ACCURACY IMPROVEMENT OF INDUSTRIAL SERIAL MANIPULATORS FOR MANUFACTURING APPLICATIONS

A Dissertation
Presented to
The Academic Faculty

by

Toni Cvitanic

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Robotics in the
School of Mechanical Engineering

Georgia Institute of Technology
August 2021

ACCURACY IMPROVEMENT OF INDUSTRIAL SERIAL MANIPULATORS FOR MANUFACTURING APPLICATIONS

Approved by:

Dr. Shreyes Melkote, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Frank Hammond
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Stephen Balakirsky
Aerospace, Transportation & Advanced Systems Laboratory
Georgia Tech Research Institute

Dr. Larry Sweet
Amazon

Dr. Jun Ueda
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: April 21, 2021

ACKNOWLEDGEMENTS

The work described in this dissertation could not have happened without the support of some exceptional engineers, professors, and students. I would like to specifically thank Dr. Shreyes Melkote, my PhD advisor, for taking a chance on me as his first Robotics PhD student. I could count on him to always put in the effort to either directly help me when I ran into issues, if he could, or connect me to the people who would be able to help me. Secondly, I would like to thank Dr. Larry Sweet for bringing me on to my first robotics research project and introducing me to the world of industrial robotics and automated manufacturing, and for now serving on my committee. He has demonstrated interest and care in me and my work during and after his time at Georgia Tech. Next, I want to thank Dr. Stephen Balakirsky, who has been another bastion of support and a great source of ideas for multiple research projects, and who is also serving on my committee. Dr. Jun Ueda and Dr. Frank Hammond also have my gratitude for serving on my committee and supporting me through my time at GT with helpful feedback on qualifying exams and the paperwork and logistics necessary to qualify for graduation. Finally, I want to thank Dr. Katherine Fu, who took me on as a Master's student doing Engineering Design research. Though I ended up transitioning into robotics research, Dr. Fu was never anything but supportive, and she deserves her reputation of being someone who places high value on students' needs.

While the people mentioned so far have been great sources of support, advice, and ideas, the following people have my thanks for all the technical skills they helped me learn and all the work they shared with me to accomplish tasks in the lab. Vinh Nguyen has

helped me with more things than I can individually list here but to name a few: teaching me best practices for fixturing and machining materials, giving me guidelines for writing and submitting research papers, and being the single most-reliable person I could call to assist with any research-related issue that was impeding progress. Michael Sobrepera and Varun Agrawal (alongside Dr. Sweet) helped bring me into the world of robotics, and, more specifically, helped me get started in picking up the electrical and software engineering skills that I lacked coming from a mechanical engineering background, in addition to both being great mentors. Joshua Johnson, Chris Marince, and Allison Brown were not only exceptional in their jobs as Boeing engineers working alongside me in the lab, but have given me perspective on how robotics research translates into industry, and have supported me in my career search. I also want to thank Dario Valenzuela, Cliff Borowicz, and Ivan Berrios for their mentorship and support during my internship at Boeing and during our research meetings.

Finally, I would like to thank my wife, Chloe, for putting up with me over the past six years, and for brightening my day every time she comes home. I would also like to thank my father, Jaksa, mother, Vesela, sister, Lucia, and other family for their love and support. I could have never made it this far without them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

A	Angle about the tool axis
ANN	Artificial neural network
c	Damping coefficient
CMM	Coordinate measurement machine
CP	Complete pose
DoF	Degree of freedom
DPP	Decoupled partial pose
EE	End-effector
EKF	Extended Kalman Filter
F	External wrench
F_{eq}	Equivalent force used for stability prediction
$F(t)$	One-dimensional external force as a function of time
FRF	Frequency Response Function
GPR	Gaussian Process Regression
IMU	Inertial measurement unit
$J(\theta)$	Pose-dependent manipulator Jacobian
k	Stiffness coefficient
K_c	Complementary stiffness matrix
K_x	Cartesian stiffness matrix
K_θ	Joint stiffness matrix
KF	Kalman Filter
m	Mass coefficient

$\mathrm{M}(\theta)$	Pose-dependent mass matrix
$N(\theta, \dot{\theta})$	Pose-dependent damping, gravity, centrifugal and Coriolis matrix
PF	Particle Filter
Q	Process noise covariance matrix
q_k	End-effector quaternion at discrete time k
R	Measurement noise covariance matrix
TCP	Tool center point
UKF	Unscented Kalman Filter
$x(t)$	One-dimensional end-effector motion as a function of time
x_k	Three-dimensional end-effector position at discrete time k
α	Angular acceleration
δ	Measured one-dimensional end-effector deflections
τ	Torque
θ	Vector of joint angles
ΔX	End-effector deflection
Δt	Sample time
ω	Angular velocity
ω_n	Natural frequency

SUMMARY

Industrial manipulators are becoming more and more commonplace in the manufacturing world, but they have two major drawbacks compared to their more traditional machine tool and gantry-style counterparts: low global accuracy and low stiffness. While the repeatability of industrial manipulators is high, their low global accuracy is an issue when teaching the robot all the necessary points in an application becomes too time expensive. Low stiffness becomes an issue for processes with high external forces, such as machining. These processes can both statically deflect the end-effector and dynamically vibrate it. Both cases result in deviation from the nominal path, and thus accuracy tolerances may not be achieved.

Much research has been dedicated to modeling industrial robot stiffness both in the static and dynamic sense. These models are then often used to choose optimal robot configurations which will minimize deflections due to external forces for a given task. However, the pose which optimally minimizes dynamic deflections may not be the same as the one which optimally minimizes static deflections. In this thesis, the efficacy of pose optimization in minimizing the external force-induced deflections of the end-effector by using a static versus dynamic stiffness model approach is analyzed. A set of guidelines is determined as to what each respective optimization offers and when to use each one.

To address the low global accuracy of off-the-shelf industrial manipulators, the most common practice is robot calibration. This involves calibrating the geometry, stiffness, and other observable properties of the robot to more accurately represent where the end-effector will end up given an arbitrary position command. However, there is a limit

to how accurate these offline methods can make the manipulator. For even higher accuracy, it is necessary to close the loop with external sensors. Real-time external feedback for industrial manipulators for reaching accuracy tolerances better than 0.1 mm is a relatively recent development. As such, most methods focus on sensing only the position and orientation of the end-effector. For applications where the velocity and angular velocity need to be measured accurately, however, there is no established method using typical industrial sensors. This thesis addresses this issue by introducing a sensor fusion method of state-estimation of the industrial manipulator end-effector using inertial and laser tracker measurements. Different methods of fusing the data are compared, and their benefits to the overall state-estimation of the end-effector are analyzed.

Finally, closed-loop control of industrial manipulators has presented another challenge: how to model the closed-loop system. Most industrial robot controllers will accept real-time inputs only in the form of Cartesian or joint angle pose offsets. This makes it difficult to incorporate knowledge of the robot's dynamics into the closed-loop model, because dynamics relate forces and torques to the output motion of the robot, but pose offset commands contain no information about the corresponding force or torque. This issue is addressed in this thesis by introducing a novel method of representing pose offset commands with their "equivalent" forces. It is demonstrated that this new method allows for a good enough approximation of the closed-loop system to predict closed-loop stability offline, something that has thus far been impossible to do

CHAPTER 1. INTRODUCTION

1.1 Motivation

Modern industrial manipulators are designed to be repeatable, meaning that they can be commanded to reach historical poses to a high degree of accuracy. However, their volumetric (global) accuracy is lacking in comparison. Thus, the most accurate implementation of industrial manipulators is the "teaching" method, where the robot is manually driven to a set of poses, those poses are "taught" to it, then it executes programs which require it to reach the taught poses in some sequence. For manufacturing applications that only require discrete pose accuracy such as pick and place, spot welding, or part scanning, the teaching method is viable, though it does incur an increasing time cost when more poses are required to be manually taught for a given application. On the other hand, when it comes to applications which require the robot to accurately follow complex trajectories, such as machining (e.g. milling), teaching is no longer an efficient option, and the trajectory-following accuracy will be limited by the robot's global accuracy.

In addition to their global accuracy, industrial robots' compliance under external loads has been a limiting factor in their application to many manufacturing tasks, particularly machining [1]. High in-process forces and torques, which are characteristic of such tasks, cause the end-effector (EE) to deflect from its nominal trajectory, resulting in a drop in the trajectory-following accuracy. These forces can be static (constant), such as in a pick and place application with heavy loads, or dynamic (time-varying), such as in milling. Static forces will cause static deflections of the end-effector, while dynamic forces

will cause vibration of the robot structure, which results in time-varying end-effector deflections.

The volumetric accuracy limitations of industrial manipulators can be addressed through robot calibration, to an extent, but for the highest accuracies, external sensor feedback is necessary. Real-time closed-loop control of industrial manipulators has proven effective, but there is room for improvement. Most methods only consider position or orientation measurements, which are relatively straightforward because highly accurate industrial sensors, such as laser trackers, can measure them directly. When it comes to velocity or angular velocity measurements, they are either typically ignored or simply computed by differentiating the position and orientation data. Differentiating live measurement data amplifies any noise present in the data, even if filtering methods are used to reduce this effect. Lacking sensors that can measure velocity or angular velocity directly and accurately, the question arises as to whether a combination of existing industrial sensors can be used improve the estimation of these quantities over typical methods.

Industrial manipulators also present a unique challenge when it comes to closed-loop feedback. This arises from the fact that most industrial robot controllers only allow for real-time commands in the form of Cartesian or joint pose offsets. Thus, the user can input a pose offset, but then the robot controller determines what joint torques will be applied to reach that commanded offset. This might not be an issue if the structural vibrations of the industrial manipulator induced by high-jerk motions did not play a significant role in the instantaneous defection of the end-effector, but, as will be presented in this thesis, they do. There are several methods of modeling these robot dynamics, but they all follow the form

of relating input forces/torques to the output vibration of the robot. This presents a problem in applying these models to model the closed-loop system of industrial robots because the forces and torques associated with each pose offset command are unknown. This problem has not yet been addressed, and because of it, closed-loop controller implementations for industrial robots all must be tuned manually by iteratively running the application trajectories and adjusting controller gains until a satisfactory steady-state error is achieved.

As for addressing the error introduced by high external forces applied to the end-effector, traditional methods involve selecting robot configurations that maximize the robot's stiffness and thus minimize the undesirable deflections. However, compliance-based pose optimization can yield different solutions depending on whether the compliance model is a "static stiffness" (zero-order) model or a "dynamic stiffness" (second-order) model. So far, there are no guidelines as to which models should be applied when. This is not only an issue when it comes to determining the optimal pose for a given task, but also when it comes to deciding the amount of effort that should be put into stiffness model calibration for a given industrial robot. Stiffness models vary significantly in their complexity and the corresponding calibration effort needed to implement them. Ideally, the robot user should be able to put in the smallest possible calibration effort to achieve the desired accuracy for a given application.

The impact of addressing these shortcomings in industrial manipulators is to make them more viable alternatives to traditional CNC and gantry style systems in the manufacturing world. Especially compared to monumental gantry systems, industrial manipulators present a lower cost and much more reconfigurable option. However, the

majority of contemporary manipulators are used in handling and assembly operations, while less than 2% of manipulators purchased are used for material removal processes [1].

1.2 Research Objectives

The last section outlined some important issues which still exist when it comes to improving the global accuracy and stiffness of industrial manipulators. The following research objectives are designed to address these issues:

1. Compare the efficacy of pose-optimization using static versus dynamic stiffness models and develop a set of guidelines for a choosing the appropriate model and how they should be used in a manufacturing setting.
2. Develop a more accurate method for estimating the velocity components of the state of the robot end-effector using a sensor fusion of common industrial sensors.
3. Develop a method for incorporating knowledge of robot dynamics into a closed-loop system model that is accurate enough to permit model-based controller gain tuning.

1.3 Thesis Outline

This thesis will be organized as follows. Chapter 2 includes a comprehensive literature review covering past work in industrial robot calibration, closed-loop control, and state-estimation methods. Chapter 3 covers the experiments and analysis done to compare the efficacy of pose-optimization using static and dynamic stiffness models. Chapter 4 includes an analysis of sensor fusion methods for improved state-estimation of

robot end-effectors using industrial sensors. Chapter 5 presents a new method that allows knowledge of robot dynamics to be used to predict closed-loop stability of an industrial robot. Finally, Chapter 6 covers overall conclusions and recommendations for future work.

CHAPTER 2. LITERATURE REVIEW

A review of the literature relevant to accuracy improvement in industrial manipulators is presented in this chapter. The review is divided into the areas of robot calibration, compliance modeling in industrial manipulators, real-time feedback, and improved state estimation with sensor fusion.

2.1 Robot Calibration

A robotic manipulator's pose is fully defined by its joint values. Thus, there exists an input-output relationship between the input joint values and the output location and orientation of the end-effector. For a typical off-the-shelf manipulator, this relationship is provided in the form of the geometric transforms between each joint of the robot. While knowing these transforms allows one to have an estimate of where the end-effector will go given a set of joint commands, transform errors, compliance effects, gear backlash, and other sources of error reduce the quality of that estimate. Robot calibration refers to the effort to model the input-output relationship between the joint state and end-effector location of the robot as accurately as possible to minimize the volumetric positioning errors of the robot.

2.1.1 Kinematic Calibration

Kinematic robot calibration refers to calibrations which only address errors in the geometric transforms between the robot joints and do not consider other sources such as compliance or backlash. For most industrial manipulators, this is equivalent to calibrating

the link lengths and joint orientations of each link. Sometimes, the transform between the robot coordinate frame and an external world frame is also considered.

The most direct method of kinematic calibration is to directly measure the geometry of the robot through experimentation. Circle-Point Analysis [2] is an example of such a strategy, where each robot joint is rotated individually, then the origin and direction vector of the joint can be identified as the center and normal of the circle that was swept by the arm, respectively. More commonly applied are model regression methods [3-5]. Here, the robot is commanded to move to a large variety of poses within its workspace and the corresponding end-effector location is measured each time. Then, the best-fit geometric model is calibrated via a regression, which minimizes the error between the predicted locations of the end-effector computed using the geometric model and the actual measured locations. Other methods of note include forming virtual closed kinematic chains by pointing a laser attached to the end-effector at a stationary target [6] and iterative methods where calibration parameters are recomputed until convergence after each measured robot motion [7, 8].

2.1.2 Non-Kinematic Calibration

Non-kinematic calibrations, despite their name, calibrate the same parameters as kinematic calibrations, but also add parameters to address non-geometric sources of error. Most commonly, the additional calibration parameters attempt to model robot compliance. In the simplest case, they take the form of constants representing a linear or torsional spring stiffness at each robot joint[9-15]. More comprehensive calibrations attempt to model the compliance of the links by treating them as flexible beam elements [16]. Besides

compliance, other modelled non-kinematic parameters include the nonlinear behavior of the last robot joint [17] and the effect of temperature [18]. While non-kinematic calibrations capture more sources of error than kinematic calibrations and so typically result in higher robot accuracies, their trade-off is that a higher calibration effort is required to provide sufficient data to be able to model the additional non-kinematic parameters.

2.1.3 Non-Parametric Calibration

Non-parametric calibrations are purely data-driven models, meaning that model parameters to be calibrated are not identified before-hand, but instead the input-output relationship between the joint state and end-effector position is learned from the experimental data alone. These methods are well-suited for capturing error that is difficult to model parametrically, or which comes from unknown sources. Typically, for robot calibration, a parametric model is used as an initial estimate of the end-effector position, then a non-parametric model is used to predict the error in that estimate. Data-driven modelling techniques for robot calibration can include fitting polynomial functions of the joint angles to predict end-effector position error [19, 20], training regression models to predict Cartesian error based on the similarity of the robot configuration in joint or Cartesian space [21-24], and training artificial neural networks (ANNs) to predict end-effector error given the input commands to the robot [25-31].

2.2 Compliance Modeling

Some robot compliance modelling is included in non-kinematic robot calibrations, but it is mainly calibrated on the deflections caused by the gravity load of the links of the manipulator itself. For applications with high external loadings, such as the machining

processes of milling and drilling, the robot calibration can serve as an initial guess for the compliance. However, more dedicated compliance models, including dynamic models which capture the robot's vibrational properties, are often necessary for accurate force-induced deflection prediction. In this thesis, "static stiffness models" refer to zero order systems, which always output the same predicted deflection regardless of the frequency of the input force, and "dynamic stiffness models" refer to second order systems with mass and damping terms whose deflection outputs are a function of both the frequency and magnitude of the input forces.

2.2.1 Static Stiffness Modeling

Modern static stiffness models can be broken up into two categories: those that consider link flexibility [32, 33] and those that do not [34-37]. The latter models are referred to as "joint stiffness models" because all the compliance is assumed to occur at the joints of the manipulator. They are simpler models, in that they have a lower number of parameters to calibrate, and so typically require less calibration effort. However, they do not capture any link deformation effects, so they are not as accurate as those that do. Models which consider link deformation typically do so by calibrating six parameters for each joint instead of a single stiffness parameter, three for each translational direction and three for each rotational direction. Some static stiffness models also include parameters to capture friction and backlash effects [38] and to address the influence of gravity compensators on large manipulators [39].

Calibrating static models is done by applying a known external load to the manipulator and measuring the associated (steady-state) deflection of either the end-

effector or the joints directly (Figure 1). Gravity-based loads are most common, though sometimes reaction forces are used by clamping the end-effector to a rigid surface and driving the robot [38]. The deflection of the robot is captured either using external measurement systems or by reading the manipulator's internal encoder values [32-35, 37-39].

Figure 1 – Example static stiffness calibration setup.

2.2.2 Dynamic Stiffness Modeling

Traditional dynamic manipulator models [40-44] are of the form:

$$\tau = \mathrm{M}(\theta)\ddot{\theta} + N(\theta, \dot{\theta}) \quad (1)$$

Here, q is the vector of manipulator joint positions, τ is the vector of torques applied to the joints, $\mathrm{M}(\theta)$ is the pose-dependent mass matrix, and $N(\theta, \dot{\theta})$ captures centrifugal, Coriolis, friction, and gravitational effects. Like the first form of static models, in most cases, they

treat the manipulator links as being infinitely stiff. Exceptions include work where the robot links are modeled as Bernoulli beams [45]. These analytic models have the drawback that they have a strong dependence on how well the geometry and mass properties of the manipulator structure are known, information which is usually limited in the case of industrial manipulators. Thus, more recent dynamic models are either purely data-driven, or hybrid models which include a parametric base model augmented by data-driven models obtained through experimentation [46-51]. By fitting models to the data, it is possible to capture sources of vibration which are difficult to model parametrically, such as the unknown internal components of the robot links.

Dynamic models based on a form of (1) are typically calibrated by driving the manipulator along known trajectories and recording the corresponding joint torques. Data-driven models are calibrated by applying a known impulse or cyclic force to the end-effector, then measuring the corresponding vibration using external sensors. The downside to these methods is that the calibration process must be repeated at discrete locations within the robot's workspace, the dynamic properties are predicted using a regression or other learning technique which interpolates the properties between locations where experiments were conducted.

2.2.3 Compliance-Based Compensation and Pose Optimization

Once a compliance model is calibrated, if the process forces are known or can be predicted, it is possible to predict and compensate for the deflections of the robot induced by these forces. Extensive research has been done on compliance-based compensation using both static models [52-58] and dynamic stiffness models [59, 60]. Compensation

methods all derive from the principle that if deflection is predicted in a certain direction, it should be compensated by making the component of commanded robot motion in the negative of that direction larger than initially planned.

For applications that are fully defined using fewer degrees of freedom than the robot has available, there is a redundancy in possible robot poses which can be used to perform the task. For example, a drilling task is fully defined using 5 constraints because the orientation about the axis of the tool is unconstrained, so a six degree of freedom robot can use an infinite number of (reachable) poses to achieve the same drilling task. Because compliance is pose-dependent, it is possible to solve an optimization problem to select poses for a given task, which will minimize the deflection of the end-effector. As for compliance-based compensation, pose optimization to minimize deflection has been achieved using both static models [61-65] and dynamic stiffness models [66, 67]. What is lacking in the literature is a comparative study of pose-optimization between using static and dynamic compliance models to determine when one outperforms the other. This topic is covered in this dissertation.

2.3 Real-Time Position Feedback

No matter how comprehensive a robot calibration or compliance model is, it is never perfect. To account for the error in robot positioning left over after calibration, compensation, and optimization techniques have been implemented, it is necessary to measure it with an external sensor and correct it iteratively or in real-time. Feedback control of industrial robots is a relatively new area of research, with early works implementing iterative pose compensation [68, 69] or exploring the effectiveness of real-time feedback

using various sensors such as secondary encoders [70], optical measurement systems [71, 72], and laser trackers [73] (Figure 2). The latest research on real-time feedback has included comparing six degree-of-freedom (DoF) control versus three DoF control [74], implementing feedback on large-scale multi-robot systems [75], and reducing backlash error and sensor synchronization jitter on complex closed-loop trajectories [76].

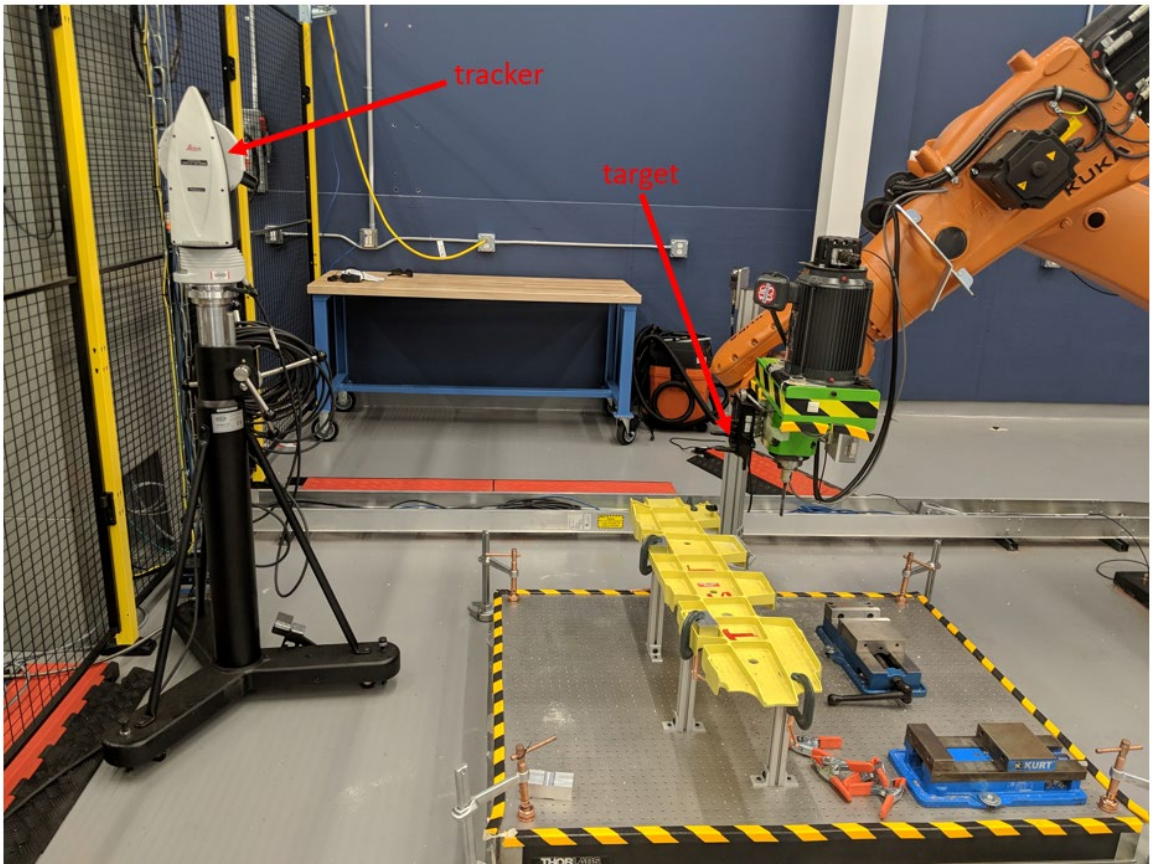

Figure 2 – Example closed-loop robotic milling setup with a laser tracker.

Because of the relatively smooth (low-acceleration) application trajectories and high accuracy of the sensors used for feedback, controller schemes in the literature have been kept relatively simple, usually in the form of PI [73, 74], PD [76], or PID [71, 77]. These schemes have the added benefit of being relatively easy to tune manually, so they can be implemented even without a good closed-loop transfer function for the robot system. In fact, because most industrial robot controllers are configured to only accept real-time inputs in the form of Cartesian or joint space offsets, obtaining a good closed-loop transfer function of the system is difficult because the input forces are unknown. Developing better closed-loop transfer functions that allow for offline controller tuning will be a subject of this thesis.

2.4 Improved State Estimation with Inertial Sensors

The sensors used for real-time feedback of industrial robots (as covered in the previous section) have high position and orientation estimation accuracy, most often in the form of laser trackers or optical tracking systems. Inertial sensors, on the other hand, can measure the acceleration and angular velocity components of the end-effector state. Because of the drift associated with integrating acceleration and angular velocity measurements to get position and orientation estimates, however, they are rarely used on their own (except for iterative compensation [78]). Outside of industrial robot applications, inertial sensors are most often combined with a low-frequency position and/or orientation sensor which corrects for the drift associated with integrating the inertial data [79-84]. However, these applications do not have especially strict accuracy tolerances, and so this type of sensor fusion is not used in industrial robotics.

In fact, the use of inertial sensors for state estimation of industrial robots is still a relatively new area of research. In [85], data from an inertial measurement unit (IMU) is fused with the position and orientation estimate of the end-effector derived from the robot's forward kinematics. An Extended Kalman Filter (EKF) and Particle Filter (PF) are compared in their ability to provide better real-time pose estimation and the ground truth is measured by a camera-based metrology system. The accuracy estimation is shown to significantly improve with the sensor fusion, but it is still not as accurate as using a more expensive laser tracker or optical measurement system to directly measure the position and orientation. Fusion with an optical tracking system is achieved in [86] using a Kalman Filter, and in [87] using an Unscented Kalman Filter and Particle Filter. Both works achieve higher position and orientation accuracy estimation with the sensor fusion. However, laser trackers have higher position measurement accuracy than camera-based systems, so it is unclear whether the position estimation could be improved by fusion with inertial data. Also, other parts of the state, including velocity and angular acceleration, are not considered by the existing literature. Finally, motion experiments in these works consist of translation only, so the difference in perceived acceleration by two different sensors on a rigid body is not a factor.

Also not present in the industrial robot literature involving fusion of inertial and position/orientation data is an analysis of the effects of sensor-to-sensor geometric transform error. This error is studied, for example, in [88], in the context of vehicle navigation. The sensitivity of the state estimation accuracy to the error in alignment of the sensor coordinate systems dictates the requirements for accurate fixturing and calibration

of the sensors. For translational motions, it is sufficient to know the orientation of each sensor's coordinate system, but for rotational motions, their origins must be known as well.

2.5 Summary

A review of the literature identifies some key gaps in knowledge regarding accuracy improvement of industrial manipulators including: 1) how to determine whether to utilize a static or dynamic stiffness model for a given application, 2) how to model a closed-loop system with an industrial robot controller well enough to predict closed-loop stability offline, and 3) whether the accuracy of the state estimation of the end-effector can be improved by adding inertial data, even if a top-of-the-line position/orientation sensor is already available and in use. Addressing the first two gaps has significant time and cost saving implications, by minimizing the stiffness model calibration effort in the first case, and by doing away with manual controller tuning for every new application trajectory in the second. Filling the last gap amounts to an overall improvement in current methods of end-effector state estimation for the relatively low cost of an IMU compared to the already used laser trackers and optical tracking systems. The remainder of the thesis will cover each of these gaps in knowledge and how they were addressed.

CHAPTER 3. POSE OPTIMIZATION USING STATIC AND DYNAMIC STIFFNESS MODELS

3.1 Introduction

This chapter explores the differences in application of both static and dynamic stiffness models for pose-optimization to reduce force-induced deflections on the end-effector. Both static and dynamic stiffness models are calibrated on a Kuka KR 500-3 serial manipulator. Then, a well-established mechanistic milling force model [89] is used to predict the forces on the end-effector for given experimental milling parameters. Using the predicted forces as inputs to the two calibrated stiffness models, optimal poses are selected using each model to minimize the deflection induced by cutting forces. The performance of each optimization is compared for a range of cutting conditions selected such that the frequencies of the cutting forces match the resonant frequencies of the robot in some cases and are far from the resonant frequencies in others. From the results, the discussion aims to answer the question of when it is appropriate to use each type of stiffness model for pose optimization, and what can be done in terms of adjusting cutting parameters to additionally minimize deflections.

3.2 Static and Dynamic Stiffness Models

A Kuka KR 500-3 has six rotational joints, the first three of which, referred to as the "arm" joints, mostly contribute to the translation of the end-effector, and the last three of which, referred to as the "wrist" joints, mostly contribute to the rotation (Figure 3). In milling, the orientation of the tool about the tool axis is unconstrained. For a six degree of

freedom manipulator, such as the KR 500-3, this means that an infinite number of pose configurations can be valid for any given milling task. Because the tool used in this work is not colinear with axis 6 of the robot, changing this degree of freedom (the orientation about the spindle axis) will correspond to a rotation in all joints of the robot, thus significantly altering its pose. By altering the pose, the robot's stiffness properties are, in turn, altered, and thus the potential for pose optimization arises. Note that pose optimization for milling is not limited to robots of this type and so long as a manipulator has at least six degrees of freedom, there is an optimization problem to be solved for maximizing the pose-dependent stiffness.

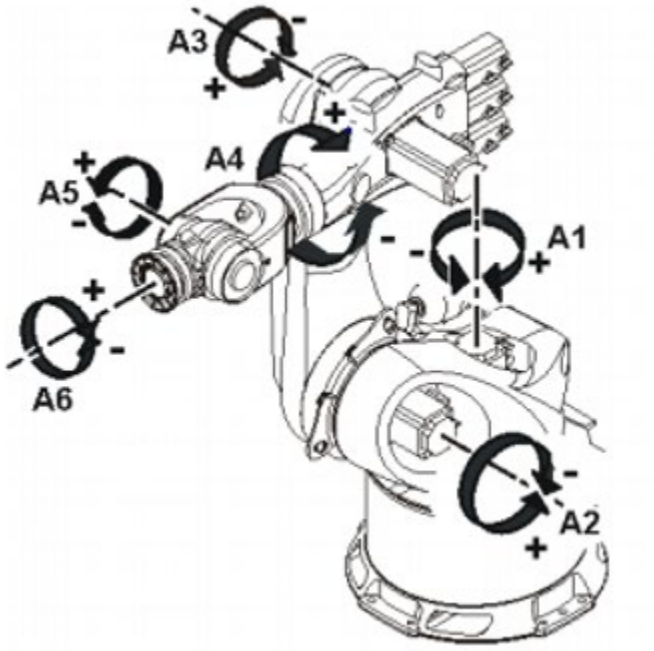

Figure 3 – Axis configuration of the KR 500-3 [90].

3.2.1 Static Stiffness Model Calibration

As covered in the literature review, there are many strategies for capturing the pose-dependent static stiffness of a serial manipulator. For this work, a six-parameter model is chosen where each joint is modelled as a linear torsional spring. This model is chosen over more complex ones, such as those in [32, 33, 38, 39], because it has a lower calibration cost and does not require extensive knowledge of the friction and structural properties of the robot links. Importantly, it is a static model, so it will always predict the same deflection no matter the frequency content of the process forces. This property is key in comparing it to dynamic stiffness models, whose deflection predictions are highly influenced by the process force frequency content.

To calibrate the static, six parameter model, both the complete pose (CP) method [36] and the decoupled partial pose (DPP) method are utilized [37]. Both methods are calibrated using the same experimental approach, but the DPP method decouples the stiffness computation of the wrist joints from that of the arm joints to better condition the calibration problem. The experimental approach entails applying a known wrench to the end-effector, F, using a gravitational load and measuring the corresponding translational and rotational deflections using three laser tracker targets (Figure 4). The applied wrenches were both directly vertical gravity loadings and directional loadings applied via a pulley. Directional loadings were chosen such that all robot joints experienced a significant torque for at least some of the loading cases.

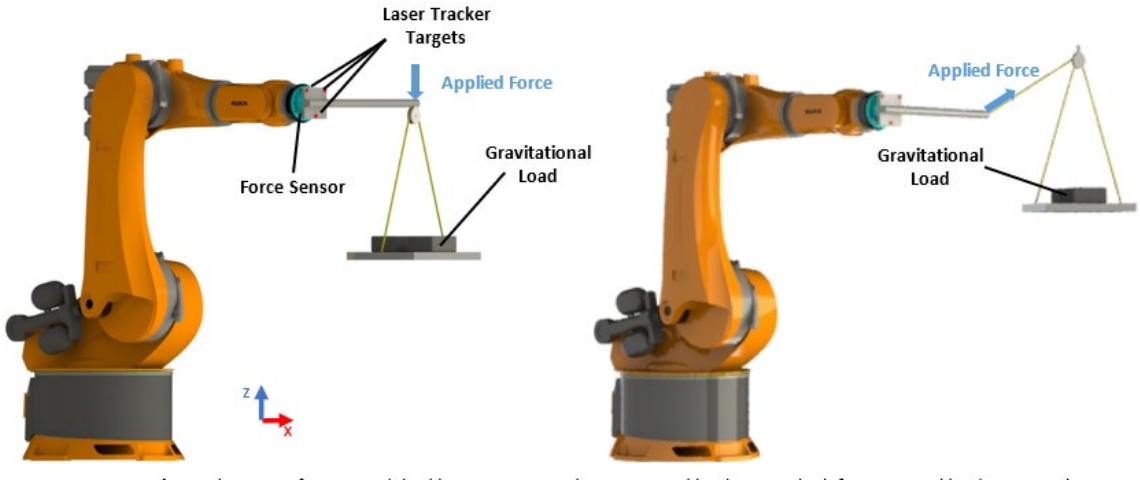

Figure 4 – Schematic of the static stiffness model calibration experimental setup. Both a vertical loading example (left) and a directional loading example (right) are pictured.

To model the active stiffness of the joints (with joint motors actively resisting deflection), the robot was constantly commanded to move back and forth for the smallest possible distance allowable. This prevents the brakes from engaging and allows the stiffness to be modelled while the robot is attempting to resist the external force-induced deflections, as it does in application. External wrenches were measured using a flange-mounted force/torque sensor (ATI Omega 160) and the corresponding deflections were measured using a Leica AT960 laser tracker. The tracker has a maximum permissible error of +/- 15 µm with + 6 µm/m. It was positioned no more than 3 m from its targets for a given experiment to avoid loss in accuracy with distance.

The deflection of the end-effector is given as a function of the joint stiffness values and pose of the robot by:

$$\Delta X = J(\theta)(K_\theta - K_c)^{-1} J(\theta)^T F \quad (2)$$

$J(\theta)$ represents the pose-dependent robot Jacobian, which describes the relationship between end-effector wrenches and joint torques, K_θ is the diagonal joint stiffness matrix, K_c is the complementary stiffness matrix, which describes the change in stiffness of the robot due to a small change in its pose from a given deflection, ΔX is the 6-dof deflection of the end-effector, and F is the external wrench. Following the methodology of [36], the influence of the complementary stiffness matrix was minimized by selecting poses which were far from singularity (by ensuring the inverse condition number of the normalized Jacobian is large). Using these poses, (11 simplifies to:

$$\Delta X = J(\theta) {K_\theta}^{-1} J(\theta)^T F \quad (3)$$

Equation (3) is used in the CP method. The DPP method, however, computes the arm joint stiffness values first. It does this by transforming the end-effector wrench to the wrist, then using:

$$\Delta X_{wrist} = J(\theta)_{wrist} {K_{\theta_{123}}}^{-1} J(\theta)_{wrist}^T F_{wrist} \quad (4)$$

Here, ΔX_{wrist} refers to the linear displacement of the center of the robot's spherical wrist, $J(\theta)_{wrist}$ is the robot Jacobian defined for the center of the wrist, F_{wrist} is the external wrench applied to the wrist, and $K_{\theta_{123}}$ is the diagonal joint stiffness matrix for the arm joints. Once $K_{\theta_{123}}$ is calculated, the wrist joint stiffness values are computed using:

$$\Delta X = J(\theta)K_{\theta_{123}}{}^{-1}J(\theta)^T F + J(\theta)K_{\theta_{456}}{}^{-1}J(\theta)^T F \quad (5)$$

Note that $J(\theta)$ in (5) is defined up to the end-effector and ΔX refers to the linear displacement of the end-effector.

Calibration experiments were conducted at ten poses (Table 1) and loading directions. Poses and loading directions were chosen such that sufficient torques were applied to each joint and the influence of K_c was minimized. To compute the joint torques for a given calibration experiment, the following relation was used:

$$\tau = J(\theta)^T F \quad (6)$$

where, τ represents the joint torques. End-effector deflections were predicted using (3) for the CP method and (4) and (5) for the DPP method. The Euclidean norm of the difference between the measured and predicted end-effector deflections was minimized using a nonlinear least squares approach to solve for K_θ. The calibrated joint stiffness values for each method are given in Table 2. These values are consistent with trends identified for other industrial manipulators of this type and payload in terms of approximate magnitudes and in that the arm joints have a higher stiffness than the wrist joints [36, 37, 91, 92]. The DPP method yielded a lower stiffness value for every joint compared to the CP method. This is most likely because of a scaling issue that arises with the CP method. The CP method is minimizing the difference between screw displacements, while the DPP method is only minimizing the difference between linear displacements. Given that the angular displacements are much smaller in magnitude than the linear displacements, the cost incurred by the angular displacements in the minimization process can underrepresent the

physical displacement. The result is that the solution estimates the robot to be stiffer than it really is. To confirm this, stiffness values were recomputed using the CP method after scaling up the angular displacements by a factor of 10, which resulted in a joint stiffness solution of [2.13, 1.59, 1.12, 1.89, 0.26, 0.17] MNm/rad, which is a lower arm joint stiffness than the solution yielded by the DPP method. Additionally, as a check, the joint stiffness values were recomputed assuming K_c was not negligible, and the resulting values differed from the given solutions by $< 10 \frac{Nm}{rad}$. Finally, the KR 500-3 has a gravity compensator which changes the effective stiffness of Joint #2 as a function of the robot configuration by up to 30% [39]. To address this influence, the joint stiffness values were recomputed after allowing Joint #2's stiffness to vary by up to 50%. Stiffness-based pose optimization yielded the same solutions with the recomputed values, and so, at least for the poses used in this work, an accurate calibration of the gravity compensator is not necessary. Once the joint stiffness values are known, the Cartesian stiffness at any pose is given by:

$$K_x = J^{-T} K_\theta J^{-1} \tag{7}$$

A final consideration for this calibration is that it was assumed that the spindle stiffness was much higher than the robot's Cartesian stiffness at any pose. To confirm this, a finite element (FE) analysis was performed on a model of the spindle. The stiffness at the least stiff point in the model, which corresponded to the tool tip, was computed to be 231 MN/m. This was compared to the robot stiffness by computing the Cartesian stiffness for the experimental poses used in this work and using the largest singular value of the Cartesian stiffness matrix to represent the maximum robot stiffness at that pose. This

yielded a maximum robot stiffness of 12 MN/m, thus validating the high spindle stiffness assumption.

Table 1 – Robot joint angles (degrees) for static stiffness calibration poses

Test No.	$\boldsymbol{\theta_1}$	$\boldsymbol{\theta_2}$	$\boldsymbol{\theta_3}$	$\boldsymbol{\theta_4}$	$\boldsymbol{\theta_5}$	$\boldsymbol{\theta_6}$
1	19.67	-52.14	117.9	11.86	-66.08	-4.84
2	26.35	-32.54	123.49	-8.99	-90.81	-0.15
3	58.28	-10.5	105.98	42.39	-115.04	-2.57
4	44.71	-70.81	58.16	138.51	-41.63	-125.84
5	-23.13	-85.41	80.84	70.31	83.63	-71.78
6	2.39	-34.82	69.97	115.13	72.1	176.98
7	25.21	-55.7	130.03	169.88	72.17	-179.78
8	42.99	-40.39	97.42	-177.38	72.1	-179.8
9	42.99	-57.91	110.04	-155.79	51.1	-179.83
10	28.12	-55.4	84.06	-167.6	27.07	-179.85

Table 2 – Calibrated joint stiffness values

Method	$\boldsymbol{K_{\theta_1}}$ $(\frac{\boldsymbol{MNm}}{\boldsymbol{rad}})$	$\boldsymbol{K_{\theta_2}}$ $(\frac{\boldsymbol{MNm}}{\boldsymbol{rad}})$	$\boldsymbol{K_{\theta_3}}$ $(\frac{\boldsymbol{MNm}}{\boldsymbol{rad}})$	$\boldsymbol{K_{\theta_4}}$ $(\frac{\boldsymbol{MNm}}{\boldsymbol{rad}})$	$\boldsymbol{K_{\theta_5}}$ $(\frac{\boldsymbol{MNm}}{\boldsymbol{rad}})$	$\boldsymbol{K_{\theta_6}}$ $(\frac{\boldsymbol{MNm}}{\boldsymbol{rad}})$
CP	3.93	3.46	2.77	1.57	0.24	0.17
DPP	2.86	2.36	1.67	0.24	0.16	0.14

3.2.2 Dynamic Stiffness Model Calibration

The dynamic stiffness model calibration follows the method developed in [49]. Figure 5 shows the experimental modal analysis test setup for a single robot configuration. At each configuration, an impulse is applied using an instrumented impact hammer (PCB 086D05) and the vibration response in the excitation direction is measured by a single-axis accelerometer (PCB 352A21) mounted at the tool tip. The experiments were conducted in the X, Y, and Z Cartesian directions (see Figure 7 for directional reference) for each robot configuration. Figure 6 shows a representative Frequency Response Function (FRF) in the Cartesian X direction. Note that the FRF shows multiple modes of vibration in addition to a single maximally compliant mode. As reported in [49, 93-95], the compliance of the non-

dominant modes is considered negligible, and only the most compliant mode is modelled. Thus, the tool tip equations of motion along a single axis (assuming negligible initial conditions) for a given robot configuration can be written as a second order transfer function with inertia, damping, and stiffness coefficients of the form:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t) \tag{8}$$

Unlike in the static model, the external forces are now time varying. To fit the mass, damping, and stiffness coefficients for a given pose and direction, the second order system that best fit the FRF's most compliant mode (blue dashed line in Figure 6) was used. To develop the complete dynamic model, experiments were repeated for each Cartesian X, Y, Z direction at discrete positions within the robot's workspace corresponding to the test locations in (Table 4). At each pose, the end-effector was rotated in ten degree increments about the spindle axis to span the range of possible poses at each location.

It should be noted that this method of dynamic model calibration does not account for cross-coupling effects in the vibration modes. In other words, an excitation force along a given direction is assumed to only produce vibration in that same direction. In the recent literature on dynamic modelling of industrial robots, this is common practice. However, to account for cross-coupling, one strategy would be to fit additional second order systems to each XY, YZ, XZ direction. Theoretically, this should better predict the full vibration of the end-effector for a given excitation force, but it substantially increases the calibration effort needed for the dynamic model, and it is unclear how much of an improvement can be gained. As a check, experimental modal analysis was performed at three test poses to determine the magnitude of the compliance in non-excitation directions in comparison to

that of the excitation direction. The results are displayed in Figure 42 and Figure 43 in Appendix A. They show that vibration in directions orthogonal to the excitation direction is much smaller compared to the vibration along the excitation direction, thus supporting the decision to neglect these modes. A more detailed exploration of the extent of pose-dependent cross-coupling vibration effects for serial manipulators is considered outside the scope of this thesis.

Figure 5 – Robot modal test setup

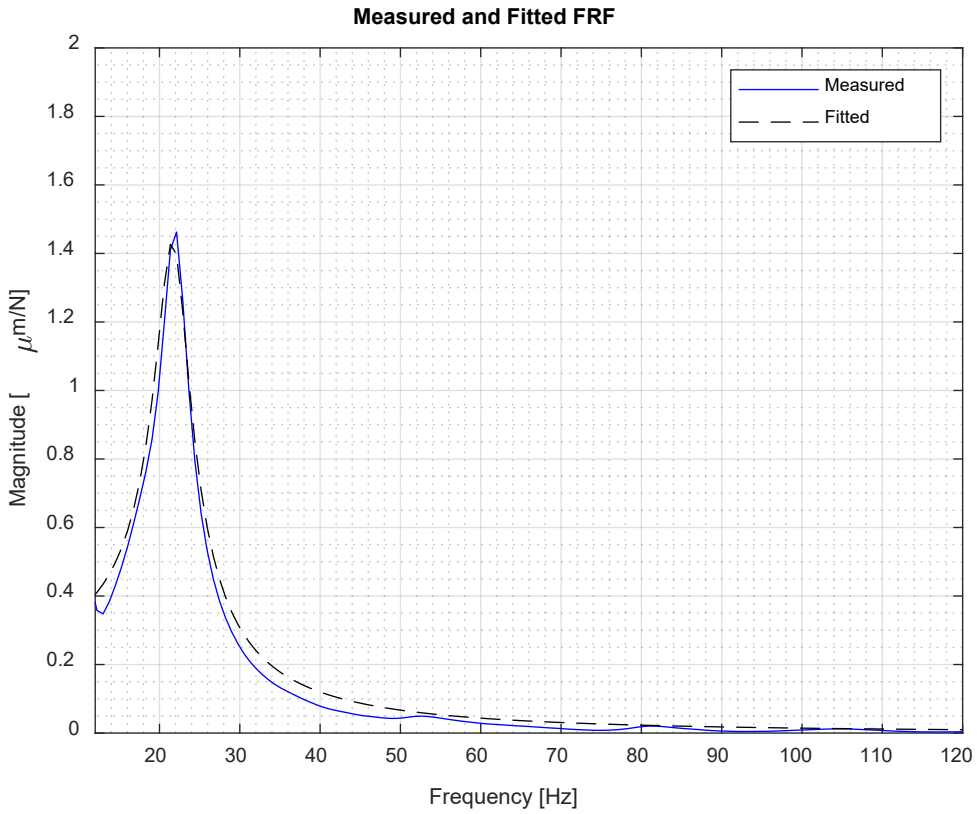

Figure 6 – Measured end-effector FRF (solid line) and the corresponding second order transfer function fit (dashed).

To determine the modal vibration properties for unsampled regions of the robot's workspace, a data-driven Gaussian Process Regression (GPR) model was used [96]. GPR is a supervised learning technique which computes the best non-parametric function to fit a set of observations. These non-parametric functions are assumed to be Gaussian Processes, meaning that they can be specified by their mean and covariance functions. Taken from [96], given a set of noisy training observations, y, and observed outputs, f_*, their joint probability distribution can be defined as:

$$\begin{pmatrix} y \\ f_* \end{pmatrix} = \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{9}$$

Here, K is the covariance matrix and X and X_* are the training and test input parameters, respectively. σ_n^2 is the variance of the Gaussian noise of the observations. GPR uses a Gaussian generator to iteratively generate sets of functions specified by their mean and covariance matrices until a function is found which best fits the observed data. Once this best-fit function is determined, the most probable output to an input vector not used in training can be predicted using:

$$\overline{f_*} = \mathrm{K}(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1} y \tag{10}$$

The covariance of the corresponding prediction can also be computed as:

$$\mathrm{cov}(f_*) = \mathrm{K}(X_*,X_*) - K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1} K(X,X_*) \tag{11}$$

For interpolating the dynamic model parameters between poses, a single GPR model was trained for each X, Y, Z direction and each mass, stiffness, and damping coefficient for a total of 9 models per pose. The input vector used to predict the dynamic model parameters at a given pose consisted of the Cartesian location of the end-effector and orientation, A, about the spindle axis. Once the dynamic parameters are known, robot deflections can be predicted for a given input time-varying force by integrating the second order transfer function (8) twice with respect to time. The computed dynamic parameters for the poses used in this work are reported in Table 16 in Appendix A. The values for *k* represent a Cartesian stiffness and are analogous to the values in the first three rows and

columns of the Cartesian stiffness matrix computed using the static model in (7). Table 17 in Appendix A gives the stiffness and frequency of the second most dominant mode of vibration at each test configuration. The high stiffness values relative to those of the most dominant modes support the exclusion of non-dominant modes from the dynamic models developed in this work.

Finally, given the high stiffness of the tool compared to the manipulator, most linear tool tip deflection due to rotational vibration of the manipulator was assumed to be captured by the data-driven dynamic modeling technique. While not covered here, future work could involve capturing the rotational vibration of the end-effector using multiple accelerometers mounted at different locations on the end effector. Alternatively, a joint vibration model could be calibrated, but these models rely on the assumption of infinitely stiff links, and on accurate knowledge of the geometric and inertial properties of the manipulator links, which is often not known for industrial manipulators.

3.3 Pose Optimization

The redundant degree of freedom present when milling with a six-axis robot is represented by the angle about the axis of the spindle, which can range from -180 to 180 degrees or less depending on joint limit boundaries. This range can also be additionally constrained by workspace obstacles, but for the purposes of this study, it is assumed that no such constraints exist. The angle about the spindle axis is designated as “A” as in Figure 7. To capture the range of possible poses for a given point along the milling path, an exhaustive search is used over the range of possible A angles in 1-degree increments. As a check that this increment is small enough to accurately capture the variation in stiffness

over poses, optimal A angles were computed using a 0.5-degree increment and the resulting solutions were within 1 degree of those computed using the 1 degree increment. The cost function of the optimization is designated as the second norm of the linear deflection of the end-effector. Depending on the application, it may be more desirable to optimize stiffness along a particular direction, but in the interest of generality, it is assumed that deflection in any direction is equally undesirable for these experiments.

Figure 7 – Optimization of angle "A" about the spindle axis.

3.3.1 Milling Force Modeling

To predict the deflections for a given pose, it is necessary to know the external forces being applied on the end-effector. In this work, a well-established mechanistic milling force model is used [89]. The infinitesimal tangential, radial, and axial milling forces at milling cutter immersion angle ϕ with respect to the i^{th} milling cutter tooth are given as:

$$dF_{t,i} = [K_{te} + K_{tc}h_i(\phi, z)]dz \tag{12}$$

$$dF_{r,i} = [K_{re} + K_{rc}h_i(\phi, z)]dz \tag{13}$$

$$dF_{a,i} = [K_{ae} + K_{ac} h_i(\phi, z)]dz \tag{14}$$

Here, subscripts *e* and *c* represent the edge force and cutting force coefficients, respectively (Table 3), *z* represents the integration height along the helix angle of the cutter, and $h_i(\phi, z)$ represents the undeformed chip thickness (related to the feed) for a given immersion angle and axial height along the helix. The integration of infinitesimal forces followed by the transformation to a fixed coordinate system results in:

$$F_F = \sum_{i=1}^{N} \int_0^a \left[-dF_{t,i} \cos \phi_i(z) - dF_{r,i} \sin \phi_i(z)\right] dz \tag{15}$$

$$F_T = \sum_{i=1}^{N} \int_0^a \left[dF_{t,i} \sin \phi_i(z) - dF_{r,i} \cos \phi_i(z)\right] dz \tag{16}$$

$$F_A = \sum_{i=1}^{N} \int_0^a \left[dF_{a,i}\right] dz \tag{17}$$

Here, F_F, F_T, and F_A represent the forces in the feed, transverse, and axial directions at a given milling cutter immersion angle, ϕ, which is incremented by a pre-determined time step and the force calculation is repeated. Note that the effect of cutter axis offset runout is assumed to be negligible for purposes of the current analysis.

Table 3 – Edge and cutting force coefficients used for force prediction.

Coefficient	Value
K_{te}	$6.0899 * 10^3$ N/m
K_{tc}	$9.0377 * 10^8$ Pa
K_{re}	$6.6017 * 10^3$ N/m
K_{rc}	$2.9379 * 10^8$ Pa
K_{ae}	$1.1169 * 10^3$ N/m
K_{ac}	$2.7783 * 10^8$ Pa

To verify the prediction of the cutting forces, cutting tests were conducted on a workpiece mounted on a three-axis piezoelectric force dynamometer (Kistler 9257b) with a sampling rate of 1 kHz. The cutting parameters (feed rate, axial and radial depths of cut, spindle speed) were selected to cover the same range of parameters used in the stiffness optimization experiments described in the subsequent sections. These tests indicated that the mean and variance of the forces could be predicted using the mechanistic milling model to within 10% of what was measured by the dynamometer and, more importantly, the dominant frequencies of the forces could be predicted to within 1 Hz of what was measured. Some discrepancy between the prediction and measured forces is expected for robotic milling due to the deflection of the tool during cutting [97]. Nevertheless, the close match in magnitude and even more precise match in frequency of the predictions to the measured forces makes predicting the milling force a viable strategy for pose optimization, which is further supported by the experimental results detailed in the subsequent sections.

3.3.2 Static Stiffness Based Optimization

To predict the end-effector deflections using the static model, the 2-norm of the predicted maximum force vector for a given milling operation is used as input (Figure 8). For each pose corresponding to each possible A angle, the deflection is computed, and the

optimal pose is chosen to be the one with the smallest deflection. This is equivalent to minimizing the objective function:

$$\|\Delta X\| = \|K_x(\theta)^{-1}F\| \tag{18}$$

Here, ΔX represents the screw deflection of the end-effector caused by the external wrench, F. $K_x(\theta)$ is the pose-dependent Cartesian stiffness matrix of the robot. There is a potential scaling issue that arises because the magnitudes of the angular deflections are smaller than that of the linear ones, so linear deflections may incur a disproportionate cost. However, in practice, angular deflections were measured to be very small ($< 0.1°$), so they were assumed to be negligible, and optimization aimed only to minimize the predicted linear deflections.

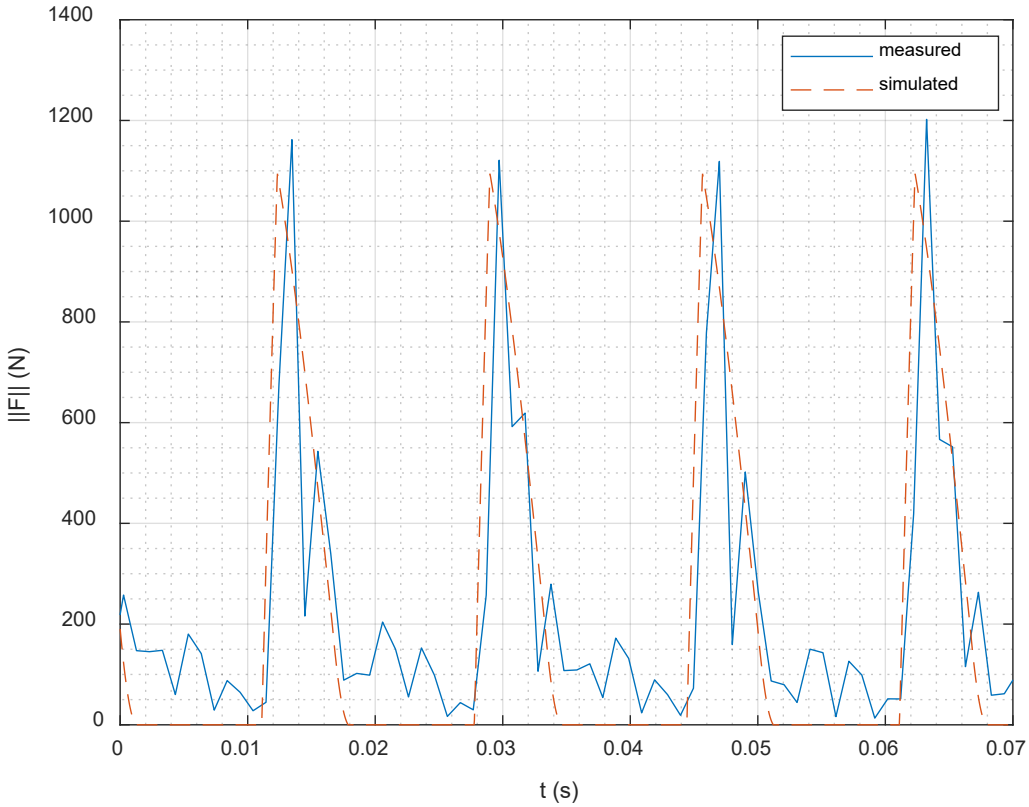

Figure 8 – Measured and predicted milling force magnitudes. The 2-norm of the peak force is used for static stiffness model-based pose optimization.

3.3.3 Dynamic Stiffness Based Optimization

Like for the static case, predicted milling forces were utilized for the dynamic stiffness-based optimization. However, instead of taking the peak forces as inputs, the full time-varying milling forces were used as inputs to the second order transfer functions of the dynamic models. The deflections were simulated for each possible A angle, and the optimal pose was designated as the one which minimizes the maximum 2-norm of the deflections. In other words, the maximum amplitude of the tool tip vibration was minimized (Figure 9). The corresponding objective function to be minimized is:

$$max(x(t)) = \iint_0^{t_{max}} \frac{F(t) - c(\theta)\dot{x}(t) - k(\theta)x(t)}{m(\theta)} dt^2 \quad (19)$$

Here, $max(x(t))$ is the maximum predicted deflection, $F(t)$ are the time-varying cutting forces, and $m(\theta)$, $c(\theta)$, and $k(\theta)$ are the pose-dependent mass, damping, and stiffness parameters determined from the corresponding dynamic models, respectively. t_{max} is the time at which the predicted maximum deflection occurs for a given cut.

This optimization strategy does not account for poses where process-induced dynamic instabilities such as mode coupling and regenerative chatter can occur. However, the focus of this work is on minimizing force-induced deflections due to robot stiffness using static versus dynamic models, not to compare the two model-based optimizations' effectiveness in minimizing chatter. However, since the mass and stiffness parameters are known for a given pose, it is possible to check for the possibility of model coupling chatter [98]. The poses used in this work were confirmed not to be susceptible to mode coupling chatter. In addition, a quick analysis of the deflections presented in the results reveals a lack of significant regenerative chatter.

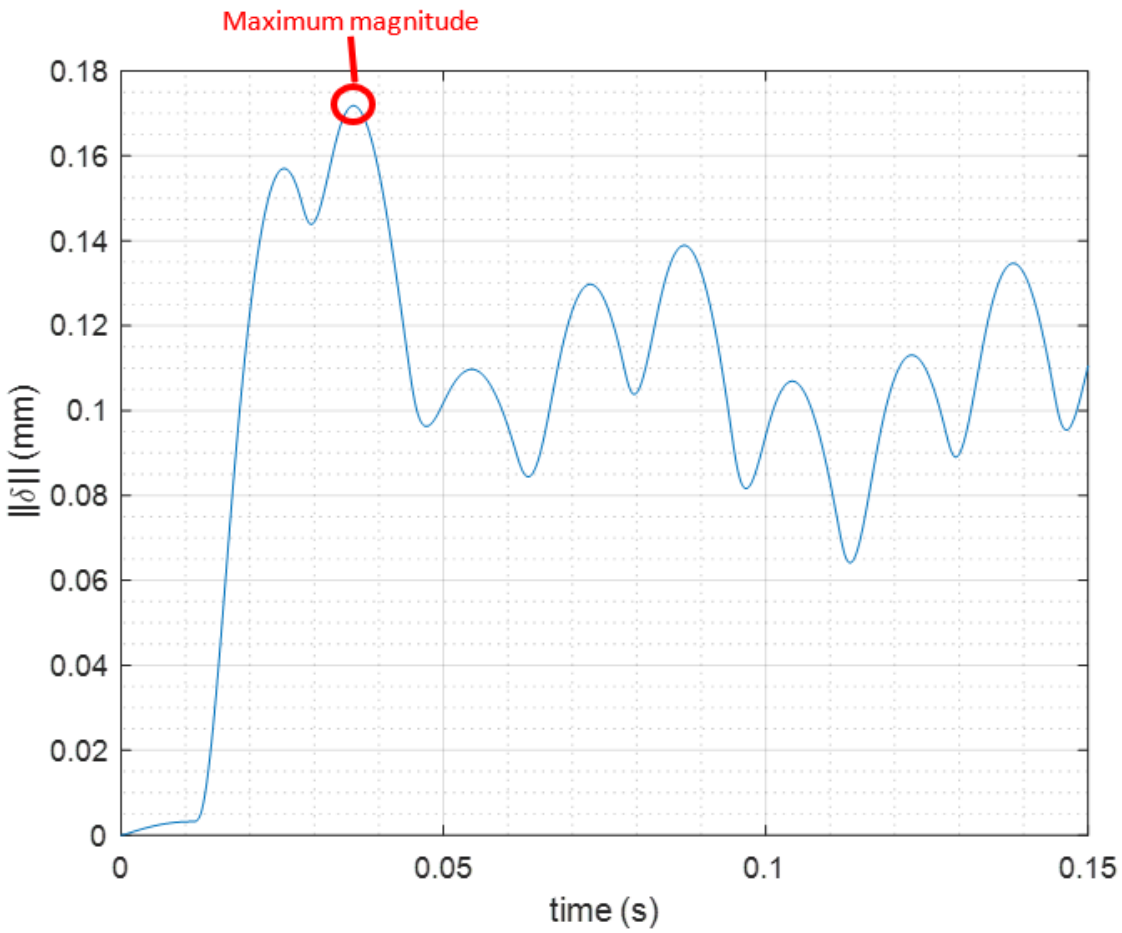

Figure 9 – 2-Norm of the simulated deflections (δ) as a function of time. The maximum amplitude of the deflections is minimized in the dynamic model-based optimization.

3.4 Experimental Setup

To compare the effectiveness of static and dynamic stiffness model-based pose optimizations, the two independent variables selected for experiments were location of the workpiece in the robot workspace and the frequency content of the milling forces. The location in the workspace is an important factor because it determines the set of poses that are achievable for a given milling operation. By varying the workspace location, both the range of pose-dependent static and dynamic stiffness properties available for the pose optimization are altered, thus it is possible to gain insight into which conditions favor one

model-based optimization over another. The second variable, the frequency content of the milling forces, only affects the dynamic stiffness-based optimization. The hypothesis is that the dynamic model-based optimization will attempt to select poses for which the resonant frequencies are far from those of the milling forces. The static model, having no knowledge of the robot's resonant frequencies, may underperform in cases where the statically optimal pose happens to correspond to one whose resonant frequencies match those of the cutting forces. Milling force magnitude was fixed because it only scales the predicted deflections and does not affect the solution for the optimal pose.

3.4.1 Choosing Locations in the Workspace

To ensure a broad range of possible robot dynamics were captured, workspace locations were selected such that the natural frequencies of the robot exhibited different trends across the reachable A angles at each location. Some locations were selected for which the natural frequencies (ω_n) had significant variation across poses (Figure 10). Note that the lack of smoothness in the ω_n plots is caused by the GPR interpolation of ω_n between sampled points. Other locations were chosen for which the natural frequencies did not significantly vary across reachable A angles (Figure 11). Note that the configuration displayed in Figure 11 is close to singular and would likely be avoided in practice. For experimental purposes, however, it is good to consider as a comparison point between static and dynamic pose optimization for a case where the dynamic model may not be able to drive the optimal pose solution's natural frequencies away from those of the milling forces. It is hypothesized that the dynamic stiffness model-based optimization should outperform the static one for poses in which the natural frequencies vary significantly, as it should have more of an opportunity to select a pose which minimizes resonance from cutting forces.

Conversely, the static model cannot account for these changing dynamics with robot pose and cannot therefore leverage them to select the theoretically most optimal pose.

Finally, to verify that a good range of the robot's stiffness characteristics was captured across the selected locations in the workspace, the maximum static stiffness, corresponding to the largest singular value of the Cartesian stiffness matrix, of the robot was evaluated at each experimental workspace location. Between nonsingular poses, the maximum stiffness values varied within 20% of each other and between close-to-singular and nonsingular poses, they varied over 100%. This indicates that the experimental poses covered a good range of relatively low and high static stiffness regions of the robot's workspace.

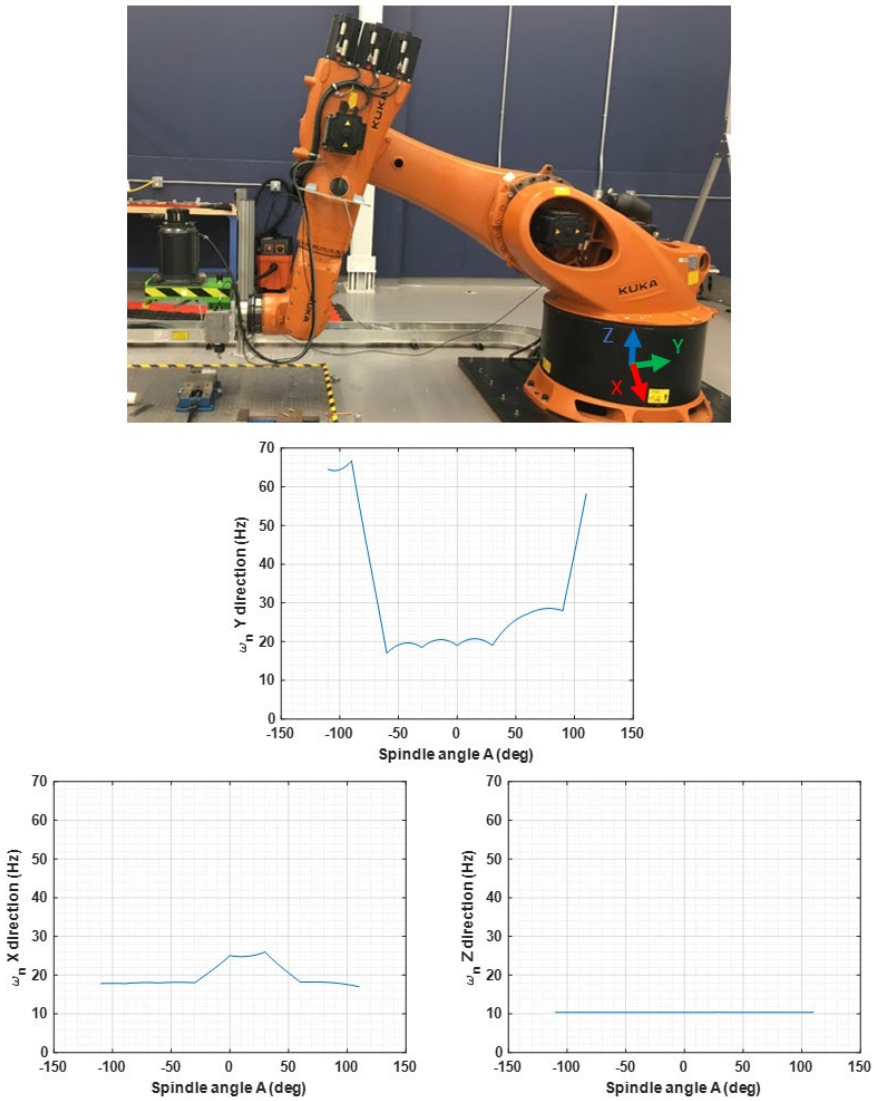

Figure 10 – Predicted robot natural frequencies in the three orthogonal directions of the robot base frame as a function of angle A. In this case, the natural frequencies in the Y direction significantly vary with A. The TCP is located at X: -10.73, Y: -2400.47, Z: 303.41 mm. In the pictured configuration, A=0°.

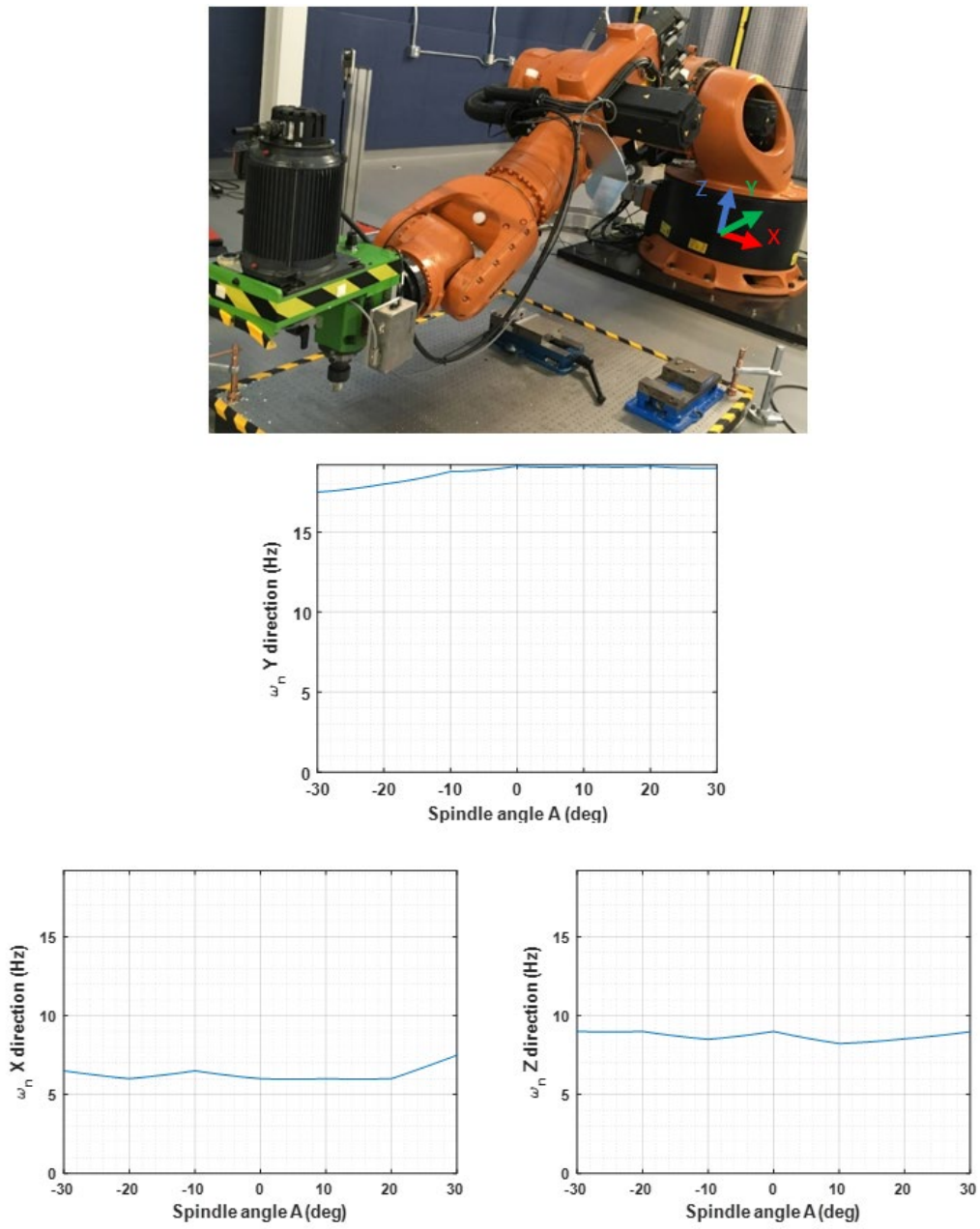

Figure 11 – Predicted robot natural frequencies in the three orthogonal directions of the robot base frame as a function of angle A. In this case, the natural frequencies do not significantly vary with A. The TCP is located at X: 11.16 mm, Y: -3100.09 mm, Z: 303.19 mm. In the pictured configuration, A=0°.

3.4.2 Milling Parameters

Milling was performed on an Aluminum 6061 workpiece. The cutting parameters were selected such that the magnitude of the milling forces was large enough to cause measurable end-effector deflections which exceeded the robot's repeatability (+/- 0.08 mm). A constant radial depth of cut of 6.35 mm, constant axial depth of cut of 5.08 mm, and a constant feed per tooth of 0.526 mm per tooth were used. The cutting tool was a 25.4 mm diameter uncoated high-cobalt content high speed steel with two helical teeth and a helix angle of 30 degrees (Kennametal Inc., Latrobe, PA). All cuts were performed dry and in the climb milling mode. To vary the frequency of the milling forces, the spindle speed was varied, but the feed rate was adjusted in each case such that the feed per tooth remained constant for all cuts.

3.4.3 Measuring Deflections

A Leica AT960 laser tracker with a six degree of freedom target (Leica T-Mac) mounted on the robot was used to measure deflections of the end-effector. The tracker can measure at 1 kHz with an error of < 0.0254 mm, which was confirmed by measurement of reference scale bars whose lengths are known to within 0.0025 mm. The transformation between the tracker target and tool tip was calibrated prior to experimentation. During cutting, position and orientation measurements were captured and transformed to the tool tip to determine the tool tip deflections.

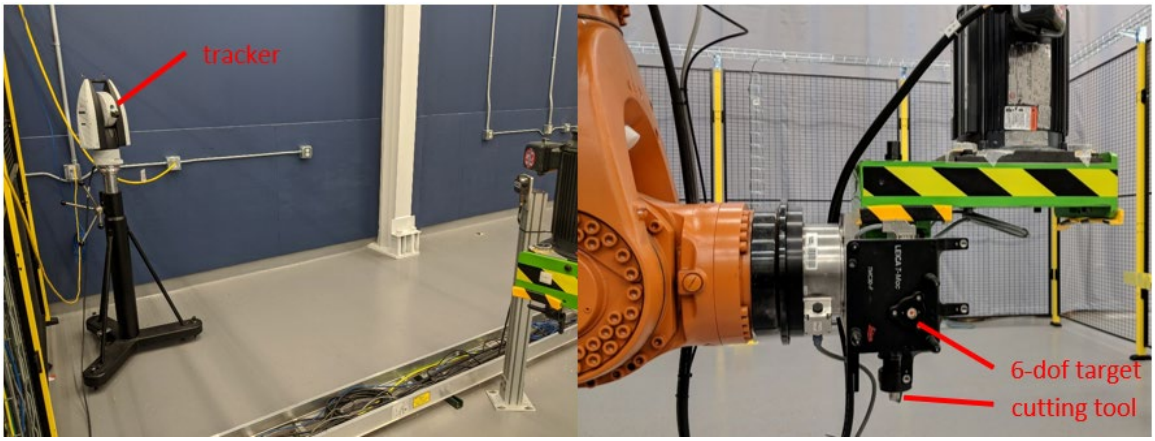

Figure 12 – Left: laser tracker (Leica AT960). Right: 6-dof target (Leica T-Mac) mounted on the end-effector.

3.4.4 Results

Table 4 lists the experimental parameters. At each test location, a set of two peripheral end milling cuts was performed: one using the static stiffness optimization and one using the dynamic stiffness optimization. All experiments were performed with the cutting tool pointing along the negative robot Z axis. Each set of two cuts was performed once at a spindle speed that generates time-varying milling forces with frequency content close to the natural frequencies of the robot, and again at a spindle speed high enough to be far from the dominant natural frequencies of the robot. The tooth passing frequency is determined from the spindle speed by:

$$\text{Tooth passing frequency} = \frac{\text{spindle rpm}}{60} * \text{number of teeth} \tag{20}$$

Note that the static stiffness optimization will always converge to the same solution regardless of spindle speed. For the static stiffness optimizations, both the CP and DPP models yielded the same solutions, so the optimal A angle to which they converged is reported as a single value in the results. The first two workpiece locations (test numbers 1-8) were in regions of the workspace where the robot dynamics varied significantly with angle A. Predicted deflections for each A angle are depicted in Figure 13 for test numbers 1-4. The Euclidean norm of the corresponding measured deflections from experimentation are depicted in Figure 14. The predicted deflections as a function of A for test numbers 5-8 are depicted in Figure 15 and the corresponding measured deflections are depicted in Figure 16. Table 5 lists the mean and standard deviations of the deflections for test numbers 1-10, for reference.

Table 4 – Experimental parameters by test number. TCP refers to the cutting tool tip.

Test No.	Tooth Passing Frequency (Hz)	TCP X (mm)	TCP Y (mm)	TCP Z (mm)	Optimization Approach	Optimal "A" Angle (deg)	Natural Frequency Range (Hz)
1	64.5	-10.73	-2400.47	303.41	Static	110	10-68
2	64.5	-10.73	-2400.47	303.41	Dynamic	52	10-68
3	107.5	-10.73	-2400.47	303.41	Static	110	10-68
4	107.5	-10.73	-2400.47	303.41	Dynamic	49	10-68
5	60.0	615.9	-2400.92	304.47	Static	110	9-67
6	60.0	615.9	-2400.92	304.47	Dynamic	45	9-67
7	107.5	615.9	-2400.92	304.47	Static	110	9-67
8	107.5	615.9	-2400.92	304.47	Dynamic	45	9-67
9	25	-0.25	-2111.24	291.63	Static	110	11-25
10	25	-0.25	-2111.24	291.63	Dynamic	53	11-25
11	60.0	11.16	-3100.09	303.19	Static	12	7-18
12	60.0	11.16	-3100.09	303.19	Dynamic	20	7-18

Table 5 – Measured deflections by test number.

Test No.	Mean Deflection (mm)	Standard Deviation of Deflections (mm)
1	0.42	0.19
2	0.17	0.08
3	0.19	0.03
4	0.15	0.03
5	0.27	0.15
6	0.25	0.07
7	0.19	0.03
8	0.19	0.04
9	0.46	0.28
10	0.49	0.28

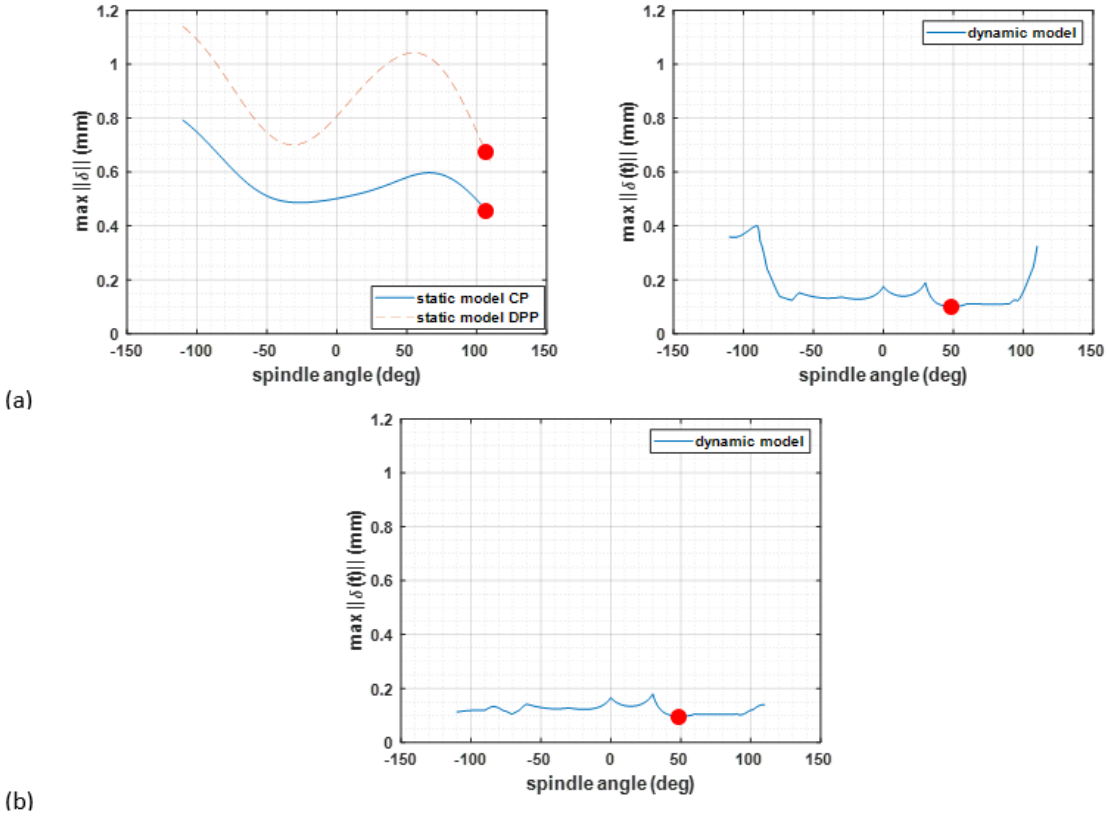

Figure 13 – (a) Maximum Euclidean norm of the predicted deflections using the CP, DPP static stiffness models, and dynamic models as a function of A for test numbers 1 and 2. (b) Predicted deflection norms using the dynamic stiffness model for test number 4. The red dots indicate the A angle corresponding to the minimum predicted deflection.

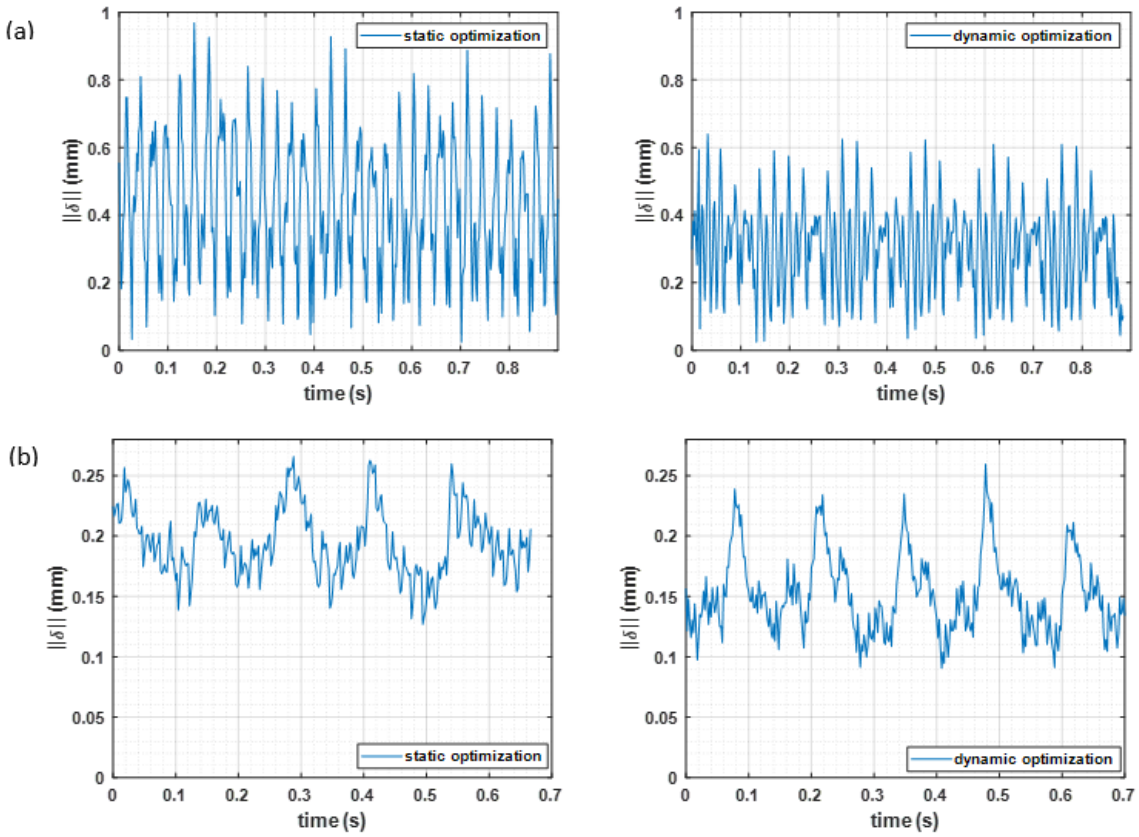

Figure 14 – Euclidean norm of the measured deflections for test numbers 1-4 for a tooth passing frequency of (a) 64.5 Hz and (b) 107.5 Hz. The lower frequency is within the range of the robot's natural frequencies at this pose while the higher one is not.

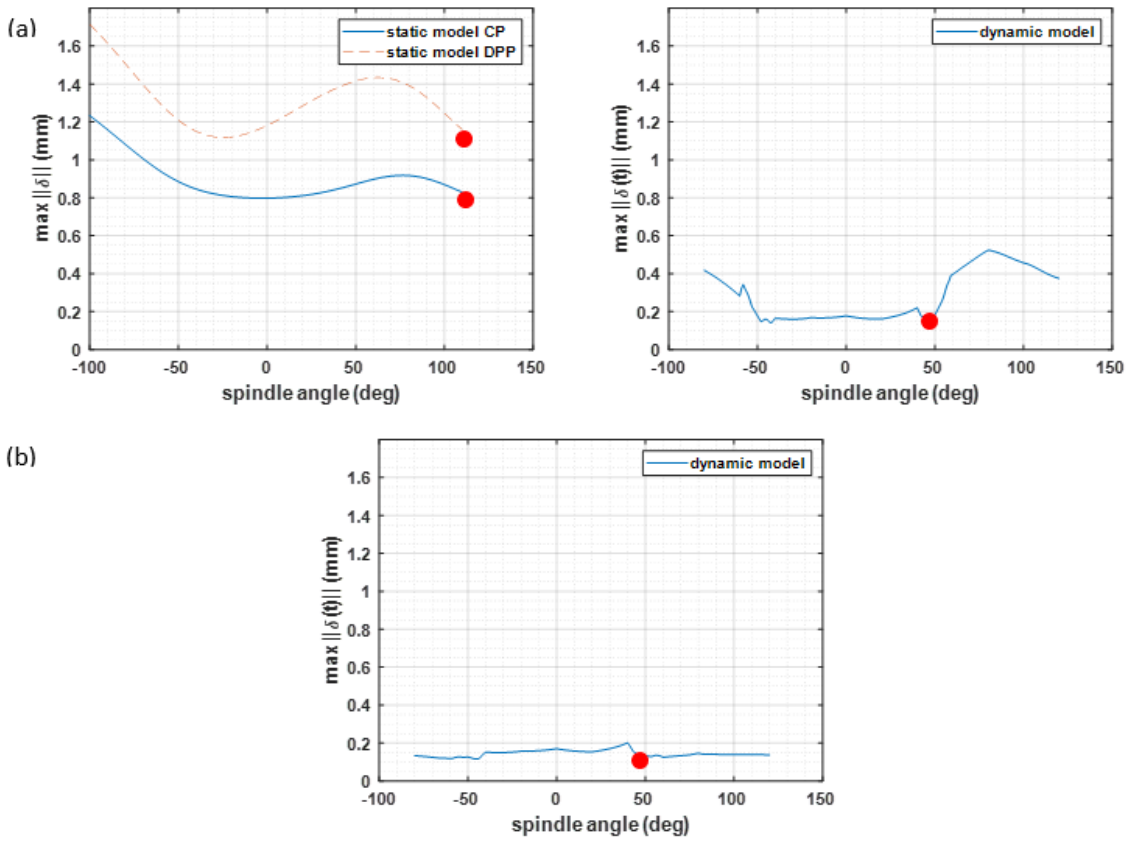

Figure 15 – (a) Maximum Euclidean norm of the predicted deflections using the CP, DPP static stiffness models, and dynamic models as a function of A for test numbers 5 and 6. (b) Predicted deflection norms using the dynamic stiffness model for test number 8.

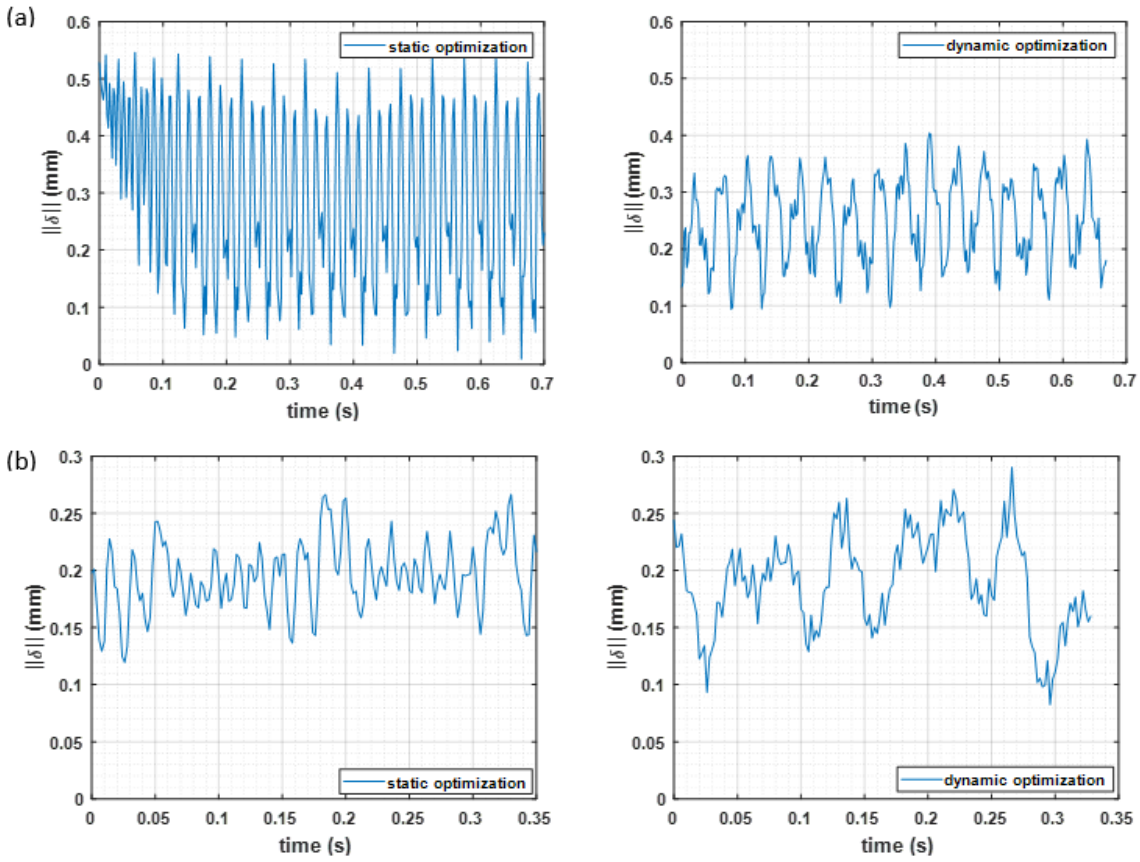

Figure 16 – Norm of the measured deflections for test numbers 5-8 for a tooth passing frequency of (a) 60.0 Hz and (b) 107.5 Hz. The lower frequency is within the range of the robot's natural frequencies at this pose while the higher one is not.

Test numbers 1-8 seem to confirm the hypothesis that a dynamic model-based optimization can outperform a static one when the milling force frequencies approach the natural frequencies of the robot (Figure 14a and Figure 16a). This is evident from the smaller 2-norm of the deflections seen in the dynamic-model based optimizations. However, for the high spindle speed tests (3,4,7,8), where the milling force frequencies were outside the range of the natural frequencies of the robot, the difference in mean measured amplitude of the end-effector deflection between the static and dynamic stiffness

optimizations was < 0.025 mm (Figure 14b and Figure 16b), which is less than the robot repeatability. Thus, for the high spindle case, neither model has a clear advantage.

For test numbers 9-10, the predicted natural frequencies of the robot varied by < 10 Hz over the range of possible A angles (Figure 17). This location in the workspace was chosen to compare the effectiveness of each pose optimization when the dynamic properties do not vary significantly with pose. The predicted and measured deflections are displayed in Figure 18 and Figure 19, respectively. The large magnitude of the measured deflections indicates that when the resonant modes of the robot cannot be avoided through pose optimization, both models struggle to prevent severe vibration.

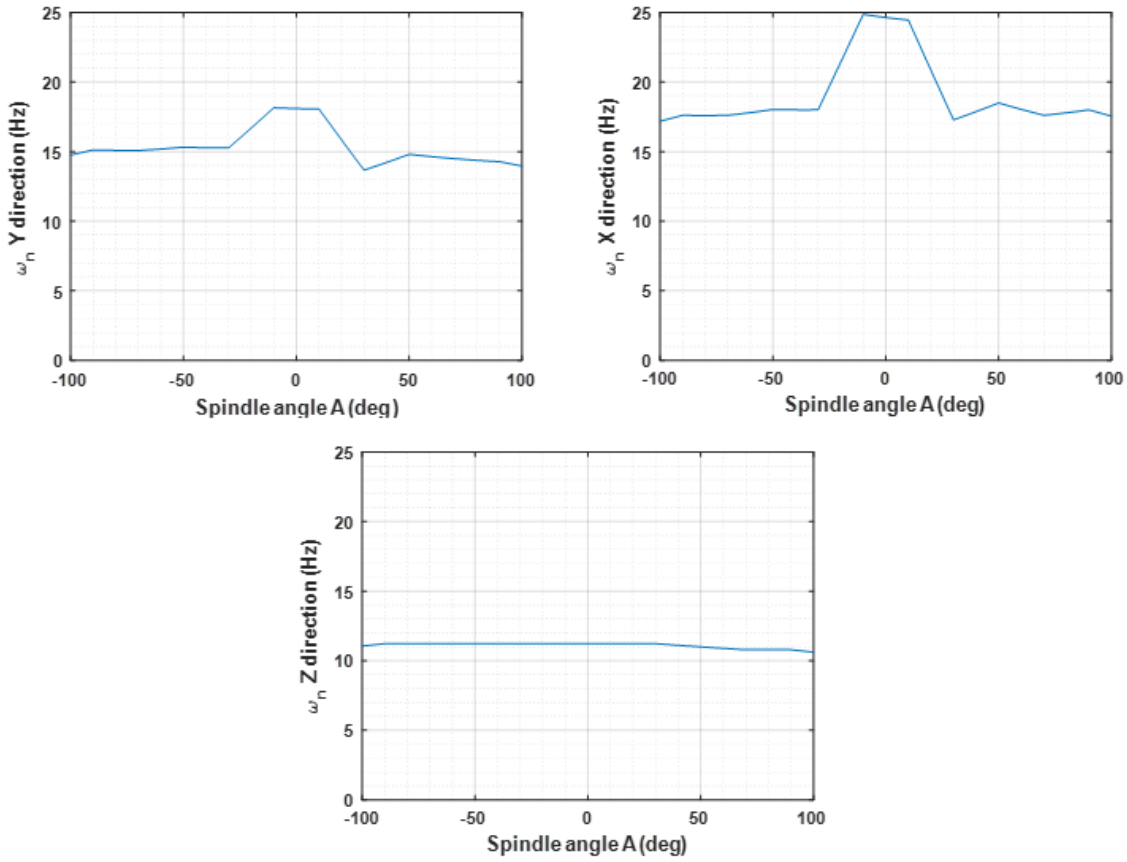

Figure 17 – Predicted natural frequency versus angle A for test numbers 9-10.

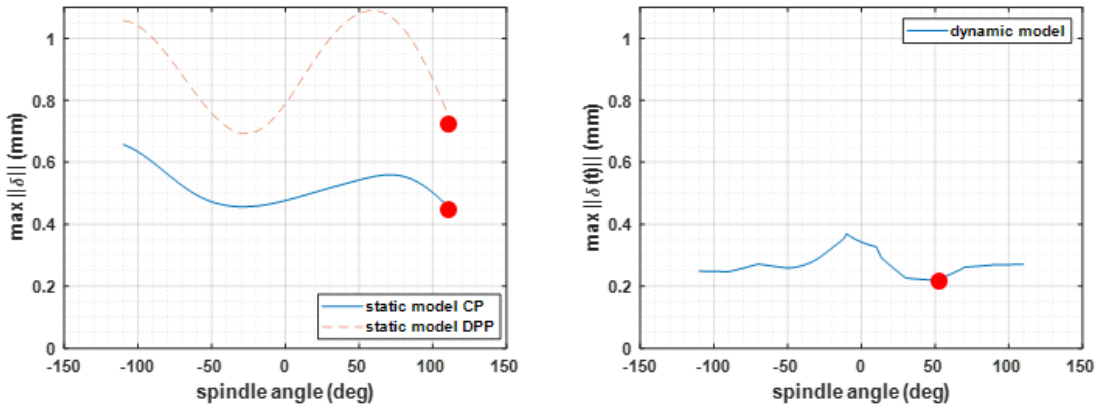

Figure 18 – Predicted deflection versus angle A for test numbers 9-10.

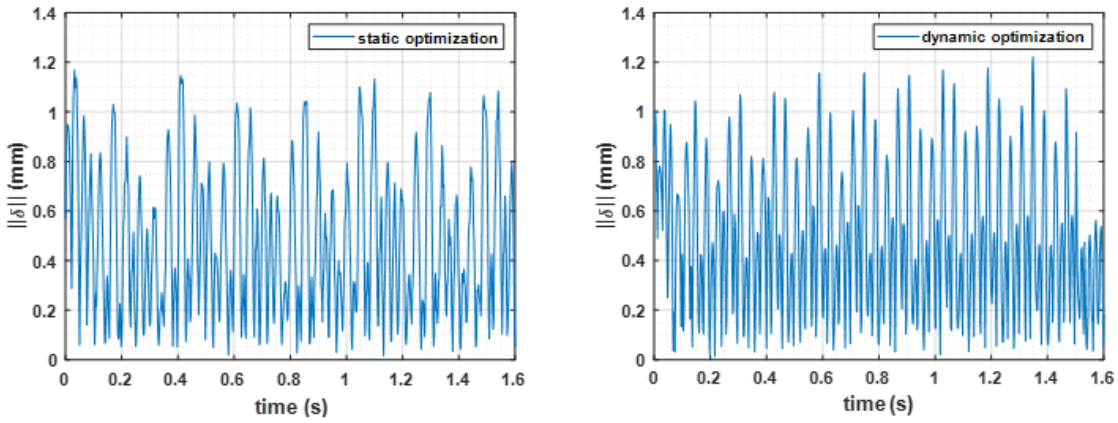

Figure 19 – Measured deflections for test numbers 9-10. The cutting force frequency is 25 Hz, which falls within the range of the robot's natural frequencies at this pose.

So far, the results shown have depicted cases where the static and dynamic model-based optimizations yielded significantly different solutions. However, this is not always the case, as can be seen for the location corresponding to test numbers 11-12 (Figure 20). The optimal pose solutions returned by the static and dynamic stiffness-based optimizations are a function of the workspace location and milling parameters and may or may not diverge depending on those factors.

A quantitative summary of the deflection results is presented in Table 6. If all test parameters other than whether a dynamic or static compliance model was used to perform the optimization are disregarded, dynamic model-based optimizations performed better with a mean deflection value of 0.56 mm compared to the static model's 0.65 mm. However, a t-test between the static and dynamic-based optimizations rejects the null hypothesis with a p-value of 0.285 corresponding to a 71.5% confidence. The large p-value reflects the nuance inherent in the benefits of using a static versus dynamic compliance model. In other words, factors like the sensitivity of the dynamics to the pose at a given

workspace location and the process parameters such as cutting force frequency are significant and should be considered in the comparison, as was demonstrated throughout this section.

Table 6 – Maximum Deflections by Test Number

Test No.	Optimization Approach	Max Deflection (mm)
1	Static	0.98
2	Dynamic	0.63
3	Static	0.27
4	Dynamic	0.26
5	Static	0.55
6	Dynamic	0.40
7	Static	0.26
8	Dynamic	0.28
9	Static	1.18
10	Dynamic	1.22

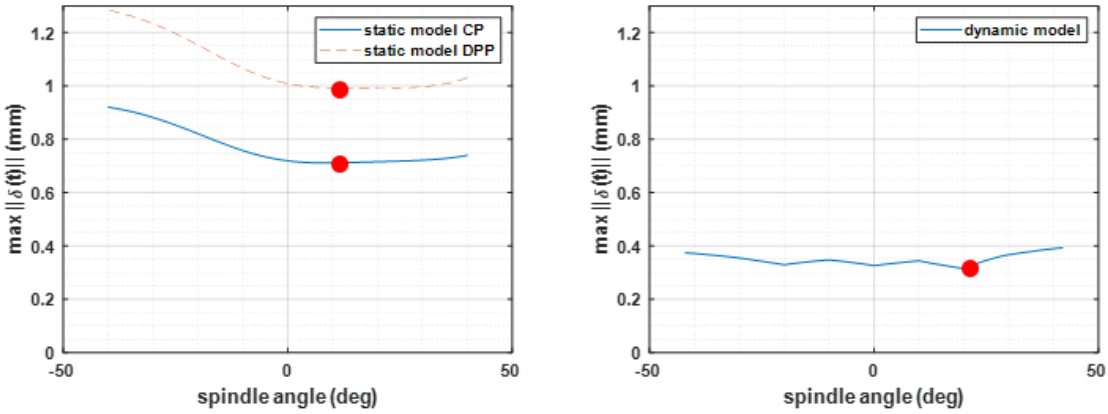

Figure 20 – Predicted deflections for test numbers 11-12.

3.4.5 Effect of Optimization on Machining Accuracy

While the focus of this work was on comparing the effectiveness of static and dynamic stiffness optimizations in minimizing deflections of the end-effector, the end-

application goal is to allow robotic machining to produce parts with better tolerances. Figure 21 displays a box and whisker plot of the measured errors in the axial and radial depths of cut. The errors were measured using a coordinate measurement machine (CMM) by measuring the depths of cut relative to a reference surface, then comparing these values to the nominal depths of cut. The "x" ticks denotate the mean values, the boxes indicate the densest quartile of measurements, and the vertical lines extending from the boxes show the variation in error for each measured cut surface. These plots correspond to test numbers 5-6, which were run with milling force frequencies within the range of the robot's natural frequencies at the location. The low axial depth of cut error is expected because the milling forces in the axial direction (Z) were low relative to the other directions. The radial depth of cut error plots confirm that the dynamic stiffness optimization outperforms the static optimization for the case that the natural frequencies of the robot approach those of the milling forces. The data also confirms that the laser tracker measurements of end-effector deflection correlate well with the geometric error in the finished part.

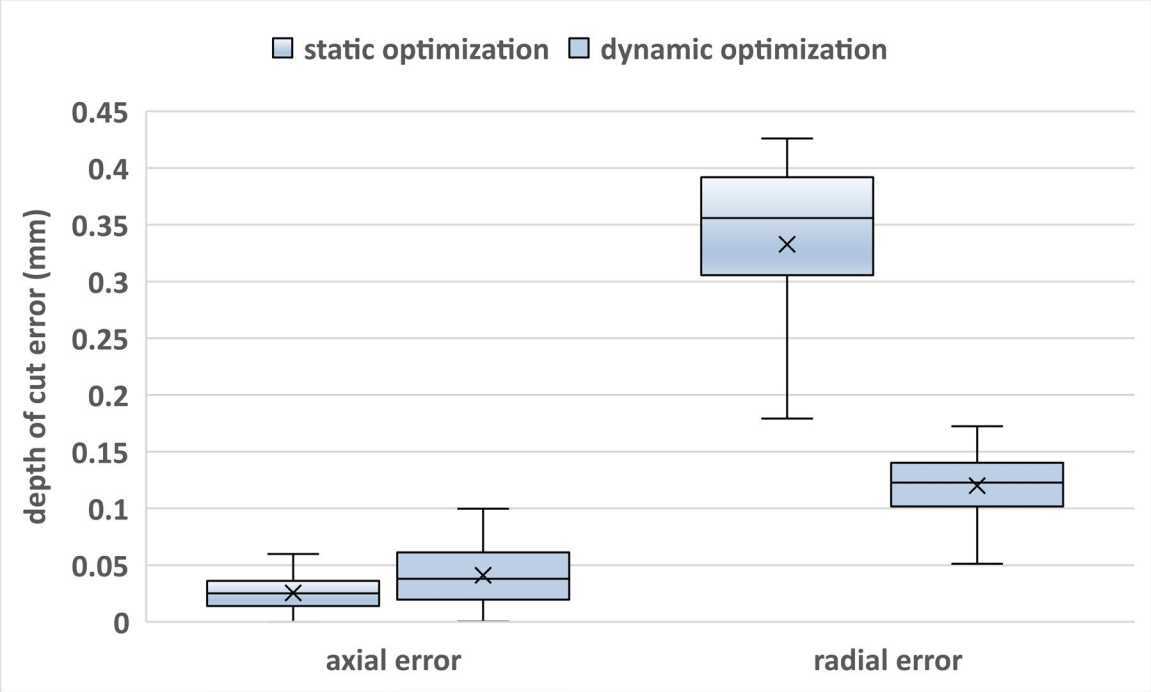

Figure 21 – Axial and radial depth of cut error measurements for test numbers 5-6.

3.5 Summary

This chapter compared the effectiveness of using static and dynamic models to select the optimal pose for robotic milling by minimizing the predicted deflections. Results indicate that a dynamic model-based optimization has more success reducing force-induced deflections when milling forces approach the resonant frequencies of the robot and when the task is in a location for which the robot can substantially alter its natural frequencies through a change in pose. For the other cases of when milling forces do not approach the robot natural frequencies or for workpiece locations where the range of available poses does not allow for a shift in the natural frequencies of the robot, there was no significant advantage in using the dynamic model to select the optimal pose over the static one.

Results also indicate that being able to control location of the workpiece in the robot's workspace and the cutting parameters can be a more powerful tool than pose optimization. As mentioned, the location in the workspace can determine what poses are available for the stiffness optimization to choose from, so locations should be chosen such that the range of achievable robot natural frequencies is large. Adjusting the cutting parameters, namely, the frequency of the cutting forces, is perhaps the most powerful tool one has in reducing deflections. If the spindle speed can be adjusted such that the tooth passing frequency is far from any of the resonant modes of the robot, then both a static and dynamic model-based optimization will produce the best results.

In practice, the results from this chapter indicate that a basic dynamic model of the robot is useful to have even if it is not used for pose optimization specifically. If the model can describe how the resonant frequencies of the robot vary with pose to an approximate level, it can be used to select for optimal workspace locations and cutting parameters for milling, which, according to the results of this chapter, can be even more important than selecting the optimal pose. The motivation for this chapter was to answer the question of what kind of model should one calibrate on their manipulator if they plan to use it for high-force processes with time-varying forces such as milling. An analysis of the deflections produced via different pose optimizations for different workspace locations and cutting parameters has shown that the best answer is an approximate (low-calibration effort) dynamic model. This model should be used to select the optimal workspace locations and cutting parameters to minimize deflections if that is an option. However, if the constraints of the task are such that cutting parameters and workspace location must be fixed, higher

resolution (more poses used in training) dynamic model should be calibrated to select the optimal poses for machining.

CHAPTER 4. STATE ESTIMATION OF A ROBOT END-EFFECTOR USING LASER TRACKER AND INERTIAL SENSOR FUSION

4.1 Introduction

The goal of this chapter is to quantify the benefits and limitations of supplementing real-time laser tracker data with inertial data from an Inertial Measurement Unit (IMU) to improve the accuracy of the overall state estimation of the robot end-effector. Data are fused using an Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filter (PF). Purely translational paths and paths with both translation and rotation are considered to assess the effectiveness of the three fusion techniques for the linear (former) and nonlinear (latter) case. In addition, the effects of sensor bias and noise characteristics are investigated to determine the limits of sensor performance beyond which data fusion provides no substantial benefit. Experiments are conducted first in simulation, then on hardware using an industrial manipulator.

4.2 Methods

4.2.1 Modeling the State

The position, orientation, and their first and second derivatives comprise the full state of the robot end-effector to be estimated. The translational part of the state is modelled as:

$$x_{k+1} = x_k + \dot{x}_k \Delta t + \ddot{x}_k \Delta t^2 \tag{21}$$

$$\dot{x}_{k+1} = \dot{x}_k + \ddot{x}_k \Delta t \tag{22}$$

$$\ddot{x}_{k+1} = \ddot{x}_k \tag{23}$$

Here, x is a 3-vector representing the 3D Cartesian position of the end-effector, subscript k represents the sample number, and Δt indicates the sample time. This model assumes that the sensor rate is high enough that acceleration change between samples can be approximated as zero. The rotational part of the state is modelled as:

$$q_{k+1} = q_k + \dot{q}_k \Delta t + \ddot{q}_k \Delta t^2 \tag{24}$$

$$\dot{q}_{k+1} = \dot{q}_k + \ddot{q}_k \Delta t \tag{25}$$

$$\ddot{q}_{k+1} = \ddot{q}_k \tag{26}$$

Here, the orientation is represented as a quaternion, q, whose derivatives are expressed in terms of the orientation, angular velocity, and angular acceleration as:

$$\dot{q} = \frac{1}{2} \omega q \tag{27}$$

$$\ddot{q} = \left(-\frac{1}{4} \|\omega\|^2 + \frac{1}{2} \dot{\omega} \right) q \tag{28}$$

The angular velocity term, ω, is a quaternion with real part equal to zero and imaginary part equal to the three Cartesian components of the angular velocity of the end-effector. Note that all non-scalar multiplications in (27) and (28) are quaternion multiplications. Like for the translational equations, the assumption is made that the sensor rate is high enough that angular acceleration changes between samples can be approximated as zero. The full state is thus 19 dimensional: position, velocity, acceleration, angular velocity, and angular acceleration have three components each, and orientation has four components because of the quaternion representation.

4.2.2 Experimental Setup: Simulation

To quantify the accuracy of the state estimation, the ground truth state should be known to within at least an order of magnitude better than what the sensors used in the estimation can measure. Because laser trackers can achieve position measurement accuracies of less than 0.015 mm, it is difficult to find a sensor which is both ten times as accurate with the same distance range (~1 m) of a laser tracker. Because of this, experiments are first set up in simulation using MATLAB Simulink software so that the ground truth state is known at all times to within machine precision.

End-effector test trajectories were selected to be sine waves in space with a constant tangential velocity, or "feed rate". Purely translational motions are referred to as "3-axis" moves while motions with translation and rotation are referred to as "5-axis" moves because there is no rotation about the axis of the tool (Figure 22). Sine waves are used because they include both close-to-linear motions with almost no acceleration between the troughs and peaks and highly nonlinear motions with high accelerations at the troughs and

peaks. 5-axis motions are used to incorporate rotation because they are representative of most industrial robotic operations that require high accuracy in the end-effector state estimation (e.g., painting, milling, drilling, and welding). The feed rate is kept constant because this is the norm for most milling and painting tasks and because it allows it to be an easily adjusted parameter to scale the linear and angular accelerations that the end-effector undergoes for a given trajectory. The sine waves were of 0.0762 m amplitude with a spatial frequency of 19.685 cycles/m. This shape was selected to provide a good scaling of linear and angular accelerations with feed rate.

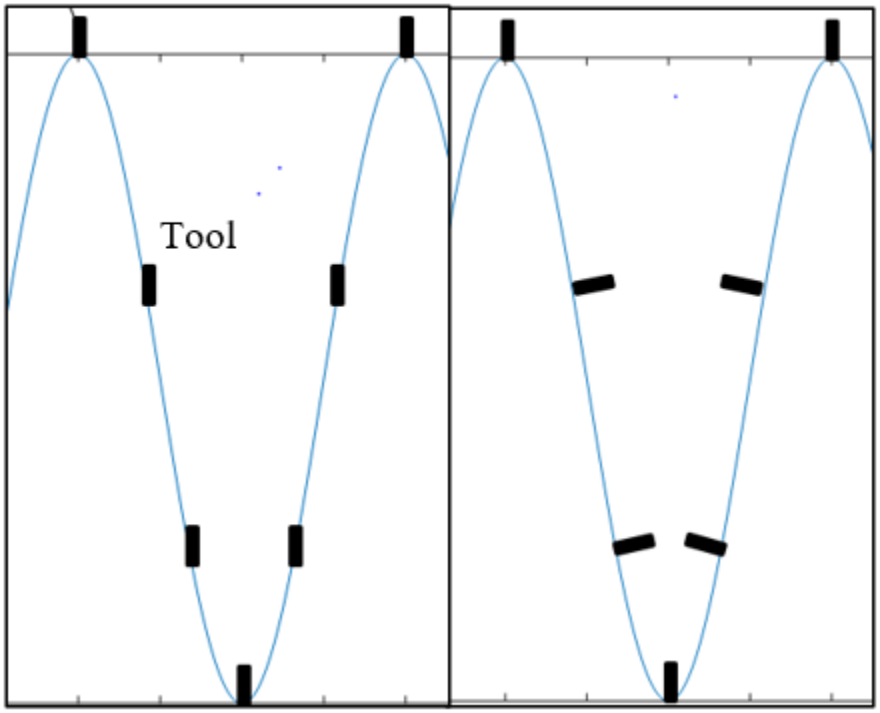

Figure 22 – Sample tool motion for a 3-axis motion (left) and 5-axis motion (right). For the 5-axis motion, the tool axis is constrained to remain normal to the path.

The laser tracker measures position and orientation for the state estimation while the IMU measures acceleration and angular velocity. Both sensors are set to measure at 1000 Hz, so no differentiation or integration of measurements is necessary between sensor

cycles. For 3-axis motions, the state equations are linear, and data are fused using a Kalman Filter (KF) and Particle Filter (PF) [99]. The process and measurement noise covariance matrices (Q and R, respectively) for the KF are based on the accuracy of the sensors and on which parts of the state are directly measurable (see Appendix B for details). The KF, Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and PF all use the state transition equations (21-28) combined with sensor measurements to output estimates of the state at each sample time. These estimators are used because they are the norm in the recent literature when fusing inertial data with other position sensors, and because they represent a tradeoff between speed and robustness to non-linear systems. Kalman Filters and their variants are extremely fast but can struggle to capture non-linear behaviors. The Extended Kalman Filter only relies on a first order Taylor series expansion to linearize the system, while the Unscented Kalman Filter improves the robustness to non-linearity by linearizing around "sigma" points which can better capture the shape of a non-linear motion [100]. The UKF thus improves the accuracy from a 1^{st} order Taylor series expansion to a 3^{rd} order one for nonlinear systems. Particle Filters, on the other hand, are the most robust to nonlinearities but also the most computationally expensive. They do not rely on any linearization, but instead propagate a large number of possible states, called particles, through the state transition equations to form the initial estimate. The propagated particles are then resampled using a weighted sampling which favors particles whose state estimates are closer to the measurement data provided by the sensors [99]. The computational expense grows with the number of particles used, as does the robustness of the estimation to nonlinearity. For the hardware used in this study, a value of 1000 particles was the limit beyond which real-time estimation could no longer be achieved with the PF.

A range of sensor parameters is used in quantifying the accuracy of the state estimation. The laser tracker and IMU measurements are corrupted by varying levels of Gaussian white noise and the IMU measurements are additionally offset by biases of varying size. Also, because the tracker target and IMU location cannot physically occupy the same space, they are separated by a six-dimensional coordinate transformation. This transformation is chosen to represent a real end-effector present in the laboratory with translational part equal to [-180, -100, 120] mm and rotational part approximately equal to a 180 degree rotation about the *y* axis (Figure 7 for reference). For reference, the sine waves are oriented such that *z* is a sinusoidal function of *x* (Figure 22). To explore the effects of sensor transform error, this transform is corrupted by varying levels of error. Both translational and rotational transform errors are applied to the z axis to maximize their effects on estimation accuracy. For a rotating rigid body, two points on the body perceive acceleration differently. Thus, the acceleration of the tracker target relative to that of the IMU is given as:

$$a_{target} = a_{IMU} + \alpha \times r + \omega \times (\omega \times r) \tag{29}$$

Here, a indicates acceleration, α and ω are the angular acceleration and velocity of the end-effector, respectively, and r is the position vector from the IMU to the tracker target. For the experiments, the position of interest for the state estimation is located on the tracker target, so the acceleration and angular velocity measured by the IMU are transformed according to (29).

The baseline and offset parameters for the simulation experiments are given in Table 7. The sensor bias and noise parameters are modeled after a Leica AT960 laser

tracker measuring a Leica T-Mac target and a Gladiator Technologies Landmark 60 IMU. Most parameters are taken from the sensors' technical specifications except for those of the orientation measurement accuracy of the T-Mac, which is taken from [101]. Noise values are represented as standard deviations of Gaussian white noise. These sensors are selected because they are both top of the line in terms of accuracy characteristics and data rates, and so are good "best case" representations of sensors for the state estimation. The baseline sensor fusion technique is a KF for 3-axis motions and an EKF for 5-axis motions because they represent the lowest computational cost for each motion type. The baseline feed rate is selected to match typical feed rates for machining aerospace materials such as aluminum and carbon fiber. Finally, sensor transform errors are set to zero for baseline, representing a perfect calibration of the various coordinate frames, then increased until the error is large enough that the sensor fusion no longer provides a benefit to the overall state estimation.

Table 7 – Simulation Experiment Parameters and Values

Parameter	Baseline	Offset
Position sensor noise (mm)	0.0064	0.064, 0.64
Orientation sensor noise (deg.)	0.015	0.15, 1.5
Angular velocity sensor bias (deg./s)	0.0008	0.008, 0.08, 0.8, 8
Angular velocity sensor noise (deg./s/√Hz)	0.0016	0.016, 0.16
Acceleration sensor bias (m/s^2)	0.00025	0.0025, 0.025
Acceleration sensor noise ($m/s^2/\sqrt{Hz}$)	0.00039	0.0039, 0.039
State estimator	KF/EKF	PF
Sample time (s)	0.001	0.002, 0.01
Feed rate (mm/s)	42.33	21.17, 4.233
Sensor transform translation error (mm)	0.000	0.254, 2.54
Sensor transform orientation error (deg.)	0	0.1, 1, 10

4.2.3 Experimental Setup: Hardware

Simulation results, discussed later, showed that the largest benefit of the sensor fusion was in the velocity estimation. To confirm the results, hardware experiments were set up using the same instruments as in the simulation. The robot used was a Kuka KR 500-3 serial manipulator which can report its pose at a rate of 250 Hz. Motion experiments indicated that the robot has an average global positioning error of 0.5 mm, which is consistent with findings for other industrial robots [8, 11, 13, 15, 17, 102-104]. This allows the Leica AT960 measurement to be used as ground truth to assess the effectiveness of fusing the reported robot pose data with IMU data.

Because the feed rate cannot be specified for this robot for a sinusoidal motion, experimental trajectories were made to be linear and circular moves, which capture the

same linear and nonlinear shapes of a sine wave. Experiments were conducted at three different feed rates for both 3-axis and 5-axis motions (Table 11). Linear motions were 1 m in length while circular motions had the robot complete a semicircle of radius 0.25 m. For the 5-axis motions, the end-effector rotated 60 degrees from the beginning to the end of the motion (Figure 23). This rotation was selected to be as high as possible without exceeding the angular acceleration limits of the robot joints to complete the motion.

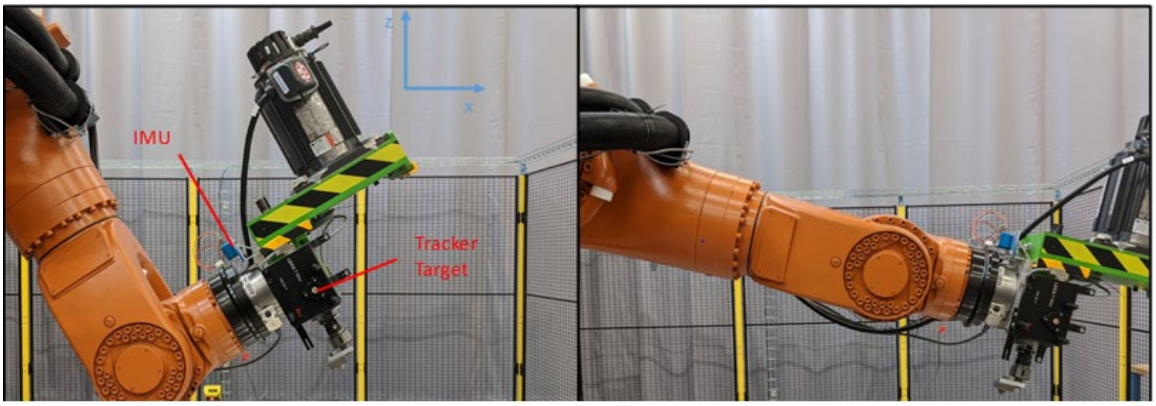

Figure 23 – Example 5-axis motion. The end-effector rotates by 60 degrees from the pose pictured on the left to the pose pictured on the right.

4.3 Results

4.3.1 Estimation of Measured Quantities in Simulation

For the baseline experiment parameters, state estimation of the measurable parts of the state (position, orientation, acceleration, and angular velocity) exhibited marginal to no improvement with sensor fusion (Table 8). If a sensor is made less accurate than the baseline, however, fusion does improve the estimation accuracy of the parts of the state directly measured by the less accurate sensor (

Table 9). This is analogous to the findings of [86] where the position measurement made by an optical measurement system, which is less accurate than a laser tracker, was improved by fusion with acceleration measurements. A comparison of the mean position, acceleration, and angular velocity estimation errors between the first two columns of Table 8 and the last two columns of Table 9 indicates that though fusion with a less accurate sensor improves the accuracy of the quantity measured by that sensor, it still cannot achieve the same accuracy as when using a better sensor alone.

Table 8 – Baseline Estimation Performance

Parameter	**Tracker/IMU alone**		**Sensor fusion**	
	3-axis	***5-axis***	***3-axis***	***5-axis***
Position estimation mean error (mm)	0.0089	0.0089	0.0089	0.0089
Position estimation std. error (mm)	0.0066	0.0066	0.0066	0.0066
Orientation estimation mean error (deg.)	N/A	0.012	N/A	0.11
Orientation estimation std.error (deg.)	N/A	0.0092	N/A	0.17
Acceleration estimation mean error (m/s^2)	0.00018	0.00061	0.00012	0.00040
Acceleration estimation std error (m/s^2)	0.00013	0.0010	0.00010	0.00010
Angular velocity estimation mean error (deg/s)	N/A	0.070	N/A	0.062
Amgular velocity estimation std error (deg/s)	N/A	0.054	N/A	0.046

Table 9 – Benefits of Fusion for Less Accurate Sensors

Offset sensor parameter	**Value**	**Mean error (sensor only)**		**Mean error (sensor fusion)**	
		3-axis	***5-axis***	***3-axis***	***5-axis***
Tracker noise (mm)	0.64	0.889	0.889	0.41	0.41
Acceleration sensor noise (m/s^2)	0.0039	0.0018	0.0020	0.00048	0.00049
Angular velocity sensor noise	0.16 deg/s/√Hz	N/A	0.70 deg/s	N/A	0.35 deg/s

4.3.2 Velocity and Angular Acceleration Estimation in Simulation

The simulation experiments revealed that velocity estimation sees the biggest benefit in the full state estimation from the fusion of the tracker and IMU data (Figure 24 and Figure 25). These plots are over a five second simulation window for the baseline experimental parameters listed in Table 7. For estimating velocity using only laser tracker measurements, a four-point linear regression was used to fit a slope to the last four position measurements. This reduces some of the noise amplification which would result from a direct differentiation of the tracker measurements at the cost of filtering out some higher frequency changes in the velocity. The spikes in error seen around 1.8 s correspond to a peak in the sine wave trajectory, which is where linear and angular accelerations are

highest. In the 5-axis motion, this peak represents the most nonlinear behavior in the state because it corresponds to the most rapid orientation change.

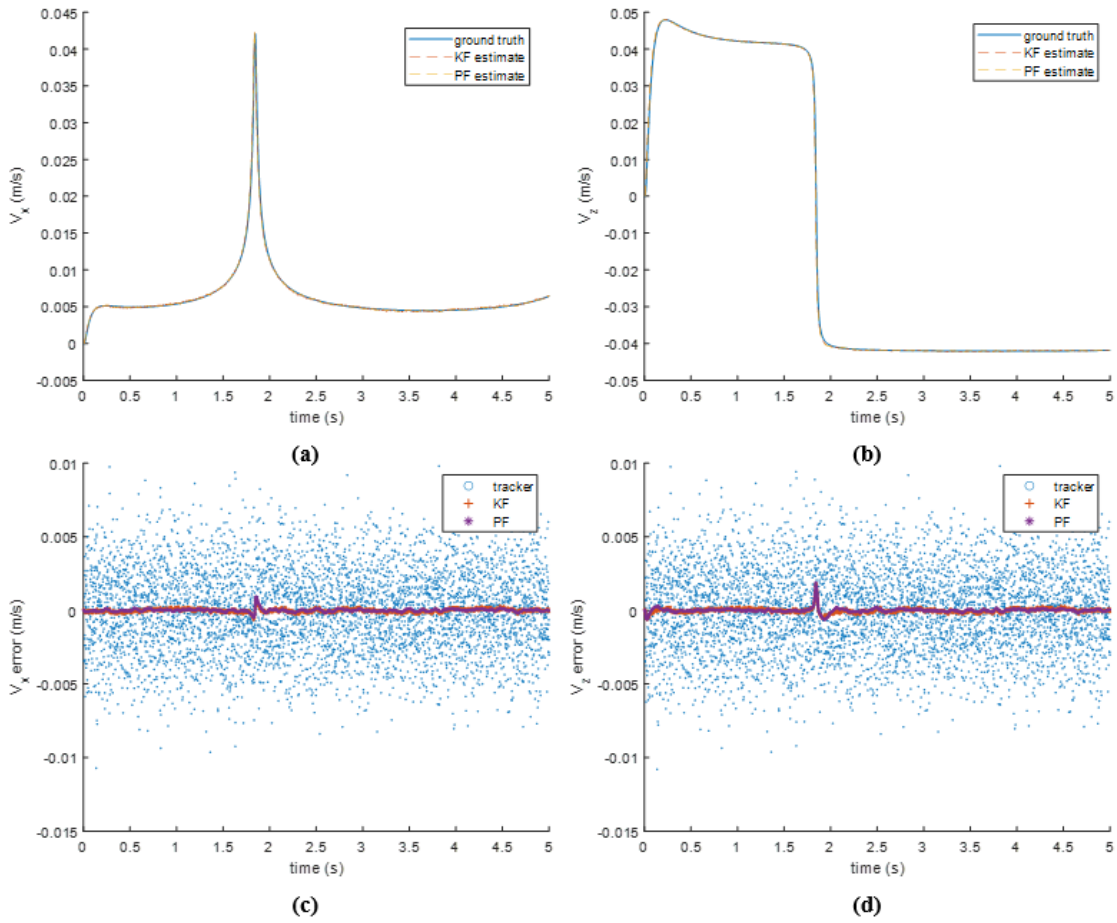

Figure 24 – Velocity estimation simulation results for the 3-axis experiments. (a) and (b) are the x and z velocity estimations plotted with ground truth, respectively, and (c) and (d) are the respective velocity estimation error plots.

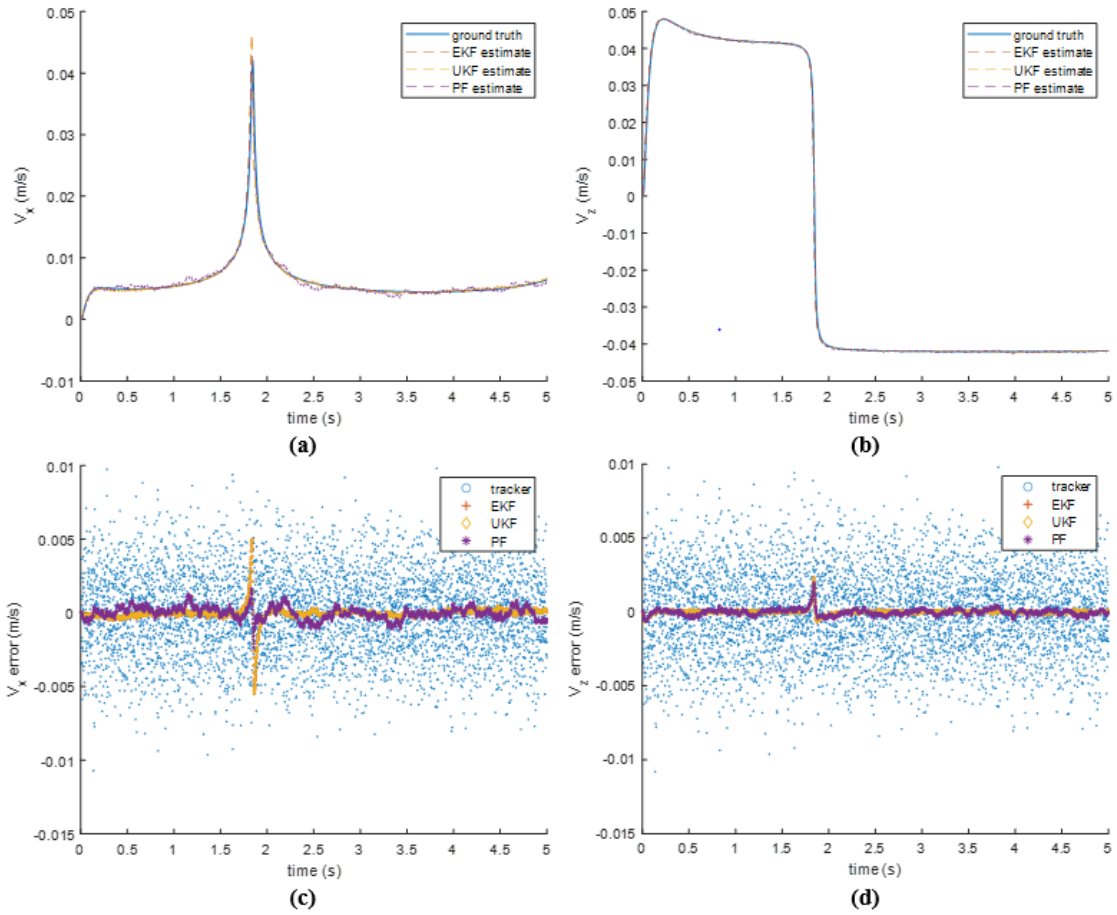

Figure 25 – Velocity estimation simulation results for the 5-axis experiments. (a) and (b) are the x and z velocity estimations plotted with ground truth, respectively, and (c) and (d) are the respective velocity estimation error plots.

The velocity estimation results indicate that for the linear and close-to-linear parts of systems (like the 5-axis motion in between sine wave peaks), a KF (for 3-axis) or EKF/UKF (5-axis) performs a better estimation than a 1000 particle PF. However, the advantage of the PF becomes clear for the highly nonlinear orientation change around the peak of the sine wave, where its estimation is more accurate than that of both the EKF and UKF.

Angular acceleration estimation accuracy shows about a 45% improvement for baseline experiment parameters everywhere except at the peaks of the sine wave (Figure 26). At the peaks, the fusion performs worse than the IMU alone because of the inability of the EKF and UKF to handle the highly nonlinear rapid rotation. However, the EKF and UKF still outperformed the PF in the estimation, most likely because more than 1000 particles would be needed for the PF to accurately capture the nonlinearity in the motion. The improvement in the angular acceleration estimation with sensor fusion shown in these plots is consistent through all the simulation experimental conditions except for increased sample time, in which case fusion does not provide an improvement in the estimation. The biggest reason for estimating the angular acceleration, however, is because of its influence in the perceived linear acceleration by the IMU as detailed in equation (29). Noting this, the angular acceleration estimation needs only to be accurate enough so that it does not corrupt the linear acceleration estimation so much that the velocity can no longer be accurately estimated with fusion.

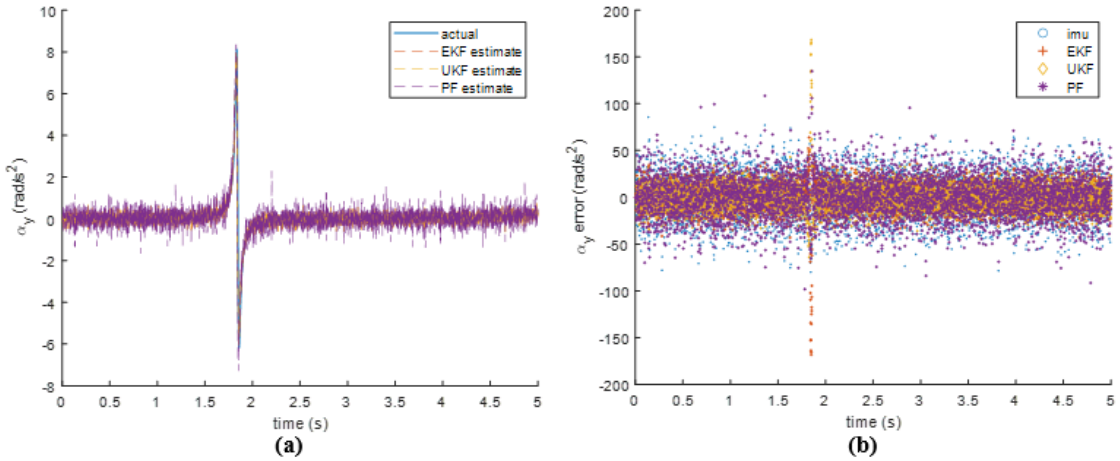

Figure 26 – Y angular acceleration estimation simulation results for the 5-axis experiments. (a) plots the estimation and ground truth and (b) plots the estimation error.

4.3.3 Sensitivity Analysis

To check the requirements necessary for the hardware used to still provide state estimation benefits with fusion, a sensitivity analysis was performed by varying the bias and noise of the sensors, sensor-to-sensor transform calibration error, and sample time. Table 10 shows the minimum requirements in these parameters for the velocity estimation to be better with fusion than by using the tracker alone. The velocity estimation was chosen as the component of the state estimation to compare because it improved the most with fusion for baseline conditions. Each value in Table 10 represents the limit for that parameter assuming all other parameters are baseline. The 5-axis case has a higher sensitivity to the sensor parameters because of the relationship between the IMU and tracker target accelerations. Note that translational sensor-to-sensor transform error does not impact the 3-axis estimation because in that case, both the IMU and laser tracker target perceive the same motion. The sample time limit is largely due to the assumptions made in

the state model that linear and angular accelerations remain constant between sample times. For a larger sample time, this assumption becomes less and less accurate, and so the algorithms used to fuse the IMU and tracker data, which rely on a good state transition model, become inaccurate. Finally, for the range of feed rates tested, the benefits of fusion did not change substantially, so feed rate is not listed as a parameter in the table.

Table 10 – Minimum Requirements for Sensor Fusion Benefits

Parameter	3-axis	5-axis
Acceleration sensor noise (m/s^2)	< 0.1	< 0.1
Acceleration sensor bias (m/s^2)	< 0.001	< 0.001
Angular velocity sensor noise (deg./s)	N/A	< 0.1
Angular velocity sensor bias (deg./s)	N/A	< 1
Sensor transform orientation error (deg.)	< 10	< 0.1
Sensor transform translation error (m)	No constraint	< 0.1
Sample time (s)	< 0.01	< 0.01

4.3.4 Hardware Results

The velocity estimation results for the motions performed on the real hardware can be seen in Table 11. In each case, fusion of the IMU and robot Cartesian data measurements exhibited significant improvement in the velocity estimation accuracy over using the robot pose data alone. Specifically, the mean error was reduced by up to 37% for 3-axis motions and up to 35% for 5-axis motions. The results are consistent with the simulation results in

that the estimation accuracy is improved by sensor fusion even though the robot joint encoder is a noisier sensor than the laser tracker.

Table 11 – Hardware Results

	No End-Effector Rotation						**Continuous End-Effector Rotation**					
Motion Type	**Linear Move**			**Semicircular Move**			**Linear Move**			**Semicircular Move**		
Feed rate (mm/s)	21.17	42.33	63.50	21.17	42.33	63.50	21.17	42.33	63.50	21.17	42.33	63.50
Robot Mean Error (mm/s)	0.93	0.80	0.93	0.93	2.03	0.97	0.80	0.97	1.10	0.85	1.02	1.23
Fusion Mean Error (mm/s)	0.76	0.51	0.51	0.64	1.27	0.64	0.64	0.64	0.72	0.64	0.72	0.80
Robot Std Error (mm/s)	0.93	1.27	2.07	0.85	2.33	1.74	0.72	1.61	1.99	0.85	1.52	2.33
Fusion Std Error (mm/s)	0.55	0.34	0.34	0.34	1.35	0.38	0.42	0.42	0.55	0.38	0.47	0.59

4.4 Summary

An investigation into sensor fusion of inertial measurements with laser tracker measurements revealed that velocity estimation can be significantly improved compared to what can be estimated using a laser tracker alone. Detailed results were acquired through simulation experiments, then the velocity estimation was performed on real hardware, where it was shown that the estimation can be improved via fusion with IMU data even when using robot data for the position measurement instead of a laser tracker. Finally, sensor bias and noise characteristics, transform error, and sample rates were varied to determine the quality of sensors and sensor calibrations needed to achieve benefits from fusion.

The main question motivating this chapter of whether the state estimation can be improved at all by adding inertial data to laser tracker measurements has been answered affirmatively for the case of the velocity and angular acceleration measurements. Because of the low cost of an IMU compared to a laser tracker, this makes it a lucrative investment for applications where the best possible velocity estimation is desired. The current state-of-the-art for accurate velocity measurement is dominated by high speed cameras and short-range laser displacement sensors. However, these kinds of sensors have a limited range and so are not practical for use in applications with the volumetric scale such as those performed by industrial robots. Thus, the fusion of laser tracker and IMU data should be the best strategy to obtain the highest possible accuracy in velocity estimation of an industrial robotic end-effector.

CHAPTER 5. A NEW METHOD FOR CLOSED-LOOP STABILITY PREDICTION IN INDUSTRIAL ROBOTS

5.1 Introduction

This chapter addresses a major hurdle that comes with controlling industrial robots in real-time via closed-loop feedback from external sensors, which concerns the difficulty in modeling the closed-loop system when input joint motor torques are not known. Because real-time commands in industrial robots come in the form of Cartesian or joint position offsets, the closed-loop system is commonly modeled using a naïve assumption that each real-time command is executed by the robot controller within the system cycle time with minimal to no error. In practice, this naïve assumption does not hold due to uncaptured sources of error present in each commanded motion, and so without a good closed-loop model of the system, it is necessary to manually tune feedback controller gains through trial-and-error. In this chapter, these unknown major sources of error in robot motion from real-time commands are identified and a new data-driven method of modeling them is presented. Then, it is demonstrated how the new model can be used to tune feedback controllers offline in simulation rather than manually through trial-and-error experimentation with the robot.

5.2 Approach

The approach used to achieve stability prediction in this chapter is as follows:

1. Test potential sources of error for real-time commanded motion
    a. Controller delay

b. Command magnitude

c. Commanded acceleration

d. Jerk-induced dynamics

2. Identify the major error sources and construct a model which allows them to be used for closed-loop stability prediction.
3. Evaluate the model for experimental closed-loop trajectories.

5.3 Methods

5.3.1 Hardware Setup

As in the other chapters, experiments were conducted using a Kuka KR 500-3 serial manipulator controlled by a Kuka KR C4 controller. A Leica AT960 laser tracker was used to measure end-effector motions. Time-stamped laser tracker measurements were transmitted via EtherCAT (EtherCAT Technology Group) to a TwinCAT software program (Beckhoff Automation), then again transmitted via UDP (User Datagram Protocol) to a Python server communicating at 1000 Hz. This server simultaneously communicated with the KR C4 controller via UDP using the Kuka Robot Sensor Interface (RSI) at 250 Hz. Figure 27 illustrates the hardware diagram corresponding to the real hardware pictured in Figure 28.

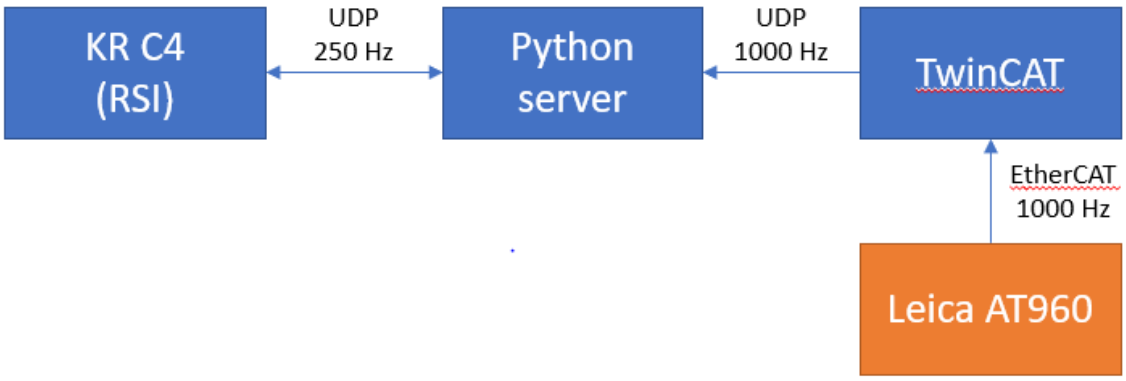

Figure 27 – Hardware diagram illustrating communication between the servers (blue) and measuring instruments (orange).

5.3.2 Controller Delay

As is the case for all real digital systems, there is delay present in the closed-loop system studied here. Delay being a well-understood factor in predicting closed-loop stability, it is necessary to have some understanding of it for a given system. Here, the delay is captured by conducting step response experiments. The amount of time between the input command and the beginning of the measured output motion is defined as the controller delay (Figure 29). The laser tracker timestamps were used to capture this delay, which was estimated to be 0.052 seconds.

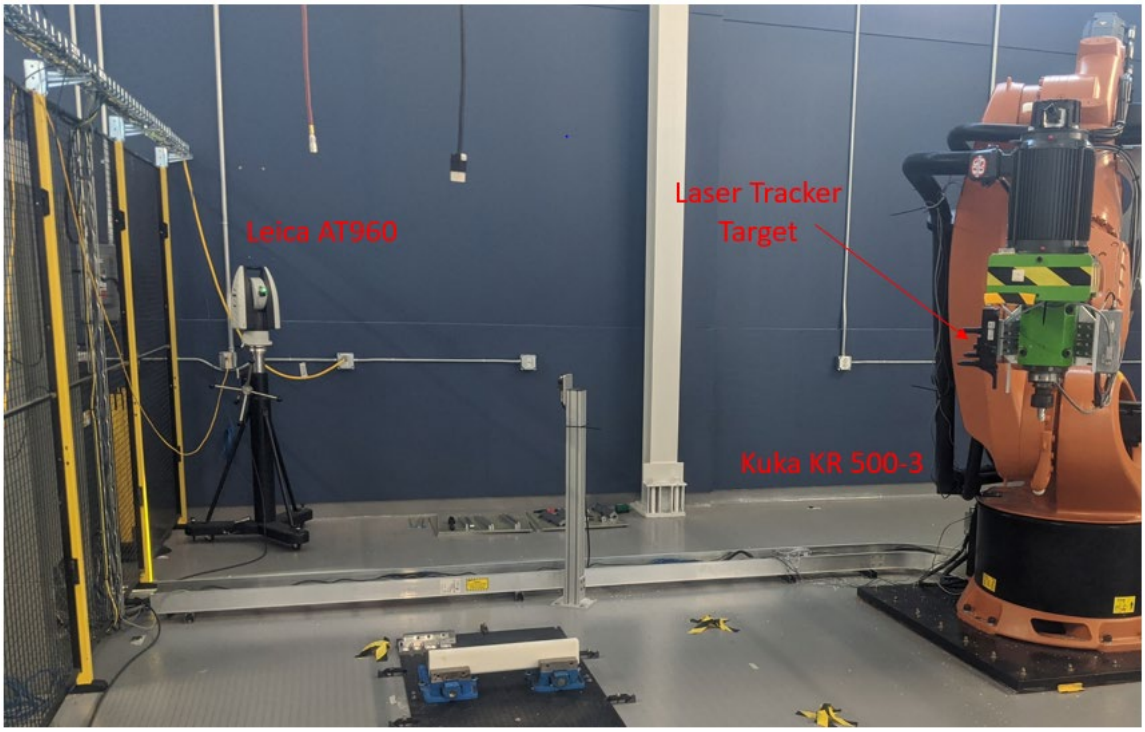

Figure 28 –Hardware setup.

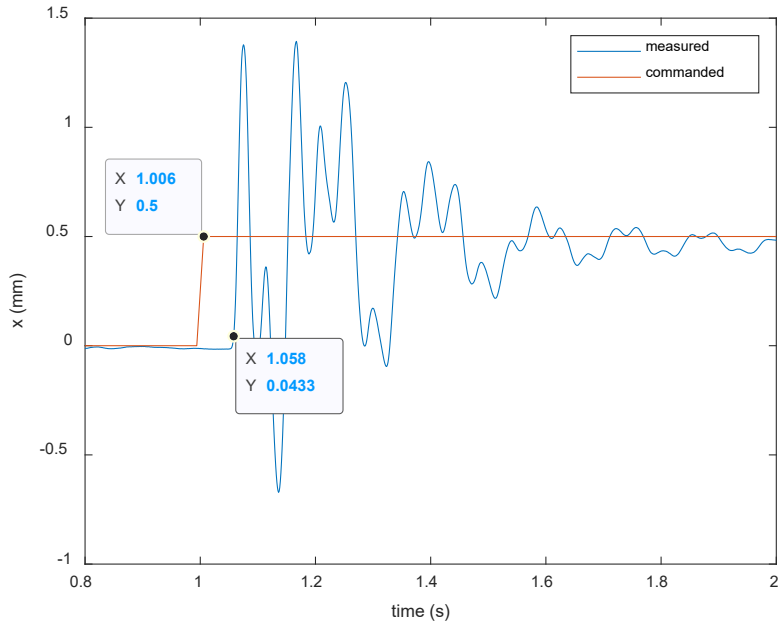

Figure 29 – Representative step response of the Kuka KR 500-3. The delay is computed as the time between the input command and the start of the initial measured motion of the end-effector.

5.3.3 Error Sources for Motion from Real-Time Commands

The first potential factors contributing to real-time commanded motion error investigated were command magnitude and path acceleration. It was initially thought that a given motion would have error that was proportional to the magnitude of its corresponding command. It was also hypothesized that the robot controller would struggle to achieve high end-effector accelerations because of the limitations in the torque outputs of the motors and so motions with high acceleration would result in higher command-induced error.

Command magnitude was ruled out as a significant contributing factor to motion error via constant velocity linear motion experiments. In these experiments, the robot was sent a Cartesian relative motion command every 12 ms, which means that it was commanded to offset its current Cartesian position by the commanded amount. For constant velocity motion, the subsequent input commands were identical, each offsetting the robot Cartesian position by the same amount over the 12 ms cycle. For the set of velocities of 20, 40, 80, and 120 mm/s, chosen to represent typical machining feed rates for hard and soft materials, the measured average command error was 0.02 mm in every case. This indicates that despite velocity commands having a higher magnitude, their associated error did not grow proportionally as hypothesized.

To examine the effect of path acceleration, circular motion experiments were used. By fixing the radius of the circle and changing the tangential velocity (feed rate), it is straightforward to apply a constant magnitude acceleration to the end-effector towards the center of the circle. If the hypothesis holds true, the robot should struggle to maintain the

inward acceleration necessary to stay on the circular path for higher tangential velocities and so the resulting circles should be larger than nominal. However, experimental results show that this is not what happens in practice (Figure 30). There is higher error at the higher feed rates, but the error source is unclear. It should be noted that the circular trajectories were achieved by a set of many small linear motions, which means that faster trajectories have fewer commands per unit path length, which could be a source of the error seen at the higher feed rates.

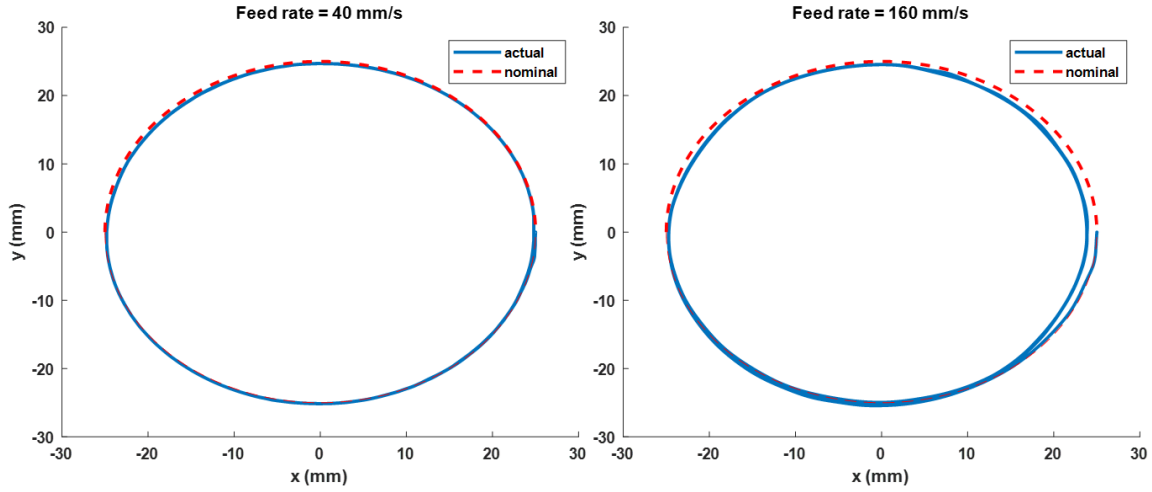

Figure 30 – Open-loop circular path experiments. Motion starts at the right side of the circle and moves clockwise. The expected result, that higher feed rates would produce larger circles, did not occur.

With command magnitude ruled out and path acceleration having an unclear relationship with motion error, it was necessary to formulate a new hypothesis for the major error sources. A close look at the robot response to individual commands, such as in Figure 29, reveals that there is a significant amount of residual vibration from the high-jerk accelerations involved in the sudden start and stop of the end-effector motion. A high-jerk acceleration must be induced by a large impulse force, which, for a second order system

such as described in (8), results in a large vibration. This motivated the hypothesis that structural vibrations caused by high-jerk accelerations of the end-effector are the biggest source of error in real-time commanded motion. To confirm this hypothesis, experimental modal analysis using an impact hammer was conducted at the pose whose step response is pictured in Figure 29. The tests yielded a frequency response function, which was then compared to the frequency content of the vibration data for the single-command robot experiment (Figure 31). The 17 Hz peak in Figure 31a is in the range of natural frequencies of the robot captured by the impact hammer test in Figure 31b (16-22 Hz). The results confirm that the resonant modes of the robot dominate the robot vibration response from the single-command experiment. Thus, structural vibration induced by motion commands is identified as a major error source when controlling industrial manipulators in real-time.

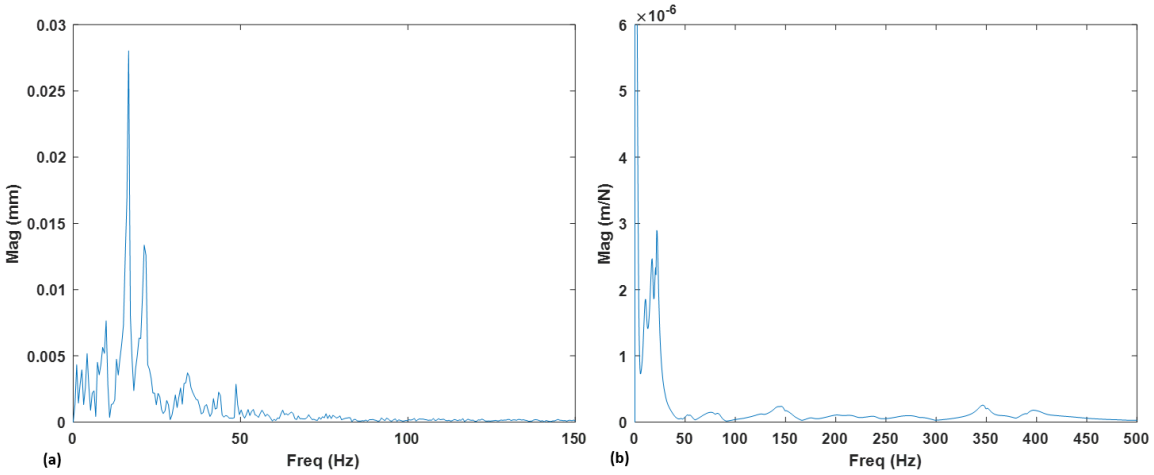

Figure 31 – Example frequency response of the robot measured from (a) command-induced vibration and (b) impact hammer experiment. The peak in (a) is at 17 Hz and is in the vicinity of the natural frequencies of the robot measured in (b). The y axis units reflect a measured vibrational response in (a) and the frequency response function in (b).

5.3.4 The Equivalent Force Model

Traditional dynamic models present an input/output relationship between forces/torques and motion of the system. This makes it difficult to apply them for real-time control of industrial robots, because the real-time commands in industrial robots come in the form of position or joint offsets for which the associated forces/torques are unknown. To address this, real-time commands are modelled by their "equivalent force" on the end-effector as follows:

$$F_{eq} = \frac{command}{cycle\ time^2} \tag{30}$$

This formulation assumes that the end-effector undergoes a constant acceleration for the duration of the command cycle time. Also note that a mass term is not present, so the equivalent force does not represent a real force on the end-effector unless that end-effector has constant acceleration and unit mass. It is instead a scaled approximation of the forces input to the end-effector.

Using the equivalent force definition, it is straightforward to construct second order dynamic models for discrete robot poses analogous to how they would be constructed using experimental modal analysis, as documented in [49]. Instead of applying an impulse force to the end-effector via an impact hammer, however, an equivalent force is applied by issuing a single position offset real-time command. By fitting a second order system model to the measured vibrational response, it is possible to estimate the mass, damping and stiffness parameters. Note again that these parameters are only valid under the equivalent force representation and represent a scaled approximation of the real mass, damping and stiffness of the end-effector. To facilitate this process, a MATLAB software application was created to allow for easy calibration of the dynamic parameters given a known input equivalent force and output measured vibration (Figure 32).

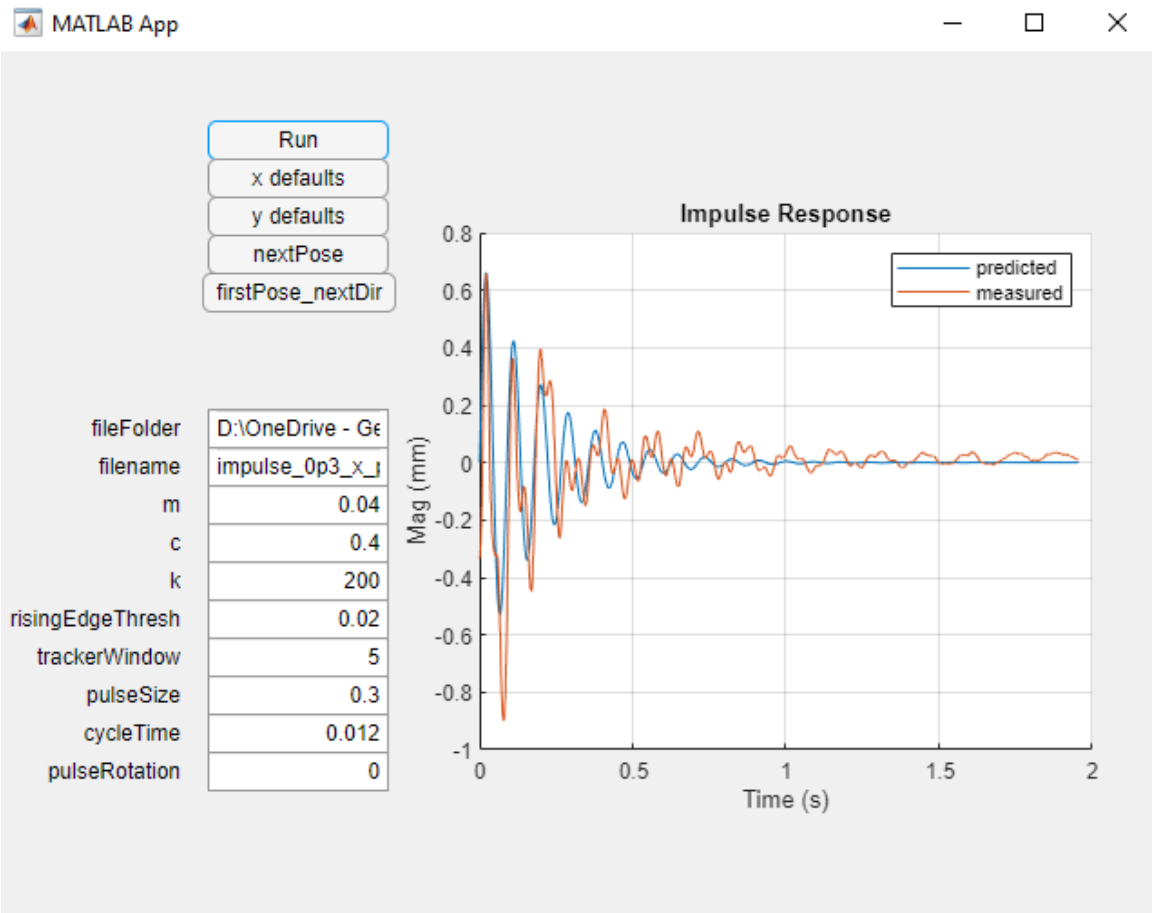

Figure 32 – Software application implemented to estimate the dynamic properties of the robot using the equivalent force model for a given pose and force/motion direction. The parameters to be estimated are *m* (mass), *c* (damping), and *k* (stiffness).

To capture the dynamic parameters for all possible poses of the manipulator would be prohibitively costly. However, the KR 500-3 used here shares a common attribute with heavy payload manipulators of its kind in that end-effector translation is mostly dominated by rotation of its first three joints, while the last three joints mostly contribute to end-effector rotation (Figure 3). Because joint 1 only rotates the arm about its base, joints 2 and 3 are the primary contributors to the overall arm configuration for a given pose. Assuming that the last three "wrist" joints have little influence on the pose-dependent dynamics [105],

the evaluation of the dynamic parameters can be constrained to only the joint 2 - joint 3 configuration space.

Because of its success in calibrating the real dynamic properties of the robot, to interpolate between end-effector dynamics captured at sampled poses, the same strategy of applying a Gaussian Process Regression (GPR) to the data is applied as in Chapter 2. The process is entirely analogous except that equivalent forces are used here instead of real forces as inputs to the dynamic model. Thus, one GPR model is trained at each pose for each Cartesian direction x, y, z and each dynamic parameter m, c, k for a total of 9 individual models. The sampling strategy to determine how many poses in the joint 2 - joint 3 space to sample is developed and evaluated in Section 0.

5.3.5 Applying the Equivalent Force Model to Predict Stability

Because it was hypothesized that only non-smooth accelerations cause significant command-induced error, it is not necessary to model every command sent to the robot as an equivalent force. For simplicity, it is assumed that robot trajectories are followed with a constant feed rate, a valid assumption for most precision manufacturing tasks involving curvilinear paths characterized by gradual curves such as in machining, welding, or painting. Under this assumption, only commands that induce non-smooth accelerations are modelled as equivalent forces as in:

$$F_{eq} = F_{commanded} - F_{feed} \tag{31}$$

Here, $F_{commanded}$ is defined as in (30) and F_{feed} is defined as:

$$F_{feed} = \frac{feed\ rate}{cycle\ time} \tag{32}$$

Note that the above forces are scalar quantities defined only in the feed direction. Any components of the commanded equivalent force not in the direction of the feed are automatically considered to induce non-smooth accelerations because they apply sudden steering to the trajectory of the end-effector. The final equivalent force to be input into the dynamic model is a vector of the equivalent forces modelled for each Cartesian direction. Note that as in Chapter 3, this means that the dynamic response in orthogonal directions (i.e. cross-coupling) is not captured.

Once the GPR models to predict pose-dependent dynamics are trained, it becomes possible to predict closed-loop stability for arbitrary smooth paths with constant feed rates. This is done by simulating the motion of the end-effector for a prescribed trajectory and closed-loop controller gain (Figure 33). The simulation is iterated in steps equivalent to the cycle time of the robot controller (4 ms). For the first step, the open-loop command sent to the robot is determined by the nominal trajectory. For this step, the robot motion is considered to have no error and to occur after the calibrated robot delay controller delay (52 ms). For subsequent steps, the motion error is computed as the difference between the current measured position of the end-effector and where it was commanded to go in the previous step. This error is input into the feedback controller which produces a position correction. This correction is input into the second order dynamic model predicted by the trained GPR model and the corresponding motion error is outputted. The simulation continues to run for the duration of the robot trajectory and stability is determined by

whether the end-effector position error reaches a steady-state value, oscillates, or grows unbounded.

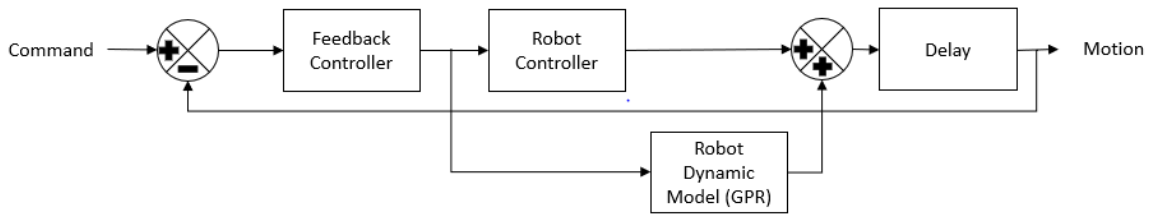

Figure 33 – Generalized block diagram of the real closed-loop system.

5.4 Experimental Setup and Results

5.4.1 Experiment Design

To calibrate the equivalent force dynamic model, 16 poses were sampled in the joint 2 – joint 3 space (Table 12). GPR models were trained on the poses and used to predict the equivalent force dynamics for 6 validation poses (Table 13). To evaluate the effectiveness of the motion error-prediction from the GPR model in predicting closed-loop stability, the robot was run through a series of trajectories in the x-y plane of the form:

$$y = 50.8\sin(2\pi * (0.0098)x) \tag{33}$$

Sine wave trajectories were used because their inherent oscillation can compound the oscillations caused by overly aggressive feedback gains and so typically require more manual experimental gain tuning than straight lines or large-radius curves. The amplitude and frequency of the sine wave were chosen to match typical experimental trajectories performed using this robot, such as described in Chapter 4. The feed rate was set at 42.33 mm/s and trajectories were run for 15 seconds.

Table 12 – Robot poses used to calibrate equivalent force model.

Pose number	Joint 2 (deg)	Joint 3 (deg)
1	-33.96	113.53
2	-31.26	98.33
3	-26.47	81.25
4	-19.33	60.68
5	-46.14	115.56
6	-41.59	100.18
7	-35.48	83.05
8	-27.42	62.67
9	-57.62	114.56
10	-51.05	99.27
11	-43.49	82.17
12	-34.31	61.7
13	-67.45	110.6
14	-59.08	95.65
15	-50.15	78.59
16	-39.74	57.71

Table 13 – Robot poses used to evaluate the stability prediction capabilities of the equivalent force model.

Pose number	Joint 2 (deg)	Joint 3 (deg)
17	-38.53	107.42
18	-34.02	91.33
19	-49.4	107.92
20	-43.22	91.79
21	-59.1	105.55
22	-51.27	89.57

A proportional (P) controller was applied to error in the y direction of motion as defined in (33). P control was used to limit the number of controller parameters influencing the stability of the closed-loop system to 1. However, the methodology presented in 5.3.5 can be applied to an arbitrary feedback controller scheme. To avoid potential timing

synchronization issues associated with a temporally defined trajectory, only the y coordinate was controlled with feedback, which allows the nominal coordinates at any given point in the path to be fully defined by the measured x coordinate.

For each validation pose in Table 13, the robot was run through a series of "x" and "y" directional sine waves. The x directional sine waves refer to trajectories where the x coordinate, as defined in (33), corresponds to the world x coordinate (Figure 34) and y sine waves refer to trajectories where the x coordinate defined in (33) corresponds to the world y coordinate. For each sine wave direction, trajectories were executed with increasing P gains until the closed-loop system became unstable. Because of the higher stiffness of the robot for x directional sine waves, P gains were incremented by 0.1, whereas the P gains were incremented by 0.01 for the less stiff y direction. The gains corresponding to stability and instability were recorded and compared to the stability predictions of the equivalent force model.

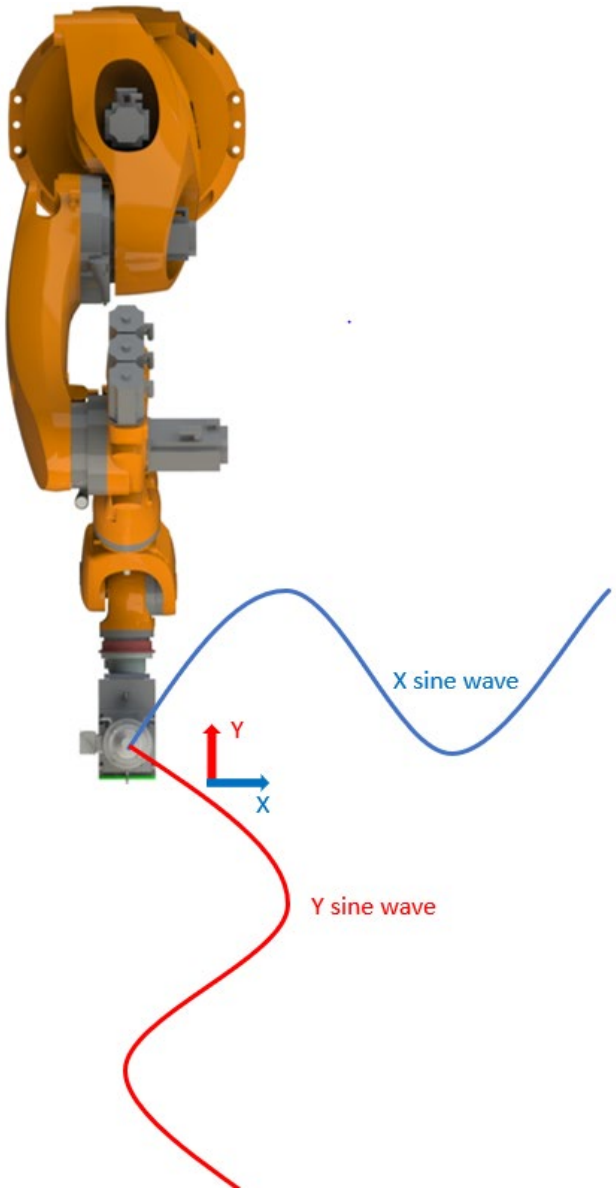

Figure 34 – World coordinate system used in the closed-loop hardware experiments and example sine wave motions.

5.4.2 Results and Discussion

Figure 35 and Figure 36 depict examples of the measured experimental trajectories versus the predicted performance by the naïve and GPR equivalent force models. In both cases, the equivalent force model is able to predict that the trajectory error grows

unbounded for the unstable gain, which the naïve model is unable to do. The plots show the stable P gain and the trajectory up to the point of instability. The sudden termination of the measured motions (black lines) occurs because the increasing error causes the robot controller to trip a maximum joint torque fault and force a stop before the run is completed. It should be noted that the equivalent force model does not always predict the exact moment when the error begins to grow unbounded in the trajectory every time. This is a limitation of the assumption that all robot motion error is caused by structural vibration. Other factors such as gear backlash and calibration error between the external sensor (laser tracker) and robot coordinate system are not modelled, meaning that the error prediction is an approximation. However, the purpose of this work is to be able to predict the stability of a gain over an entire trajectory, so pinpointing the moment when instability occurs is outside the scope of this work. Finally, the naïve model shown in the plots is the same as that defined in Section 5.1 but with the addition of the robot controller delay. The full set of measured versus predicted stable and unstable controller gains is given in Table 14.

Results show that the equivalent force model is able to predict the closed-loop stability up to a P gain resolution of 0.02 for this robotic system, trajectory, and control scheme. The gain prediction resolution may vary depending on the accuracy of the data available to calibrate the equivalent force model, the specific dynamics of the robot used, and the feedback controller scheme (P, PI, PID, etc.). In practice, it is recommended to choose controller gains conservatively, meaning two or three gain increments smaller than the maximum stable gain, to provide a margin of error for ensuring a stable closed-loop system. If accuracy tolerances are not met with these conservative gains, then it may be

necessary to incrementally increase them to achieve the desired accuracy.

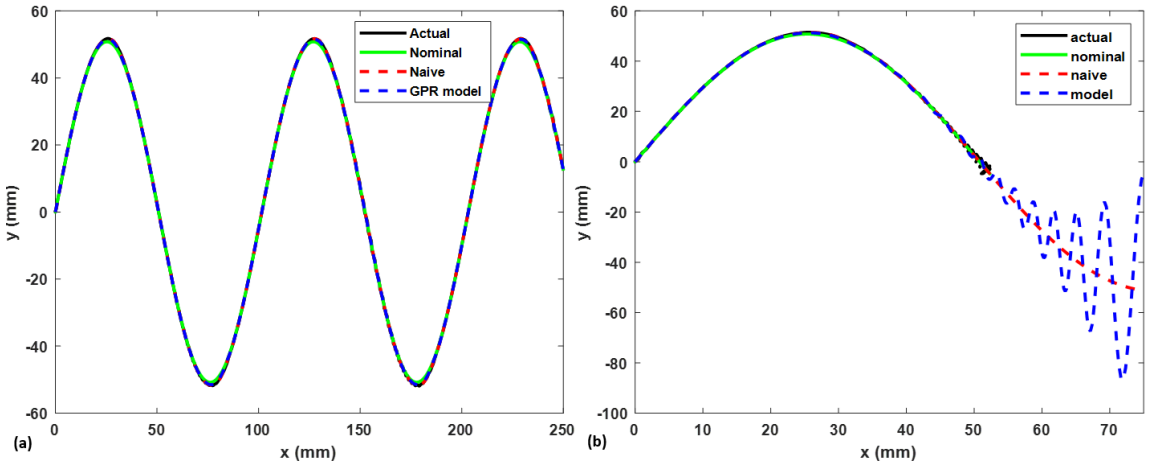

Figure 35 – Example closed-loop measured and predicted trajectory for a x-direction run with a stable P gain of 0.2 (left) and an unstable gain of 0.3 (right). This data is from the pose 17 experiments.

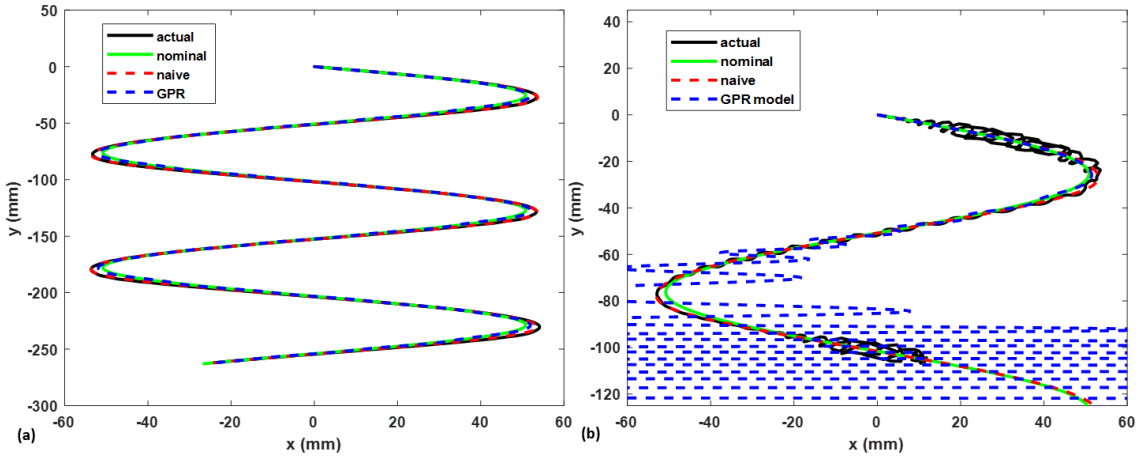

Figure 36 – Example closed-loop measured and predicted trajectory for a y-direction run with a stable P gain of 0.02 (left) and an unstable gain of 0.04 (right). This data is from the pose 19 experiments.

As mentioned, the gains corresponding to the y direction trajectories were incremented in much smaller increments than for the x direction. The variation in the stiffness of the robot in certain directions thus influences the range of stable gains available for each direction. In application, this result can be used for trajectory selection. If possible,

one should select the trajectory with the largest range of stable gains to get the maximum accuracy improvement from the closed-loop corrections.

Table 14 – Closed-loop stability prediction results.

Pose number	Predicted max. stable P gain for x direction	Actual max. stable P gain for x direction	Predicted max. stable P gain for y direction	Actual max. stable P gain for y direction
17	0.2	0.2	0.03	0.04
18	0.2	0.2	0.03	0.04
19	0.2	0.2	0.01	0.02
20	0.2	0.2	0.01	0.02
21	0.2	0.2	0.01	0.02
22	0.2	0.2	0.02	0.02

When the equivalent force is not aligned with the directions used to calibrate the equivalent force models, it must be decomposed into its components. For thoroughness, an additional set of experiments was conducted at pose 19 for directions not aligned with x or y. The trajectories were equivalent to those already used but rotated 30, 45, and 60 degrees below the world x axis (Figure 37). Experimental results confirm that the stability prediction is successful with a stable gain resolution of 0.02 (Figure 38). It should be noted that the measured trajectories depicted in Figure 38 do not terminate early for the unstable case. Instead, they exhibit heavy oscillation because they depict the boundary of stability. Raising the gain any higher results in an immediate maximum torque fault triggered by the robot controller at the start of the trajectory. Also, there is a quirk of simulating feedback in the local y direction (as defined in (33)) only that any local x error is propagated throughout the simulation. This has no impact on the previous experiments, which were aligned with the x or y axes because all the vibrational error was contained in the direction of the corrections. However, for the rotated trajectories, because the world x and y dynamic

properties are not equivalent, when predicted vibrational error is summed from the x and y directions, some error is accumulated in the x direction (Figure 38a). This accumulated x error does not affect the stability prediction, however, because the local y feedback controller ignores it.

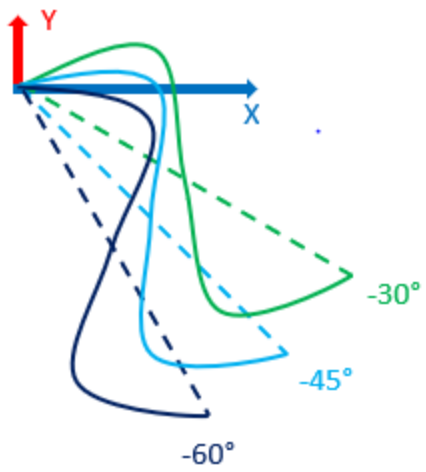

Figure 37 – Directions of off-axis experiments.

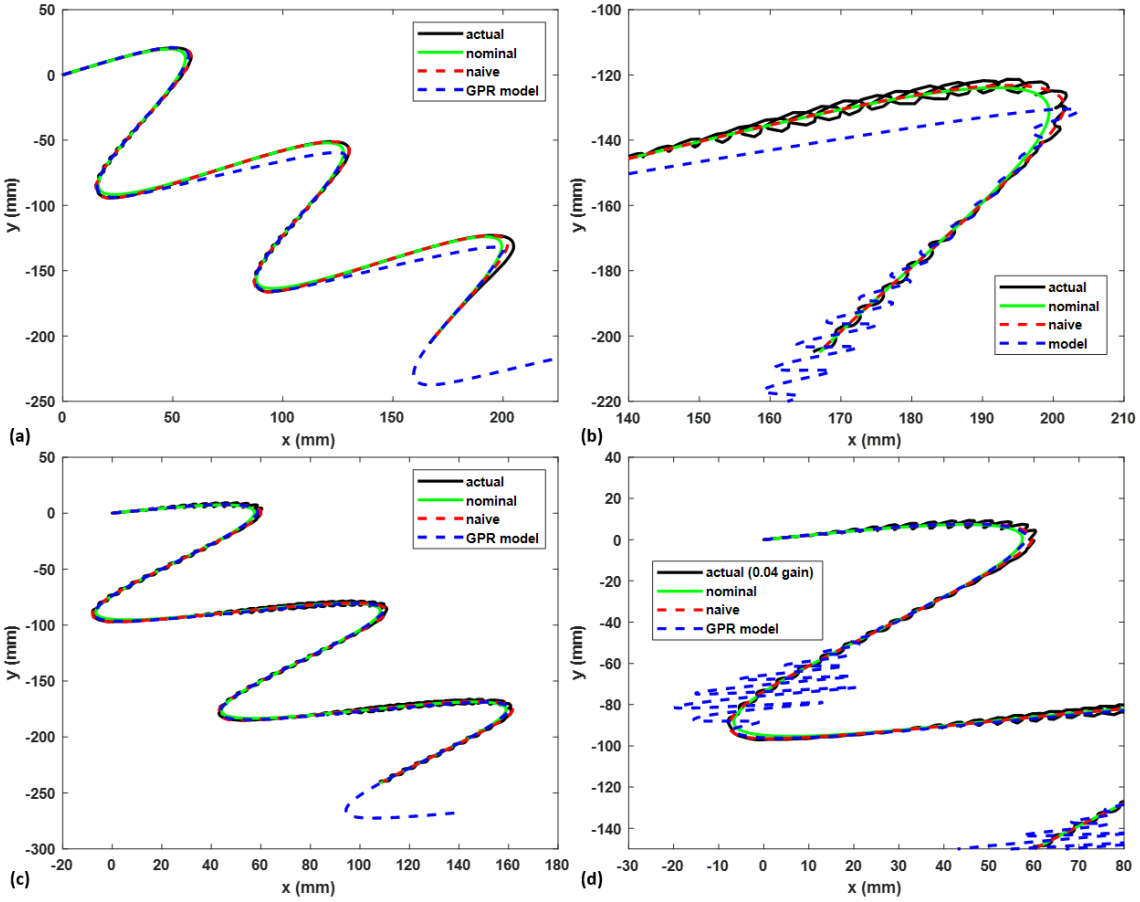

Figure 38 – Off-axis experiments. Figures (a) and (b) show the 45 degree rotated runs with P gains of 0.03 and 0.05 respectively. Figures (c) and (d) show the 60 degree

rotated runs with P gains of 0.04 and 0.06 respectively. Note that in (d), the actual data is the same as in (c) because the real system was already unstable at a P gain of 0.04 and so a run at a P gain of 0.06 was not conducted.

5.4.3 Sampling the Joint Space

The 16 pose sampling provided a good enough density to get accurate stability prediction. However, it would be useful to have a sampling strategy guideline for selecting the number of calibration poses for the equivalent force model. To develop this guideline, calibration experiments were conducted with increasingly dense sample sizes (Figure 39). Sampling steps were decreased from 15 degrees to 5 degrees in steps of 2.5 degrees, corresponding to an increase in sampled poses from 8 to 60. An equivalent force model was trained for each data set corresponding to each sampling density. To compare the ability of the models to capture the dynamic properties over the joint 2 – joint 3 space, the product of the mass (m) and stiffness (k) values was plotted across the joint space (Figure 40 and Figure 41). The product of m and k is proportional to the undamped vibration of the system for a given impulse input, so it is chosen as the metric for comparing the different models because it captures the most relevant information for closed-loop stability prediction (the motion error corresponding to the initial vibration after each command).

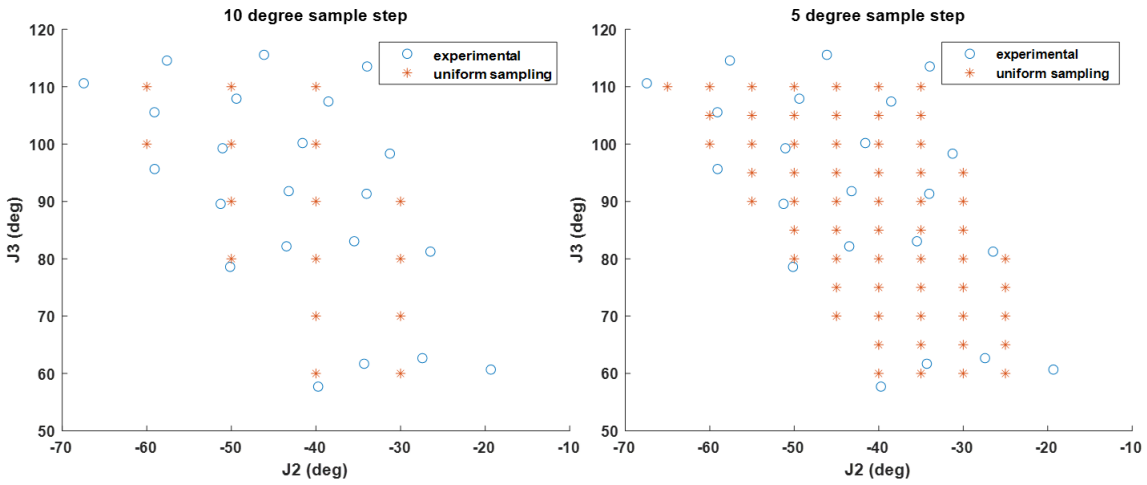

Figure 39 – Example sampling schemes of the joint 2 – joint 3 space. The experimental points represent poses 1-22 (for reference), while the orange points indicate the poses to be used for the corresponding uniform sampling density.

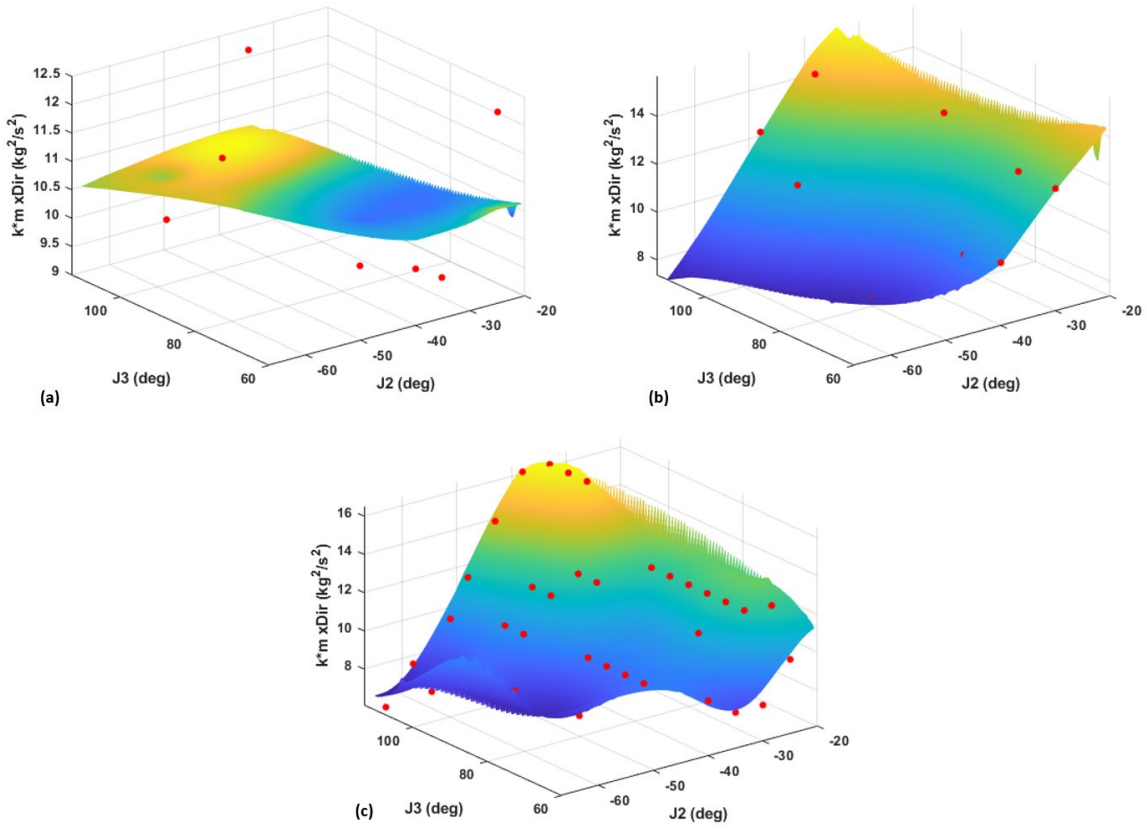

Figure 40 – Mass (m) times stiffness (k) plots over the joint 2 – joint 3 space for the world x direction. They correspond to models trained with a 15 degree (a), 10 degree (b), and 5 degree (c) pose sample step.

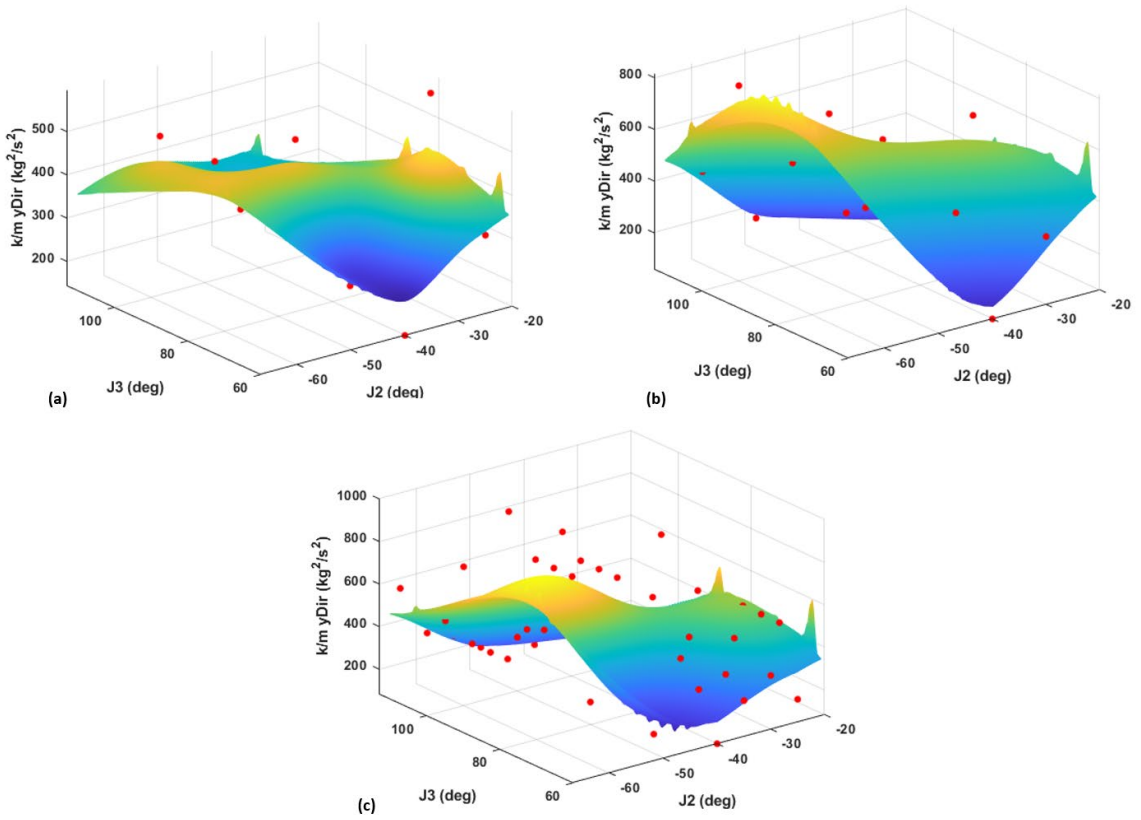

Figure 41 – Mass (m) times stiffness (k) plots over the joint 2 – joint 3 space for the world y direction. They correspond to models trained with a 15 degree (a), 10 degree (b), and 5 degree (c) pose sample step.

From Figure 40 and Figure 41, it is clear that for very low sampling densities, the GPR models default to close-to-constant mean values across the entire joint space. Only with denser sampling is it possible to capture the overall shape of the dynamic properties over the joint space. Table 15 lists the average percentage change of the m*k value between GPR models trained on data gathered from decreasing sample steps (increase pose sampling densities). The table shows that there is a large difference when going from extremely sparse to somewhat dense sampling, but that these differences drop off as the samplings approach their densest values. The data suggests that in practice, a sample step of no more than 10 degrees should be used to avoid the GPR models predicting a constant

mean value across the joint space. To determine the upper limit for the sampling density, a practical metric is to increase sampling density until the change in m*k between successive sample steps falls below the repeatability value of the equivalent force model calibration. The calibration repeatability is defined as the average amount by which the calibrated dynamic model parameters (m, c, and k) change between impulse experiments conducted at the same location and direction. For the experiments conducted in this work, this repeatability value was 15%.

Table 15 – Percent change of average m*k between data of decreasing sample steps.

Direction	15° to 12.5°	12.5° to 10°	10° to 7.5°	7.5° to 5°
X	22%	4%	8%	8%
Y	15%	50%	24%	18%

5.5 Summary

This Chapter aimed to improve the feedback controller design process for industrial robots by alleviating the need for manual gain tuning experimentation. To do this, first the major sources of real-time commanded motion error were identified to be robot controller delay and structural vibration of the robot. Then, the equivalent force model was established as an approximate model for the robot dynamics, which can be directly implemented as a control system plant model for an industrial robot controller, even though the controller can only take position/joint offsets as inputs. This model is used to predict closed-loop stability by simulating closed-loop trajectories and identifying which feedback controller designs result in stable versus unstable performance.

The most significant impact of this method of stability prediction is the time and cost saved by eliminating the need for iterative runs on application trajectories for manual gain tuning. If one considers a robotic system that performs a large variety of closed-loop trajectories in its application, the feedback controller would need to be re-tuned to get stable and optimal performance for each individual trajectory. With the new method, this retuning can be done quickly in simulation.

Finally, another benefit of the equivalent force model is that it allows for optimal trajectory direction selection. In the case that multiple trajectory orientations are viable options for a given application, it is possible to identify which of these orientations allow for the most aggressive controller gains to be implemented while still maintaining closed-loop stability. The more aggressive gains can allow for an overall higher closed-loop accuracy of the system.

CHAPTER 6. CONCLUSIONS AND FUTURE WORK

This chapter summarizes the major conclusions of this dissertation and lays out some recommendations for possible future work.

6.1 Conclusions

6.1.1 Static Versus Dynamic Stiffness Modeling for Pose Optimization

Pose optimization based on static and dynamic stiffness models was performed for various regions of the robot workspace and cutting conditions. Experimental data suggests that dynamic stiffness model-based optimization outperforms static-model based optimization when process forces approach the natural frequencies of the robot *and* when the space of possible arm configurations for the workspace location of the task contains poses with substantially different dynamic properties. Additionally, data shows that modifying the process parameters (e.g. spindle speed) to avoid excitation of the robot dynamics is a more powerful method for reducing deflections than pose optimization alone. Finally, it is demonstrated that when robot dynamics are not excited, a static model-based optimization performs comparably to a dynamic model-based optimization. This means that if process parameters can be configured to avoid excitation of the robot dynamics, a low-calibration effort static model is the recommended choice for calibrating a compliance model of the robot.

6.1.2 State Estimation Using Laser Tracker and Inertial Data

Simulation and hardware experiments were conducted to evaluate the accuracy of the end-effector state estimation with and without data fusion of position/orientation data with inertial data. Results indicate that while the directly measured quantities are not significantly improved, linear velocity estimation can be improved by up to 95% and angular acceleration estimation can be improved by up to 45%. Additionally, Kalman Filter estimations outperform a Particle Filter with 1000 particles for cases of linear or close-to-linear systems, but the Particle Filter performs better for highly nonlinear systems, such as when undergoing rapid rotation. Finally, a sensitivity analysis reveals the bounds of sensor bias and noise characteristics, sampling rate, and sensor-to-sensor transform error beyond which fusion is no longer beneficial.

6.1.3 Stability Prediction for Industrial Robots

A new method of stability prediction for closed-loop systems involving industrial robots is presented. The new method addresses the issue that the forces and torques associated with real-time commands sent to the robot are unknown by modeling real-time corrections by their "equivalent force". This equivalent force is based on the assumption that each real-time correction is achieved through a constant acceleration over the cycle time of the robot and that the end-effector has unit mass. Closed-loop trajectory following experiments indicate that the new method allows for stability prediction of a proportional controller for a resolution of 0.02 in the proportional gain. This stability prediction can be performed in simulation, and thus alleviates the need to do manual experimental gain-tuning for every new closed-loop trajectory the robot needs to be run through.

6.2 Original Contributions

The original contributions contained in this dissertation are as follows:

1. The circumstances under which using a dynamic stiffness model for pose optimization to minimize deflections is better than using a static stiffness model are identified through experimentation.
2. The benefits to a full state estimation of the end-effector gained by adding an inertial measurement unit to a system already containing a laser tracker are quantified and identified to be greatest in velocity and angular acceleration estimation. Kalman Filtering techniques are compared against Particle Filters for achieving the sensor fusion, and a sensitivity analysis is performed on the accuracy of the state estimation given different sensor bias and noise characteristics, sample rate, and transform error.
3. A new method for predicting closed-loop stability in systems involving industrial robots is presented and tested. This method allows for closed-loop controllers to be tuned in simulation rather than experimentally.

6.3 Recommendations for Future Work

Based on the work done in this dissertation and the relevant literature, the most accurate industrial robotic system can be achieved by performing extensive robot calibration and compliance modeling, estimating the state in real-time using laser tracker and inertial sensors, and by correcting the state in real-time via closed-loop feedback. Some possible direct extensions of this research include using the robot calibration to tune the state transition noise parameters as a function of pose for improved state estimation, adding

rotational deflection as part of the cost in pose optimization, examining the efficacy of active braking to increase joint stiffness, and fusing force/torque measurements with the inertial and laser tracker data for state estimation. A more general question that remains is how important the offline calibration methods are to the overall closed-loop performance. For example, if a robot's state is estimated and controlled using highly accurate sensors with high data rates, how well can it compensate for process-induced deflections even without a compliance model? Conversely, if only lower-end sensors are available for feedback, can an accurate robot calibration and compliance model bring the closed-loop accuracy to be in line with a system with the best sensors?

One other avenue of research that could benefit from this and related work is industrial robot design. Knowing the importance of compliance and closed-loop feedback, it should be possible to design manipulators with easy-to-model compliance and with more sophisticated tools for implementing feedback. Additionally, the sensors needed for feedback could all be built-in to the manipulator all with well-known transforms, eliminating the sensor-to-sensor transform error that comes with adding external sensors.

Finally, while this thesis provides some guidelines for practical implementation, such as when to use a dynamic versus static compliance model, there is room for developing a more comprehensive, automated approach to accurate control of industrial manipulators. The end goal would be an application that takes process parameters (workspace location, depths of cut, spindle frequency, etc.) as inputs, then outputs the optimal robot configurations and closed-loop gains to achieve the task. Additionally, any process parameters that are not locked can be given as a range that the application can optimize over. For example, if the spindle frequency must be between 250-500 Hz for a given cut,

the application would select the best frequency which maximizes the expected accuracy of the operation. The theoretical work for developing this sort of application is contained in this thesis, so all that remains is the software engineering and algorithm development to realize it.

APPENDIX A. DYNAMIC MODEL PARAMETERS

Table 16 – Dynamic model parameters at each test pose.

Test No.	m (Kg)			c (Ns/m)			k (MN/m)		
	x	y	z	x	y	z	x	y	z
1	166.5	41.1	281.5	3261.3	1486.3	281.5	1.900	5.500	1.206
2	174.8	223.6	281.4	4047.9	6526.0	281.4	2.790	5.967	1.206
3	166.5	41.1	281.5	3261.3	1486.3	281.5	1.900	5.500	1.206
4	173.7	212.2	281.4	4165.3	6527.9	281.4	2.974	5.405	1.206
5	165.3	36.9	207.8	4124.0	1653.8	207.8	2.098	5.204	0.820
6	252.7	75.8	185.4	6286.9	4630.5	185.4	3.346	2.415	0.787
7	165.3	36.9	207.8	4124.0	1653.8	207.8	2.098	5.204	0.820
8	252.7	75.8	185.4	6286.9	4630.5	185.4	3.346	2.415	0.787
9	190.2	352.1	352.5	3373.0	5749.9	352.5	2.204	2.596	1.520
10	234.0	358.4	377.2	5285.4	7255.4	377.2	3.114	3.079	1.791
11	338.8	398.0	206.1	5209.0	7709.2	206.1	0.479	5.722	0.556
12	351.7	380.7	240.2	5436.9	7779.0	240.2	0.500	5.499	0.690

Table 17 – Second most dominant modes at each test pose.

Test No.	k (MN/m)			Freq (Hz)		
	x	y	z	x	y	z
1	40	26	700	48	24	50
2	80	20	200	60	25	82
3	40	26	700	48	24	50
4	80	20	200	60	25	82
5	100	10	250	60	28	55
6	12	14	300	26	25	47
7	100	10	250	60	28	55
8	12	14	300	26	25	47
9	20	11	150	26	48	70
10	11	17	100	26	26	90
11	8	100	100	66	90	92
12	8	100	100	66	90	92

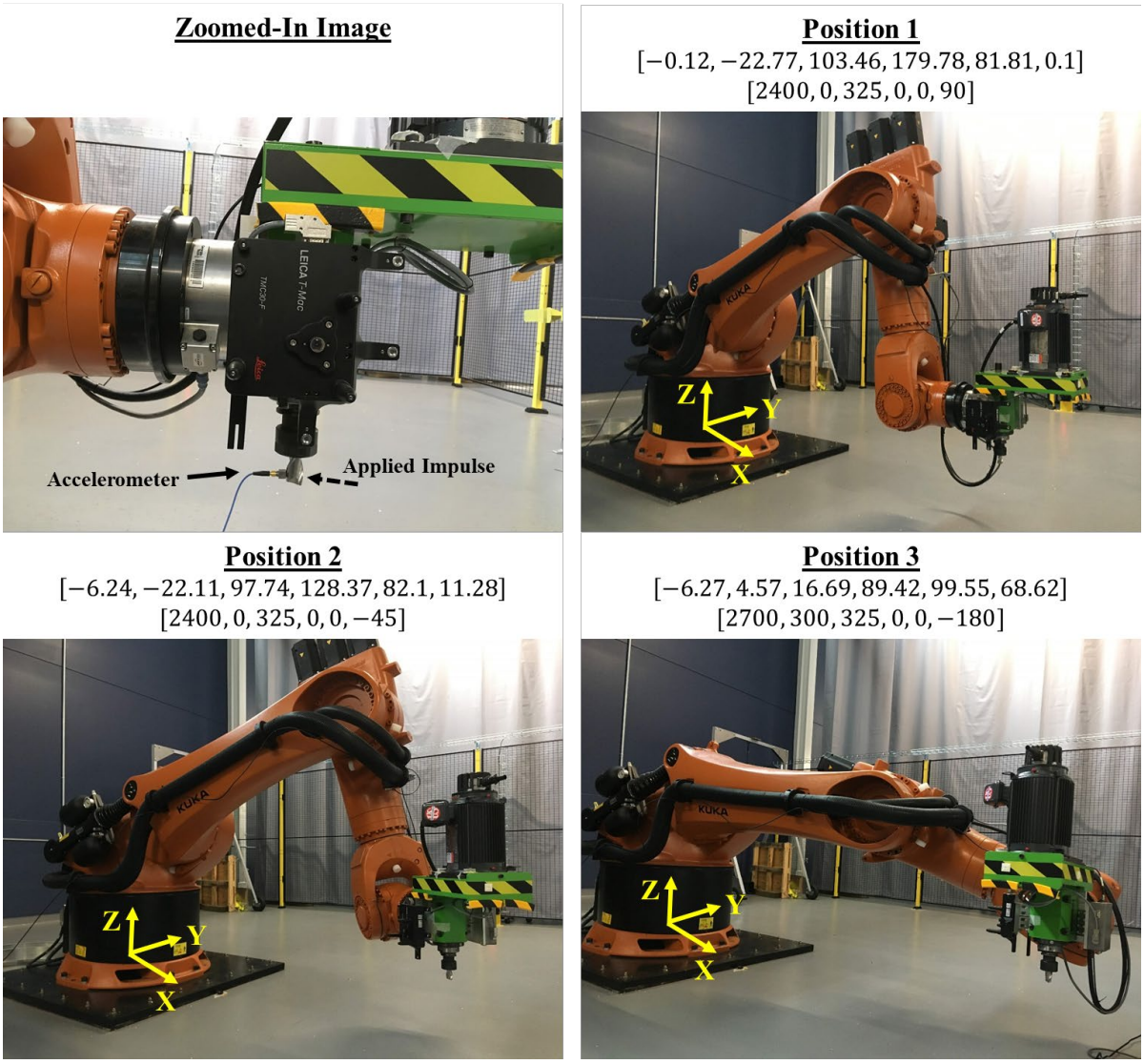

Figure 42 – Test poses used to check cross coupling effects.

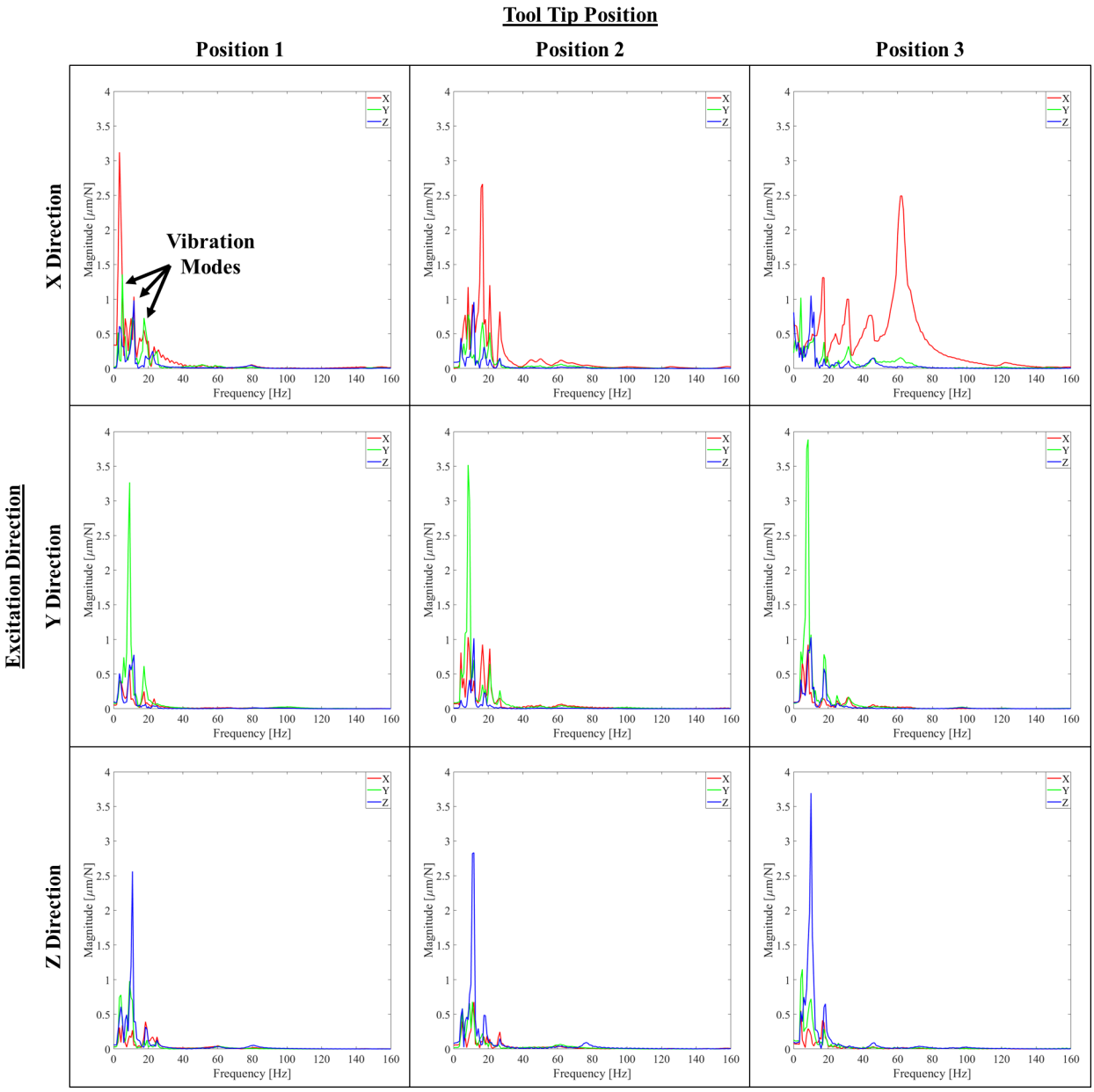

Figure 43 – Frequency responses from experimental modal analysis at each test pose used to check for cross coupling. Note that the vibration in the direction of the excitation is much larger than in the other directions.

APPENDIX B. KALMAN FILTER NOISE COVARIANCE MATRICES

The values of the process noise covariance matrix for the KF used in Chapter 4 were chosen to reflect the amount of information going into each part of the state when it is updated. The position value is updated with the most information because it depends on all three parts of the linear state: the past position, velocity, and acceleration (see (21)). Acceleration, on the other hand, is updated with the least information because it is assumed to be constant (see (23)). Thus, the first 3 diagonals, associated with the position state variables (x, y, z), are smaller than the subsequent values associated with velocity, which are in turn smaller than those associated with acceleration. The process noise covariance matrix (Q) is diagonal with entries of:

$$10^{-4}[2.54, 2.54, 2.54, 21.2, 21.2, 21.2, 98.1, 98.1, 98.1]$$

For the measurement noise covariance matrix (R), values were chosen to match the sensor specifications. The reduced matrix (with only position and acceleration associated terms) is diagonal with diagonal entries of:

$$[4.03 * 10^{-11}, 4.03 * 10^{-11}, 4.03 * 10^{-11}, 1.54 * 10^{-4}, 1.54 * 10^{-4}, 1.54 * 10^{-4}]$$

REFERENCES

1. Verl, A., et al., *Robots in machining.* CIRP Annals, 2019. **68**(2): p. 799-822.

2. Newman, W.S., et al. *Calibration of a Motoman P8 robot based on laser tracking*. in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*. 2000.

3. Motta, J.M.c.S., G.C. De Carvalho, and R. McMaster, *Robot calibration using a 3D vision-based measurement system with a single camera.* Robotics and Computer-Integrated Manufacturing, 2001. **17**(6): p. 487-497.

4. Ginani, L.S. and J.M.S.T. Motta, *Theoretical and practical aspects of robot calibration with experimental verification.* Journal of the Brazilian Society of Mechanical Sciences and Engineering, 2011. **33**: p. 15-21.

5. Meng, Y. and H. Zhuang, *Autonomous robot calibration using vision technology.* Robotics and Computer-Integrated Manufacturing, 2007. **23**(4): p. 436-446.

6. Gatla, C.S., et al., *An Automated Method to Calibrate Industrial Robots Using a Virtual Closed Kinematic Chain.* IEEE Transactions on Robotics, 2007. **23**(6): p. 1105-1116.

7. Ye, S.H., et al., *Robot Calibration Using Iteration and Differential Kinematics.* Journal of Physics: Conference Series, 2006. **48**: p. 1-6.

8. Gao, G., et al., *Structural parameter identification for 6 DOF industrial robots.* Mechanical Systems and Signal Processing, 2017.

9. Jang, J.H., S.H. Kim, and Y.K. Kwak, *Calibration of geometric and non-geometric errors of an industrial robot.* Robotica, 2001. **19**(3): p. 311-321.

10. Lightcap, C., et al., *Improved Positioning Accuracy of the PA10-6CE Robot with Geometric and Flexibility Calibration.* IEEE Transactions on Robotics, 2008. **24**(2): p. 452-456.

11. Joubair, A., A. Nubiola, and I. Bonev, *Calibration efficiency analysis based on five observability indices and two calibration models for a six-axis industrial robot.* SAE International Journal of Aerospace, 2013. **6**(2013-01-2117): p. 161-168.

12. Zhou, J., H.-N. Nguyen, and H.-J. Kang, *Simultaneous identification of joint compliance and kinematic parameters of industrial robots.* International Journal of Precision Engineering and Manufacturing, 2014. **15**(11): p. 2257-2264.

13. Joubair, A. and I.A. Bonev, *Non-kinematic calibration of a six-axis serial robot using planar constraints.* Precision Engineering, 2015. **40**: p. 325-333.

14. Kamali, K., et al. *Elasto-geometrical calibration of an industrial robot under multidirectional external loads using a laser tracker*. in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. 2016. IEEE.

15. Filion, A., et al., *Robot calibration using a portable photogrammetry system.* Robotics and Computer-Integrated Manufacturing, 2018. **49**: p. 77-87.

16. Khalil, W. and S. Besnard, *Geometric Calibration of Robots with Flexible Joints and Links.* Journal of Intelligent and Robotic Systems, 2002. **34**(4): p. 357-379.

17. Nubiola, A. and I.A. Bonev, *Absolute calibration of an ABB IRB 1600 robot using a laser tracker.* Robotics and Computer-Integrated Manufacturing, 2013. **29**(1): p. 236-245.

18. Gong, C., J. Yuan, and J. Ni, *Nongeometric error identification and compensation for robotic system by inverse calibration.* International Journal of Machine Tools and Manufacture, 2000. **40**(14): p. 2119-2137.

19. Alici, G. and B. Shirinzadeh, *A systematic technique to estimate positioning errors for robot accuracy improvement using laser interferometry based sensing.* Mechanism and Machine Theory, 2005. **40**(8): p. 879-906.

20. Alıcı, G., et al., *Prediction of geometric errors of robot manipulators with Particle Swarm Optimisation method.* Robotics and Autonomous Systems, 2006. **54**(12): p. 956-966.

21. Zeng, Y., W. Tian, and W. Liao, *Positional error similarity analysis for error compensation of industrial robots.* Robotics and Computer-Integrated Manufacturing, 2016. **42**: p. 113-120.

22. Zeng, Y., et al., *An error-similarity-based robot positional accuracy improvement method for a robotic drilling and riveting system.* The International Journal of Advanced Manufacturing Technology, 2017. **88**(9): p. 2745-2755.

23. Chen, D., et al., *A Compensation Method for Enhancing Aviation Drilling Robot Accuracy Based on Co-Kriging.* International Journal of Precision Engineering and Manufacturing, 2018. **19**(8): p. 1133-1142.

24. Cai, Y., P. Yuan, and D. Chen, *A flexible calibration method connecting the joint space and the working space of industrial robots.* Industrial Robot: An International Journal, 2018. **45**(3): p. 407-415.

25. Monica, T., et al. *A closed-loop neuro-parametric methodology for the calibration of a 5 DOF measuring robot*. in *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No.03EX694)*. 2003.

26. Liu, J., Y. Zhang, and Z. Li, *Improving the Positioning Accuracy of a Neurosurgical Robot System.* IEEE/ASME Transactions on Mechatronics, 2007. **12**(5): p. 527-533.

27. Aoyagi, S., et al. *Improvement of robot accuracy by calibrating kinematic model using a laser tracking system-compensation of non-geometric errors using neural networks and selection of optimal measuring points using genetic algorithm*. in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010.

28. Messay, T., et al. *GPGPU acceleration of a novel calibration method for industrial robots*. in *Proceedings of the 2011 IEEE National Aerospace and Electronics Conference (NAECON)*. 2011.

29. Nguyen, H.-N., J. Zhou, and H.-J. Kang, *A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network.* Neurocomputing, 2015. **151**: p. 996-1005.

30. Yuan, P., et al., *A compensation method based on extreme learning machine to enhance absolute position accuracy for aviation drilling robot.* Advances in Mechanical Engineering, 2018. **10**(3): p. 1687814018763411.

31. Chen, D., et al., *A positional error compensation method for industrial robots combining error similarity and radial basis function neural network.* Measurement Science and Technology, 2019. **30**(12): p. 125010.

32. Schneider, U., et al. *Stiffness modeling of industrial robots for deformation compensation in machining*. in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014.

33. Klimchik, A., et al., *Identification of the manipulator stiffness model parameters in industrial environment.* Mechanism and Machine Theory, 2015. **90**: p. 1-22.

34. Alici, G. and B. Shirinzadeh, *Enhanced Stiffness Modeling, Identification and Characterization for Robot Manipulators.* IEEE Transactions on Robotics, 2005. **21**(4): p. 554-564.

35. Abele, E., M. Weigold, and S. Rothenbücher, *Modeling and Identification of an Industrial Robot for Machining Applications.* CIRP Annals - Manufacturing Technology, 2007. **56**(1): p. 387-390.

36. Dumas, C., et al., *Joint stiffness identification of six-revolute industrial serial robots.* Robotics and Computer-Integrated Manufacturing, 2011. **27**(4): p. 881-888.

37. Ambiehl, A., et al. *New method for decoupling the articular stiffness identification: Application to an industrial robot with double encoding system on its 3 first axis*. in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017. IEEE.

38. Lehmann, C., et al., *Robot joint modeling and parameter identification using the clamping method.* IFAC Proceedings Volumes, 2013. **46**(9): p. 813-818.

39. Klimchik, A., et al. *Calibration of industrial robots with pneumatic gravity compensators*. in *Advanced Intelligent Mechatronics (AIM), 2017 IEEE International Conference on*. 2017. IEEE.

40. Johansson, R., et al., *State-space system identification of robot manipulator dynamics.* Mechatronics, 2000. **10**(3): p. 403-418.

41. Gautier, M. and P. Poignet, *Extended Kalman filtering and weighted least squares dynamic identification of robot.* Control Engineering Practice, 2001. **9**(12): p. 1361-1372.

42. Grotjahn, M., M. Daemi, and B. Heimann, *Friction and rigid body identification of robot dynamics.* International Journal of Solids and Structures, 2001. **38**(10): p. 1889-1902.

43. Swevers, J., W. Verdonck, and J. De Schutter, *Dynamic model identification for industrial robots.* IEEE Control Systems, 2007. **27**(5): p. 58-71.

44. Gautier, M., A. Janor, and P.-O. Vandanjon. *DIDIM: A new method for the dynamic identification of robots from only torque data*. in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. 2008. IEEE.

45. Mousavi, S., et al., *Dynamic modeling and stability prediction in robotic machining.* The International Journal of Advanced Manufacturing Technology, 2017. **88**(9-12): p. 3053-3065.

46. Jiang, Z., T. Ishida, and M. Sunawada. *Neural Network Aided Dynamic Parameter Identification of Robot Manipulators*. in *2006 IEEE International Conference on Systems, Man and Cybernetics*. 2006.

47. Mejri, S., et al., *Dynamic characterization of machining robot and stability analysis.* The International Journal of Advanced Manufacturing Technology, 2016. **82**(1): p. 351-359.

48. Chen, C., et al., *Posture-dependent stability prediction of a milling industrial robot based on inverse distance weighted method.* Procedia Manufacturing, 2018. **17**: p. 993-1000.

49. Nguyen, V., T. Cvitanic, and S. Melkote, *Data-Driven Modeling of the Modal Properties of a Six-Degrees-of-Freedom Industrial Robot and Its Application to Robotic Milling.* Journal of Manufacturing Science and Engineering, 2019. **141**(12).

50. Busch, M., et al., *Probabilistic information fusion to model the pose-dependent dynamics of milling robots.* Production Engineering, 2020. **14**(4): p. 435-444.

51. Wu, K. and B. Kuhlenkoetter, *Experimental Analysis of the Dynamic Stiffness in Industrial Robots.* Applied Sciences, 2020. **10**(23): p. 8332.

52. Wang, J., H. Zhang, and T. Fuhlbrigge. *Improving machining accuracy with robot deformation compensation*. in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009.

53. Sörnmo, O., et al., *Increasing time-efficiency and accuracy of robotic machining processes using model-based adaptive force control.* IFAC Proceedings Volumes, 2012. **45**(22): p. 543-548.

54. Lehmann, C., et al. *Milling with industrial robots: Strategies to reduce and compensate process force induced accuracy influences*. in *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*. 2012.

55. Belchior, J., et al., *Off-line compensation of the tool path deviations on robotic machining: Application to incremental sheet forming.* Robotics and Computer-Integrated Manufacturing, 2013. **29**(4): p. 58-69.

56. Haage, M., et al. *Increasing Robotic machining accuracy using offline compensation based on joint-motion simulation*. in *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*. 2014. VDE.

57. Slavkovic, N.R., D.S. Milutinovic, and M.M. Glavonjic, *A method for off-line compensation of cutting force-induced errors in robotic machining by tool path modification.* The International Journal of Advanced Manufacturing Technology, 2014. **70**(9): p. 2083-2096.

58. Kaldestad, K.B., I. Tyapin, and G. Hovland. *Robotic face milling path correction and vibration reduction*. in *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. 2015.

59. Klimchik, A., et al., *Compliance error compensation in robotic-based milling*, in *Informatics in Control, Automation and Robotics*. 2014, Springer. p. 197-216.

60. Tunc, L.T. and D. Stoddart, *Tool path pattern and feed direction selection in robotic milling for increased chatter-free material removal rate.* The International Journal of Advanced Manufacturing Technology, 2017. **89**(9): p. 2907-2918.

61. Schneider, U., J.R.D. Posada, and A. Verl. *Automatic pose optimization for robotic processes*. in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. 2015. IEEE.

62. Guo, Y., H. Dong, and Y. Ke, *Stiffness-oriented posture optimization in robotic machining applications.* Robotics and Computer-Integrated Manufacturing, 2015. **35**(Supplement C): p. 69-76.

63. Diaz Posada, J.R., et al., *Automatic Motion Generation for Robotic Milling Optimizing Stiffness with Sample-Based Planning.* Machines, 2017. **5**(1): p. 3.

64. Bu, Y., et al., *Stiffness analysis and optimization in robotic drilling application.* Precision Engineering, 2017. **49**: p. 388-400.

65. Xie, H., W. Li, and Z. Yin. *Posture Optimization Based on Both Joint Parameter Error and Stiffness for Robotic Milling*. in *Intelligent Robotics and Applications*. 2018. Cham: Springer International Publishing.

66. Mousavi, S., et al., *Stability optimization in robotic milling through the control of functional redundancies.* Robotics and Computer-Integrated Manufacturing, 2018. **50**: p. 181-192.

67. Qin, H., Y. Li, and X. Xiong, *Workpiece Pose Optimization for Milling with Flexible-Joint Robots to Improve Quasi-Static Performance.* Applied Sciences, 2019. **9**(6): p. 1044.

68. Liu, S., et al. *Laser tracker-based control for peg-in-hole assembly robot*. in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on*. 2014. IEEE.

69. Posada, J.D., et al. *High accurate robotic drilling with external sensor and compliance model-based compensation*. in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. 2016. IEEE.

70. Devlieg, R., *Expanding the Use of Robotics in Airframe Assembly Via Accurate Robot Technology.* SAE Int. J. Aerosp., 2010. **3**(1): p. 198-203.

71. Schneider, U., et al., *Position control of an industrial robot using an optical measurement system for machining purposes*, in *11th International Conference on Manufacturing Research (ICMR2013)*. 2013: Cranfield University, UK. p. 307-312.

72. Holden, R., P. Lightowler, and S. Andreou, *Robot Accuracy: Online Compensation (EU COMET Project).* SAE Int. J. Aerosp., 2014. **7**(2): p. 274-279.

73. Droll, S., *Real Time Path Correction of Industrial Robots with Direct End-Effector Feedback from a Laser Tracker.* SAE International Journal of Aerospace, 2014. **7**(2): p. 222-228.

74. Moeller, C., et al., *Real Time Pose Control of an Industrial Robotic System for Machining of Large Scale Components in Aerospace Industry Using Laser Tracker System.* SAE International Journal of Aerospace, 2017. **10**(2017-01-2165): p. 100-108.

75. Stadelmann, L., et al., *End-Effector Pose Correction for Versatile Large-Scale Multi-Robotic Systems.* IEEE Robotics and Automation Letters, 2019. **4**(2): p. 546-553.

76. Wang, Z., R. Zhang, and P. Keogh, *Real-Time Laser Tracker Compensation of Robotic Drilling and Machining.* Journal of Manufacturing and Materials Processing, 2020. **4**(3): p. 79.

77. Gharaaty, S., et al. *Accuracy enhancement of industrial robots by on-line pose correction*. in *2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*. 2017.

78. Lin, Y., H. Min, and H. Wei, *Inertial measurement unit–based iterative pose compensation algorithm for low-cost modular manipulator.* Advances in Mechanical Engineering, 2016. **8**(1): p. 1687814015626850.

79. Ligorio, G. and A.M. Sabatini, *Extended Kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: Comparative analysis and performance evaluation.* Sensors, 2013. **13**(2): p. 1919-1941.

80. Forster, C., et al., *On-Manifold Preintegration for Real-Time Visual--Inertial Odometry.* IEEE Transactions on Robotics, 2017. **33**(1): p. 1-21.

81. Lupton, T. and S. Sukkarieh, *Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions.* IEEE Transactions on Robotics, 2012. **28**(1): p. 61-76.

82. Steidle, F., A. Tobergte, and A. Albu-Schäffer. *Optical-inertial tracking of an input device for real-time robot control*. in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016.

83. Nam, C.N.K., H.J. Kang, and Y.S. Suh, *Golf Swing Motion Tracking Using Inertial Sensors and a Stereo Camera.* IEEE Transactions on Instrumentation and Measurement, 2014. **63**(4): p. 943-952.

84. Tailanián, M., et al. *Design and implementation of sensor data fusion for an autonomous quadrotor*. in *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*. 2014.

85. Olofsson, B., et al., *Sensor Fusion for Robotic Workspace State Estimation.* IEEE/ASME Transactions on Mechatronics, 2016. **21**(5): p. 2236-2248.

86. Chen, J., et al., *A robust optical/inertial data fusion system for motion tracking of the robot manipulator.* Journal of Zhejiang University SCIENCE C, 2014. **15**(7): p. 574-583.

87. Du, G., et al., *Online Robot Kinematic Calibration Using Hybrid Filter with Multiple Sensors.* IEEE Transactions on Instrumentation and Measurement, 2020: p. 1-1.

88. Syed, Z.F., et al., *Civilian Vehicle Navigation: Required Alignment of the Inertial Sensors for Acceptable Navigation Accuracies.* IEEE Transactions on Vehicular Technology, 2008. **57**(6): p. 3402-3412.

89. Altintas, Y., *Manufacturing automation: metal cutting mechanics, machine tool vibrations, and CNC design*. 2012: Cambridge university press.

90. *Kuka KR 360-3; KR500-3 Specification*. [cited 2019; Available from: https://www.kuka.com/-/media/kuka-downloads/imported/48ec812b1b2947898ac2598aff70abc0/spez_kr_3603_kr_5003_en.pdf.

91. Klimchik, A., et al. *Identification of geometrical and elastostatic parameters of heavy industrial robots*. in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. 2013. IEEE.

92. Cen, L. and S.N. Melkote, *CCT-based mode coupling chatter avoidance in robotic milling.* Journal of Manufacturing Processes, 2017. **29**: p. 50-61.

93. Wang, G., et al., *Chatter mechanism and stability analysis of robotic boring.* The International Journal of Advanced Manufacturing Technology, 2017. **91**(1-4): p. 411-421.

94. Yuan, L., et al., *Mode coupling chatter suppression for robotic machining using semi-active magnetorheological elastomers absorber.* Mechanical Systems and Signal Processing, 2019. **117**: p. 221-237.

95. He, F.-x., Y. Liu, and K. Liu, *A chatter-free path optimization algorithm based on stiffness orientation method for robotic milling.* The International Journal of Advanced Manufacturing Technology, 2019. **101**(9-12): p. 2739-2750.

96. Rasmussen, C.E., *Gaussian processes in machine learning*, in *Advanced lectures on machine learning*. 2004, Springer. p. 63-71.

97. Cen, L. and S.N. Melkote, *Effect of Robot Dynamics on the Machining Forces in Robotic Milling.* Procedia Manufacturing, 2017. **10**: p. 486-496.

98. Zengxi, P. and Z. Hui. *Analysis and suppression of chatter in robotic machining process*. in *2007 International Conference on Control, Automation and Systems*. 2007.

99. Thrun, S., W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. 2005: The MIT Press.

100. Simon, J.J. and K.U. Jeffrey. *New extension of the Kalman filter to nonlinear systems*. in *Proc.SPIE*. 1997.

101. Wang, Z., *Experimental Evaluation of Leica T-Mac Angular Accuracy.* The Journal of the CMSC, 2018. **13**(2): p. 28-32.

102. Gharaaty, S., et al., *Online pose correction of an industrial robot using an optical coordinate measure machine system.* International Journal of Advanced Robotic Systems, 2018. **15**(4): p. 1729881418787915.

103. Young, K. and C.G. Pickin, *Accuracy assessment of the modern industrial robot.* Industrial Robot: An International Journal, 2000. **27**(6): p. 427-436.

104. Slamani, M., A. Nubiola, and I. Bonev, *Assessment of the positioning performance of an industrial robot.* Industrial Robot: An International Journal, 2012. **39**(1): p. 57-68.

105. Juyi, P., et al., *Design of learning input shaping technique for residual vibration suppression in an industrial robot.* IEEE/ASME Transactions on Mechatronics, 2006. **11**(1): p. 55-65.