C Yang, Phillip Quin, David Pagano, et al. Climbing robot for steel bridge inspection: Design challenges. In *Austroads Bridge Conference*. ARRB Group, 2014.

[96] Son Thanh Nguyen and Hung Manh La. A climbing robot for steel bridge inspection. *Journal of Intelligent & Robotic Systems*, 102(4):75, 2021.

[97] Son T. Nguyen, Anh Q. Pham, Cadence Motley, and Hung M. La. A practical climbing robot for steel bridge inspection. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9322–9328, 2020.

[98] Minh Nhat Vu, Michael Schwegel, Christian Hartl-Nesic, and Andreas Kugi. Sampling-based trajectory (re) planning for differentially flat systems: Application to a 3d gantry crane. *IFAC-PapersOnLine*, 55(38):33–40, 2022.

[99] Richard L Neitzel, Noah S Seixas, and Kyle K Ren. A review of crane safety in the construction industry. *Applied occupational and environmental hygiene*, 16(12):1106–1117, 2001.

[100] Maria Francesca Milazzo, Giuseppa Ancione, and Vesna Spasojević Brkić. Safety in crane operations: An overview on crane-related accidents. In *6th International Symposium on Industrial Engineering SIE 2015*, pages 36–39. University of Belgrade, Faculty of Mechanical Engineering, 2015.

[101] Avi Raj and Jochen Teizer. State of the art review of technological advancements for safe tower crane operation. In *41st International Symposium on Automation and Robotics in Construction*, pages 364–373. International Association for Automation and Robotics in Construction (IAARC), 2024.

[102] Neng Wan, Aditya Gahlawat, Naira Hovakimyan, Evangelos A Theodorou, and Petros G Voulgaris. Cooperative path integral control for stochastic multi-agent systems. In *2021 American Control Conference (ACC)*, pages 1262–1267. IEEE, 2021.

[103] Stepan Dergachev and Konstantin Yakovlev. Model predictive path integral for decentralized multi-agent collision avoidance. *PeerJ Computer Science*, 10:e2220, 2024.

Appendix A

TELEOPERATION EXPERIMENTAL PARTICIPANT-LEVEL PERFORMANCE

A.1 Participant-level performance for fastener inspection tasks

While the main result of the fastener inspection user study captures improvements of semi-autonomous control (SC) over velocity control (VC) across cumulative participant data in Section 4.4.3, participant-level results are discussed in this Appendix to address variation across users. When comparing results from individual participants, trends of reduced undesired oscillation, improved collision avoidance, and improved efficiency remain consistent when evaluating SC against VC. Interestingly, user variation demonstrates some outliers and some variation in how significantly improvements are for each participant.

A.1.1 Undesired Oscillations

Across all participants, Q1 in Fig. A.1(a) reported a mean reduction in undesired oscillations while using SC as compared with VC. However, some specific tasks, such as one task under participant 6, report large undesired oscillations under SC. It is hypothesized that undesired oscillations reported under SC are largely due to the camera payload oscillating axially, because this mode cannot be removed by the crane robot if excited. Overall, evaluating participant-level metrics suggest that SC reduces undesired oscillations compared with VC.

A.1.2 Collision Rate

Collision rate across cumulative participant data was improved by using SC as compared to VC as collisions were mitigated under SC. Subjective responses of Q2 in Fig. A.1(b) largely align with these results, where most participants perceived that collisions were easier to

avoid under SC, even if collision had not occurred during any completed tasks. Participants experienced differing collision rates under VC, as participants 5, 7, and 10 did not collide while using VC and participants 4, 6, and 11 experienced a collision rate of 50% or higher. A high discrepancy in collision rate under VC between participants could be from differing skill levels and differing risk tolerances.

A.1.3 Task Completion Time

Fig. A.3(a) shows the mean completion time per participant using VC and SC for successful trials. Overall, SC consistently reduced task completion times for the majority of participants as demonstrated through mean completion time reduction from VC to SC, as shown in Fig. A.3(b). Specifically, participants 1, 3, 6, 8, and 12 achieved reductions over 30%, demonstrating significant efficiency gains when using SC.

Variation in completion time improvements could be a result of varying participant control strategies. As an outlier, participant 4 experienced a time decrease of 23.5% on successful trials between VC and SC. This quantitative measure contradicts qualitative responses indicating increased undesired oscillations through Q1 when using VC compared with SC and perceiving higher efficiency through Q3 in Fig. A.1(c) with SC as compared to VC. The combination of a 50% collision rate and fast completion time on successful trials suggest participant 4 adopted a relatively aggressive control strategy compared with other participants.

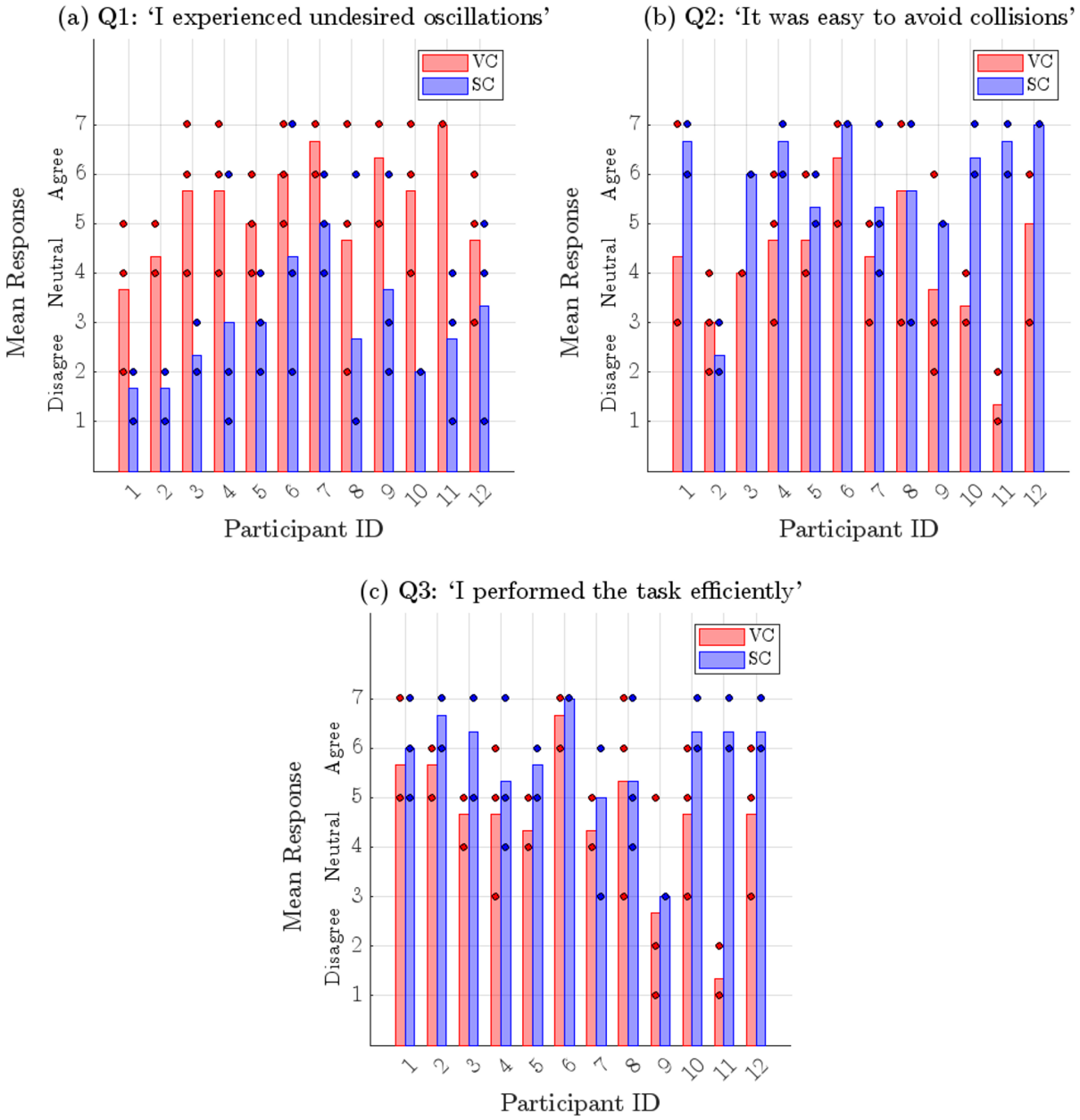


Figure A.1: Subjective questionnaire responses from 12 participants after performing teleoperated fastener inspection tasks. For each participant ID, responses collected after each task are depicted as dots on a seven point Likert scale (1: Strongly disagree; 4: Neutral; 7: Strongly agree) for velocity control (VC) and semi-autonomous control (SC). A bar indicating the mean response for each participant overlays the data points. Q1-Q3 are shown in subplots (a)-(c), respectively.

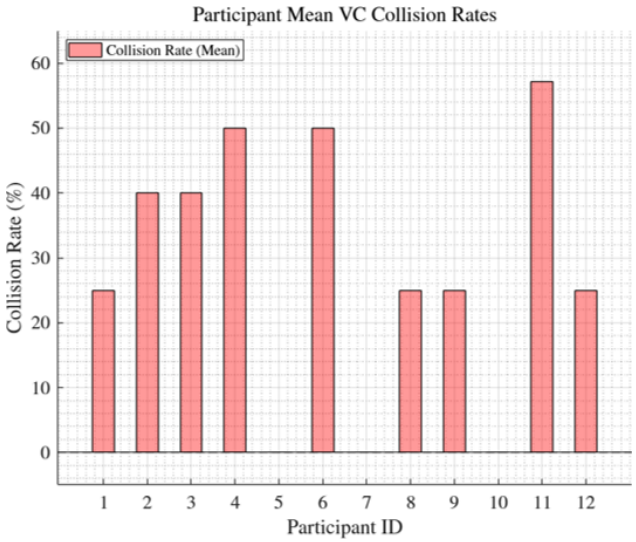


Figure A.2: Mean collision rate of 12 participants using velocity control (VC) while performing teleoperated fastener inspection tasks.

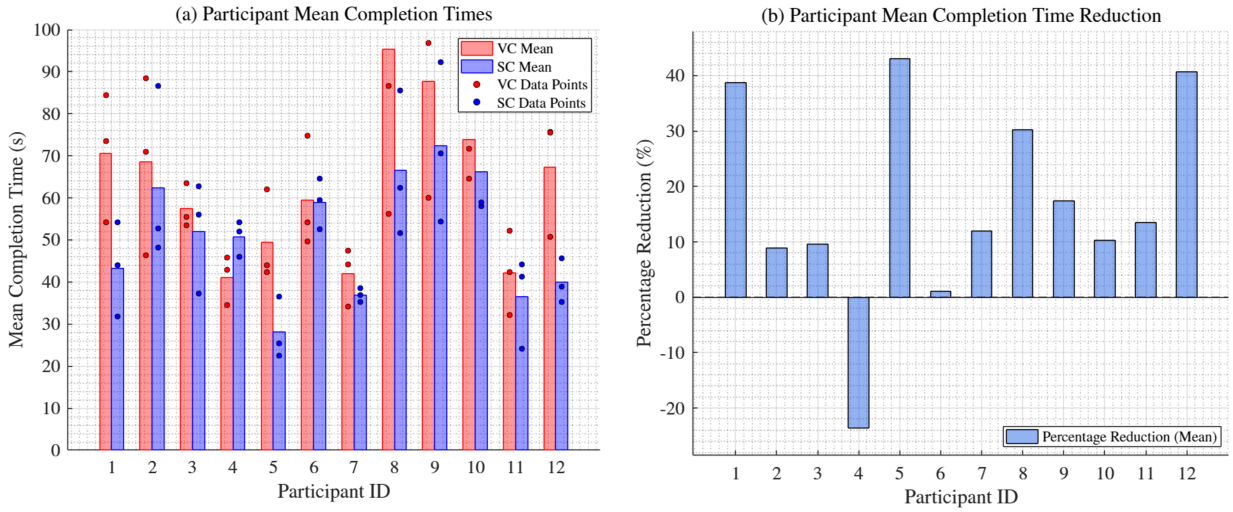


Figure A.3: Completion time of 12 participants performing teleoperated fastener inspection tasks for successful trials. (a) For each participant ID, task completion times are depicted as dots for velocity control (VC) and semi-autonomous control (SC). A bar indicating the mean completion time for each participant overlays the data points. (b) The percentage reduction in completion time of successful trials performed with SC as compared to VC for each participant. All but one participant demonstrated completion time improvements using SC as compared with VC.

Appendix B

TELEOPERATION EXPERIMENTAL PROCEDURES

B.1 Teleoperation procedures for fastener inspection tasks

Section 4.4.3 describes the fastener inspection task, in which participants teleoperated the crane robot around an obstacle to reach an inspection site within an experimental confined space and identified whether gaps were present in three fasteners using a video feed from the camera payload. All participants, either undergraduate or graduate students in mechanical engineering, were expected to have basic familiarity with joystick controls. However, participants were not expected to have prior knowledge of the crane robot or the fastener inspection task. Therefore, participants were introduced to the crane robot and task instructions through a presentation that comprehensively covered all necessary information, providing each participant with consistent information. A tentative participant schedule was established from preliminary non-participant trials as

1. System description and task instructions (estimated five minutes)

2. Fastener inspection task (repeated six times)

    (a) Facilitator configures fastener configuration (estimated one minute)

    (b) Participant performs fastener inspection task (estimated one minute)

Three additional minutes were included in the experimental time estimate to account for potential task restarts due to collisions. Each participant was estimated to complete the entire experiment in approximately 20 minutes. The rest of this section covers the procedural details for introducing and conducting the fastener inspection teleoperation experiments.

B.1.1 System description and task instructions

A presentation provided participants with a system description and fastener inspection task instructions. The first part of the presentation provided a system description by introducing the crane robot, control interface, and participant views. The presentation began by explaining how the crane robot operates inside aircraft wings by suspending a camera and described that the experiment aimed to blindly test different underlying controllers within the experimental confined space, as shown in Fig. B.1. Participants were then introduced to the crane robot controls, as outlined in Fig. B.2, where the joystick was described as controlling the camera payload's velocity horizontally and vertically along its two axes. A verbal description explained that left and right joystick motions corresponded to left and right camera payload velocity components, respectively, whereas forward and backward joystick motions corresponded to up and down camera payload velocity components, respectively. Additional information regarding the underlying control was not discussed to ensure that participants were not influenced by the selection of control algorithm. The following slide in Fig. B.3 described the camera views accessible to participants during experiments. To access the inspection view, the RICOH THETA iOS application live-streamed the camera payload video feed to a tablet. Participants were informed that they could pan and zoom the camera payload view during experiments, mirroring realistic capabilities available to operators using the device. The external camera was explained as supplying a static situational view for navigating the crane robot. These system descriptions provided participants with enough information to operate the crane robot.

The remainder of the presentation described the fastener inspection task to participants. Fig. B.4 depicted potential collisions within the experimental confined space, highlighting confined space walls and obstacles in red. Collision was described as failing the task, and the task would be repeated until completion. Participants were informed that any collision was considered a task failure, prompting participants to repeat the task until successful completion. The fastener inspection task was then introduced in Fig. B.5, where participants

were instructed to complete the task as efficiently as possible. The concept of efficiency was clarified as minimizing the time taken to complete the task. The initial position and fastener location were displayed along with a sample path to ensure participants understood the objective of navigating through the confined space. To clarify the inspection criteria, participants were shown sample images of fasteners with and without gaps. A sample set of markings was provided for the example fasteners, demonstrating the correct way for participants to mark them. After marking which fasteners had gaps, participants captured an image of the fasteners to simulate operator actions, then pressed a button to complete the task. The final slide explaining the task was displayed for participants to reference throughout the experiments.

B.1.2 Fastener inspection task

Fastener inspection tasks were set up by a facilitator and carried out by participants. Between a task start and a participant pressing the final button, the crane robot's timestamps and states were automatically logged. To systematically capture data that could not be recorded autonomously, such as whether a collision occurred, a facilitator manually filled out the sheet shown in Fig. B.6. Each participant completed three trials, with each trial consisting of two tasks — velocity control (VC) and semi autonomous control (SC) — performed in random order to remove order effects. Before each trial, a facilitator used a pseudo-random number generator to determine which control to use first and recorded it on the sheet under VC/SC. Note that for the second task in the trial, the complementary control choice to the first task would be used to collect an equal number of successes under each control. A pseudo-random number generator then determined the fastener gap order, which the facilitator configured and recorded on their sheet. As participants performed the task, a facilitator recorded the collision sequence (C) leading to success (S) under the sequence line to align with the recorded robot data.

The form in Fig. B.7 was provided to participants for systematic task data recording and completion of the subjective questionnaire. The form included a section to indicate whether

the fasteners had gaps, which participants completed during tasks. The form also included a questionnaire to be completed after each task. The participant filled out the form for the task as the facilitator set up the following task. Data from the facilitator and participant, along with the crane robot's timestamp and state data, was used to compile the results in Section 4.4.3.

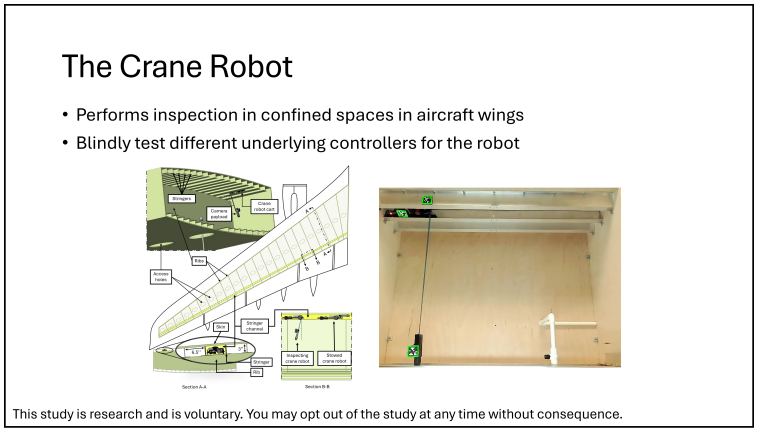


Figure B.1: Fastener inspection task system description and task instructions: The Crane Robot. This slide contains figures which describe the crane robot and the experimental confined space. The purpose of the fastener inspection task user study is introduced as well.

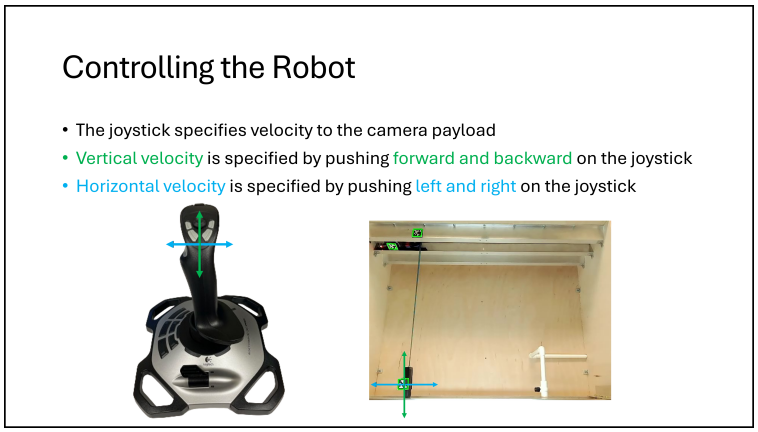


Figure B.2: Fastener inspection task system description and task instructions: Controlling the Robot. This slide contains figures of the joystick axes and corresponding camera payload velocities.

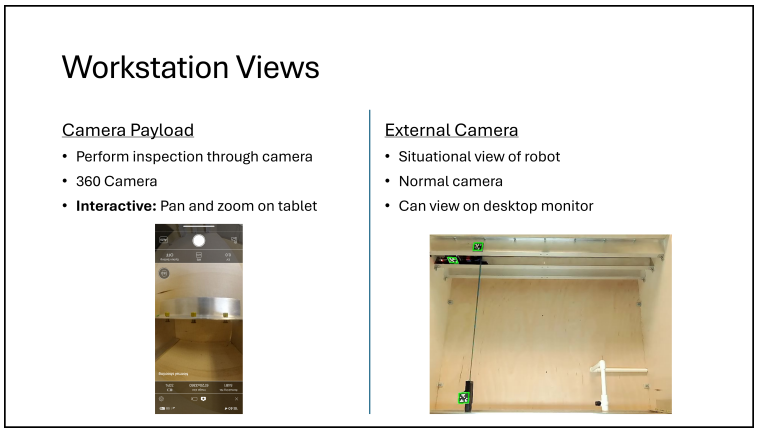


Figure B.3: Fastener inspection task system description and task instructions: Workstation Views. This slide shows sample views from the camera payload and the external camera. It explains how the live payload view is used for fastener inspection and how the static external camera view provides participants with situational awareness during navigation.

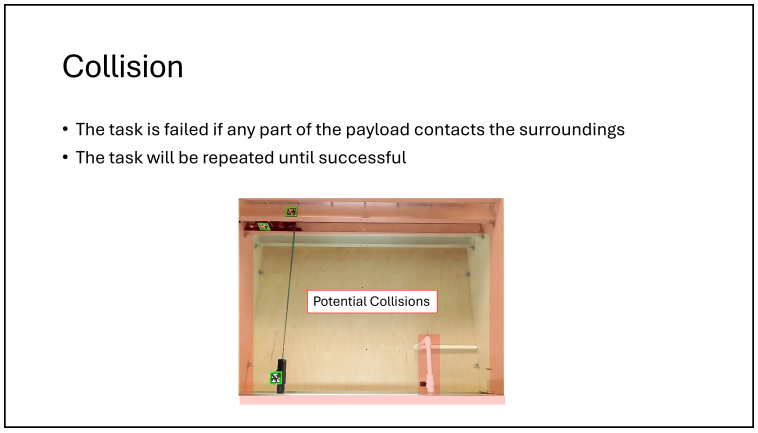


Figure B.4: Fastener inspection task system description and task instructions: Collision. This slide highlights potential collisions with the confined space walls and pipeline obstacle. It also describes the procedure of restarting tasks upon completion until success.

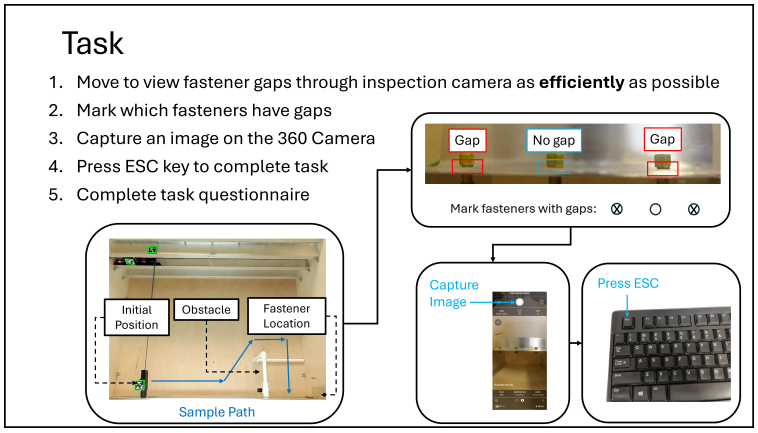


Figure B.5: Fastener inspection task system description and task instructions: Task. This slide includes a flowchart and outline of the main fastener inspection task steps. It provides information of the inspection site as well as provides a sample path to clarify navigation information. Example fasteners with gaps and corresponding marked indicators provide a clarifying reference for completing the task upon reaching the inspection site. Instructions for capturing images and confirming task completion are depicted for participant reference.

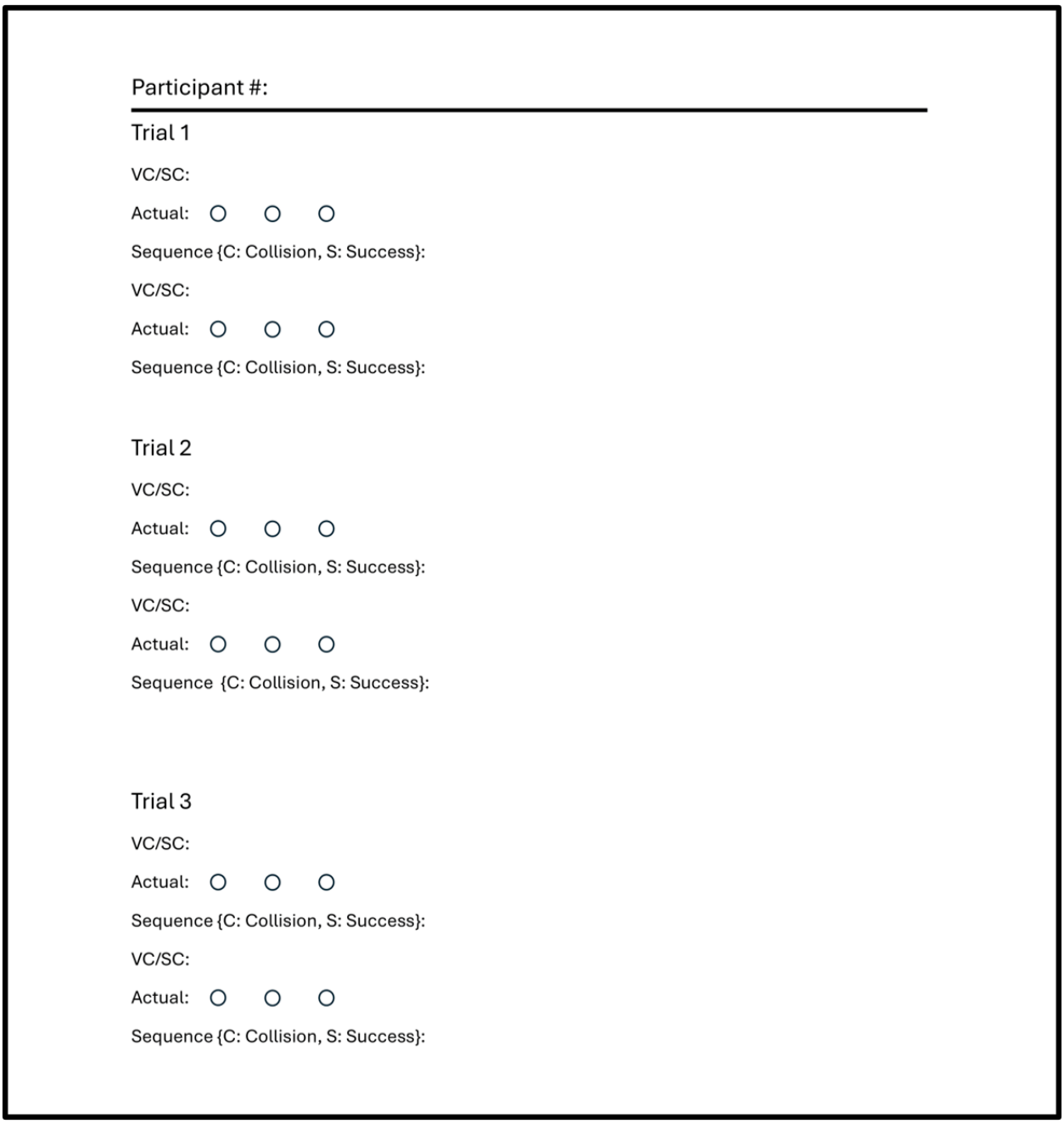


Figure B.6: Fastener inspection task trial information recording sheet. This form was used by the facilitator to record trial data. It includes sections for noting the control mode (velocity control (VC) or semi-autonomous control (SC)), fastener gap configurations determined through a pseudo-random generator, and a sequence for recording collisions leading to successful task completion.

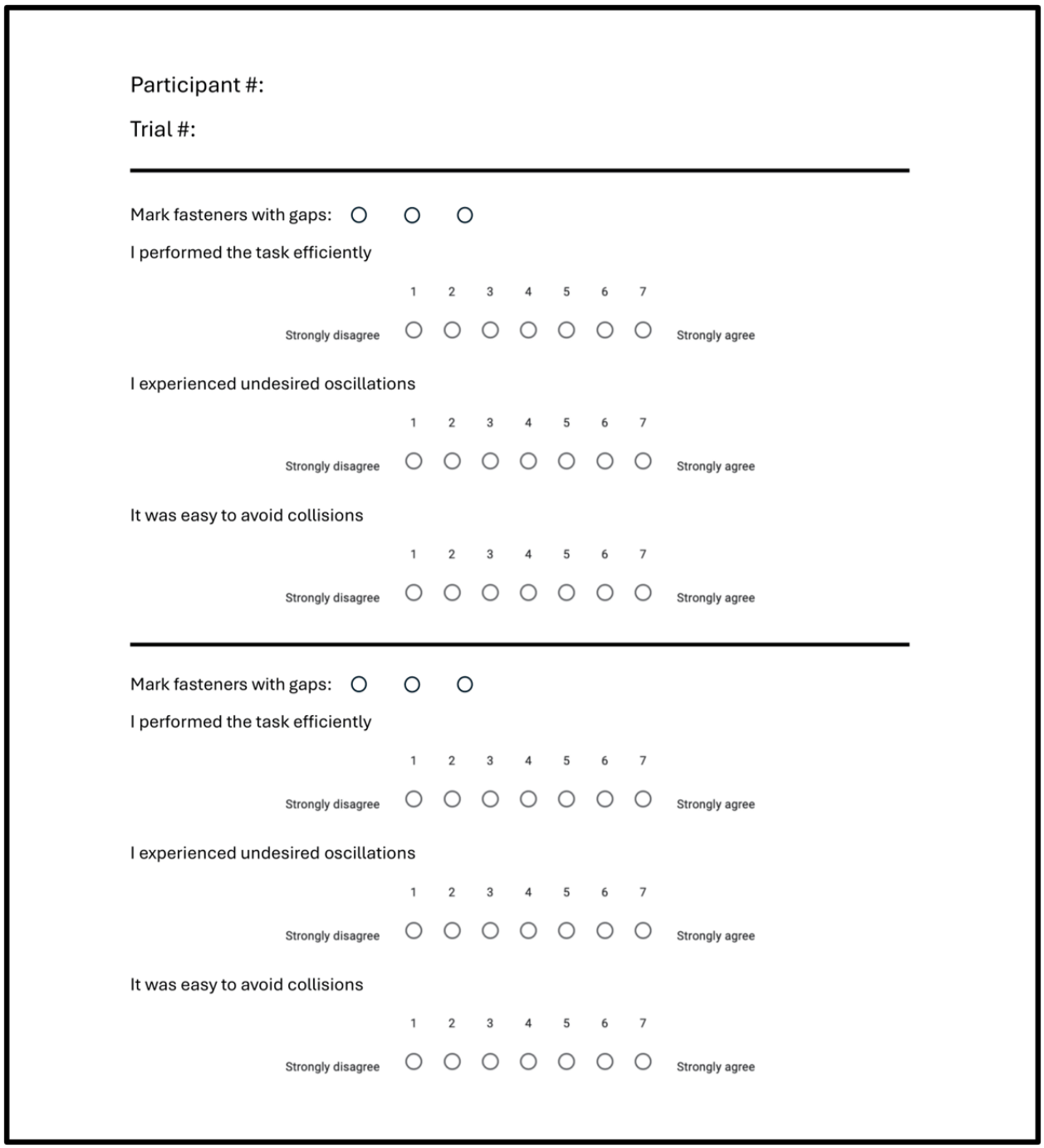


Figure B.7: Fastener inspection task participant data and feedback questionnaire. This form provided participants a structured means of marking fastener gaps during the task and answering the subjective questionnaire following task completion.