国立大学法人
東京農工大学
Tokyo University of Agriculture and Technology

Philosophiae Doctor Thesis

Study on Essential Technologies for Practical and Productive Social Robot Development

From 1st April 2020 to 31st March 2023

Shohei HAGANE
Tokyo University of
Agriculture and Technology

Faculty of Engineering

Department of Mechanical System
Engineering Venture Lab
Student ID: 20833009

Tokyo University of Agriculture and Technology
東京農工大学
Naka-chō 2-24-16
184-0012 Koganei-shi
Tōkyō-to, JAPAN

Supervisors

Prof. Gentiane VENTURE
Director of GV Lab

Prof. Yasutaka TAGAWA
Director of Tagawa Lab

Abstract

English

This doctoral dissertation discusses the technologies that will be needed for robots to enter and become productive members of human society in the future. Based on previous research, social robots may add social functions, including emotional expression, in order to be more acceptable to people. On the other hand, it is also true that the market demands not only sociable functions but also the ability to perform productive tasks. Therefore, both practical and communicative aspects are necessary for future robots to participate in society. In this paper, a 7-degree-of-freedom serial link robot arm is treated as the subject of research. We propose a generalized design of a control system that enables the robot arm to perform more intelligent and diverse tasks, including emotional expressive behavior. The research can be divided into three main parts.

In the first study, an adaptive generalized predictive control (AGPC) was designed using kinematic parameters estimated by dynamic identification of the robot arm to allow the robot to perform tasks in response to changes in the grasped object at its end-effector. The performance of AGPC was compared with PID control, which is still dominant in industry, and generalized predictive control (GPC) that is for systems linearized by feedback linearization. The results showed that AGPC exhibited the best position control and external force control performance based on the parameters estimated from the dynamic identification, even when the size and mass of the grasped object changed.

Same as in the previous section, the second study is about generalized predictive control, but here, the main focus is on linearizing the dynamical system of a robot using the concept of the Koopman operator. Using this concept of the Koopman operator, a nonlinear dynamical system of a robot can be represented by a linear time-invariant state-space model. This allows us to apply advanced linear control laws, including generalized predictive control, to the robot arm. Furthermore, since the computational cost is lower than that of predictive control based on rigorous modeling, high control frequency can be achieved in a real environment. Here, a 7-DOF robot is simplified to a 3-DOF robot, and the Koopman operator is identified using a method inspired by the robot arm dynamics model. A state-space model was extracted from the identified Koopman operator, and its accuracy was compared with that of the state-space model using Koopman operators in previous studies. We also designed a generalized predictive controller for the obtained linear state-space model of the robot and compared its control performance with that of a conventional PD controller. The results show that the GPC with Koopman operators maintains constant control performance over a wide range of postures, speeds, and work ranges compared to the PD controller with fixed gains.

In the last study, we proposed a motion control system that uses the kinematic redundancy of the robot arm to perform emotional expressive motion when the robot arm performs a main task at its end-effector. As mentioned earlier, in order for robots to play an active role in society, they must be able to perform both communicative and productive tasks. Here, we used null-space control to exploit kinematic redundancy and applied the concept of manipulability ellipsoid to maximize motion within the null space. The proposed method was implemented on an actual upper body humanoid robot "Nextage-Open" and recorded the interactions with a person. Using the recordings, the robot's emotions were evaluated online by more than 200 people via Google-Form. The results

showed that current rules surrounding industrial machines operating near people severely limit the robot's speed and prevent it from expressing the movement-by-movement changes necessary to express emotion. However, we found that people who viewed unclear emotional behavior used the content of the interaction, the robot's gaze, and their own prejudices about the robot, in addition to the emotional behavior, to interpret the robot's emotions.

Finally, we concluded that for a robot to perform emotionally expressive actions in a human-close environment, it needs a precise motion controller that realize different elements of subtle movements, and a structure made of soft and light materials to to clear safety standard set by society. Model predictive control is suitable for controlling the robot to use different subtle motions, and this study is significant in that it provides an example of systematic application of MPC, generalized predictive control, to a robot. Furthermore, the new attempt to use kinematic redundancy for emotional expression is also academically meaningful.

Keywords: Model Predictive control, Generalized Predictive Control, Dynamic Parameter Identification, External Force Control, Feedback Linearization, Optimal Control, Koopman Operator, Dynamic Mode Decomposition, Redundant Robot Arm, Robotic Manipulator, Null Space Control, Manipulability Ellipsoid, Pleasure-Arousal-Dominance Model (PAD), Emotional motion, HRI, NARS

要旨

この博士論文は、ロボットが将来に人間社会に参入して生産的な社会のメンバーとなる為に必要となる技術について考察したものである。先行研究から、社会的なロボットがより人々に受け入れられるためには、感情表現をはじめとした社交的な機能を追加することが考えられる。一方で、社交的な機能だけでなく、生産的なタスクをこなす能力が市場では求められていることも事実であることから、実用的な側面とコミュニケーション的な側面の両立が、未来のロボットの社会参画には必要である。本稿では、ロボットとして7自由度直鎖型ロボットアームが研究の対象として扱う。ロボットアームが感情表現動作を含む、より知的で多様なタスクをこなすことができる制御システムの一般化されたデザインを提案する。研究は大きく分けて3つに分けられる。

一つ目の研究は、ロボットが把持物体の変化に対応してタスクを行う事ができるように、ロボットアームの力学同定によって推定された動力学パラメータを用いて、適応型一般化予測制御（AGPC）を設計した。AGPCは未だに産業界で支配的なPID制御と、フィードバック線形化によって線形化されたシステムに対して設計された一般化予測制御（GPC）と性能を比較された。その結果、AGPCは把持する物体の寸法と質量が変わっても、力学同定から推定されたパラメータを基に最も優れた位置制御性能と、外力制御性能を示した。

2つ目の研究では、前章と同じく、一般化予測制御についてなのだが、ここでは、クープマン作用素という概念を用いてロボットの力学系を線形化することを主眼に置いて研究を行った。このクープマン作用素という概念を用いれば、非線形なロボットの力学系を線形時不変状態空間モデルで表現する事ができる。このことによって、一般化予測制御を始めとする高度な線形制御則をロボットアームに適用することができる。さらには、厳密なモデリングによる予測制御と比べて計算コストが小さく済むため、実環境でも高い制御周波数を実現できる。ここでは、7自由度ロボットを3自由度ロボットとして利用し、ロボットの動力学モデルからヒントを得た方法でクープマン作用素を同定した。同定されたクープマン作用素から状態空間モデルを抽出し、先行研究のクープマン作用素による状態空間モデルとの精度の比較を行った。また、得られたロボットの線形状態空間モデルに対して一般化予測制御器を設計し、古き良きPD制御器と制御性の比較を行った。その結果、固定ゲインのPD制御器と比べて、クープマン作用素によるGPCは幅広い姿勢、速度、作業範囲に対して一定の制御性能を保つことが示された。

最後の研究では、ロボットアームが手先のタスクを行う際に、余った運動学的な自由度を利用して感情表現動作を行う動作制御システムを提案した。前述の通り、ロボットが社会で活躍するためには、コミュニケーションと生産的なタスクを両立することが求められる。ここでは、運動学的な冗長性を利用するために、ヌル空間制御を利用し、ヌル空間内の動作を最大化するために、可制御性楕円のコンセプトを応用した。提案された手法は上半身ヒューマノイドロボット“Nextage-Open”の実機に実装され、人間とのインタラクションを録画した。その録画を使って、Google-Formによってロボットの感情を200人以上の人にオンラインで評価してもらった。その結果、人の近くで動作する産業用機械を取り巻く現行の規則は、ロボットの速度を著しく制限し、感情を表現するために必要な動作ごとの変化を表現できないことが分かった。しかし、不透明な動作を見た人々-は、感情的な動作の他に、インタラクションの内容や、ロボットの目線、そして自身のロボットに対する偏見をロボットの感情の解釈に利用していることが分かった。

最終的に、ロボットが人間に近い環境で感情表現豊かに動作するためには、微妙な動作の要素を使い分ける精密な動作制御器と、柔らかで軽い素材の構造が必要であると結論付けた。ロボットが微妙な動作を使い分けるための制御には、モデル予測制御（MPC）が適しており、本研究は一般化予測制御というMPCをロボット対してシステマチックに適用する一例を示した点で有意義である。さらに、運動学的な冗長性を感情表現に利用するという新たな試みも学術的に意味のあるものである。

キーワード： モデル予測制御、一般化予測制御、力学同定、外力制御、フィードバック線形化、最適制御、 クープマン作用素、線形化、冗長ロボットアーム、ロボットマニピュレータ、null空間制御、可制御性楕円、冗長空間、PADモデル、感情表現、HRI、NARS

Acknowledgement

This doctoral dissertation was made possible with the help of many people. I would like to thank them for their help.

First of all, I would like to express my special thanks to Professor Venture Gentiane for giving me many opportunities for research experience and guiding me through my research life.

I would also like to express my gratitude and respect to Professor Yasutaka Tagawa, who opened the world of control engineering to me.

I would also like to express my deepest gratitude to everyone at the Venture Lab with whom I shared my research life. In particular, I would like to thank Siméon Capy and Pablo Osorio, who helped each other as fellow doctoral students during my doctoral studies.

I would also like to thank Dr.Vincent Hernandez, Dr.Enrique Colorado and Professor Liz Rincon for being good mentors and friends.

I would like to thank Professor Danna Gulotta, the teachers at ELS, and my fellow international students who have been a great help to me in my English studies during my time in Boston and St. Petersburg.

I would like to thank Professor Dana Kulić and the RoboHub members at the University of Waterloo, Canada. It was a valuable experience for me to operate the actual robot, and I was able to make great progress in my research.

Professor Lorenzo Jamone of Queen Mary University of London and the ARQ members warmly welcomed me in the midst of the Covid-19 pandemic and made my research life in London fruitful. I would like to thank them once again.

I would also like to thank Kawada Robotics Corporation for their support of the IREX2019 International Robot Exhibition and my research, with special thanks to Yuichiro Kawasumi, Victor Leve, and Naoya Rikiishi.

I would also like to express my gratitude here to friends who have supported me in my personal life.

Finally, I would like to express my sincere gratitude to my parents and siblings who have understood and supported me to this point.

ここまで自分を理解し、支えてくれた両親と兄弟に心からの感謝をします。

Contents

List of Figures

List of Tables

List of Acronyms

AGPC Adaptive Generalized Predictive Control/Controller. 3, 13, 85, 87

CoM Center of Mass. 17

CTC Computed Torque Control/Controller. 87

DoF Degree of Freedom. 2, 3, 14, 85, 86

DPI Dynamic Parameter Identification. vi, 7, 13, 21, 87

GPC Generalized Predictive Control/Controller. 3, 4, 6, 13, 20, 22, 23, 26, 27, 29, 30, 37, 41, 44, 85, 87

HRI Human-Robot Interaction. 1, 12

KP Koopman Operator. 3, 85–88

MPC Model Predictive Control/Controller. 2, 3, 5, 6, 12, 85, 88

OET Optimal Exciting Trajectory. 18, 30

PD Proportional-Differential. 3

PID Proportional-Integral-Differential. 2, 12

SIP Segment Inertial Parameters. 7, 13, 17, 87

SVD Singular Value Decomposition. 86, 88

Chapter 1

Introduction

SUMMARY

Dramatic improvements in computing power have given rise to great expectations for social implementation of robots through the application of machine learning. This thesis paves the way for robots to achieve both social and productive functions that are required for robots to coexist with humans. In this chapter, we outline the main motivations for implementing emotional expression functions in robots and provide background on the underlying control and machine learning methods. Furthermore, we summarize the structure of this thesis.

1.1 Motivation and Context

It has been almost 100 years since the concept of robots was invented by Karel Čapek in 1920, and they have gone from being mere industrial tools to being expected to be functional members of the human society. This trend seems to have been accelerated by the recent prosperity of machine learning methods that have accompanied the dramatic increase in computing power. However, advanced machine learning approaches have further revealed the complexity of the real world and highlighted the difficulty for robots to move with the same intelligence as living organisms. Furthermore, despite daily advances in cognitive engineering, robots are still at the mercy of elusive human behavior. "How can we make robots acts intelligently?" and "How will people and society react when confronted with such robots?". These two questions about technology and significance are probably simultaneous if they are to be answered, and therefore must be explored simultaneously. And this thesis describes both the exploration of modern control methods for precise robot motion (technology), and evaluation of Human-Robot Interaction (HRI) (social impact).

As robots become more integrated into daily life, an important question arises: how can robots become more acceptable for people? In the 2017 Pew Research Center survey “Automation in Everyday Life", 4135 people in the United States were asked, “Are you interested in using robots in your daily life, including care-giving?" Out of the respondents 64% had negative perceptions on robots, stating that they “can’t read robots’ mood" and that robots are “simply scary" [6]. Furthermore, in a recent 2019 survey by Provide Japan Inc. on “Questionnaire on ’Human Work and Robot Work’," 1020 individuals in Japan were asked, “How did you feel about robots compared to people?" the results showed that 49.7% of the respondents had negative opinions, with the contents being “lack of warmth, not understanding of emotions, and scary" [7]. Thus, people’s feelings toward robots are still not quite positive.

One of the ways being explored to gain user acceptance is to incorporate emotional movements. In order to overcome the negative reputation of robots, researchers have been implementing emotions into robots [8, 9, 10, 11]. One of the main functionality of robots’ mood/emotion expression is that it gives social value to interactions between people and robots [8]. Yamamoto et al. discussed human behavior toward computer programs (i.e., pseudo-interpersonal behavior) [12], and found that the "improvement of system functions alone does not cause pseudo-interpersonal behavior, but some other element is necessary." The "element" is the social value of interaction between

people and robots/program, and numbers of researchers have been making effort towards creating that social value. This can be done by displaying affective content on the screen and/or through expressive sound and movements. Through the emotional expression of robots which can draw out people's empathy in a natural way, the robots are no longer recognized as simply a "tool" but as something social, which can help to appease feelings of aversion and dislike for robots [13].

If a robot does not have a display board or voice function, such as a robot arm, it may still be able to communicate through motion. Researchers have been trying to convey certain emotions through robot motions (see section 2.5). For instance, Knight attempted to use a small robot that only has a few Degree of Freedom (DoF) [8], and Beck et al. used a humanoid that utilize human gesture information to generate expressive robot motions [9]. In these studies, the general tactics for differentiating emotions are adjusting pre-existing gestures/postures and/or the velocity of the motions. Those attempts have successfully conveyed arousal and mood through robot motion.

Human ability to detect slight motion is outstanding and it makes the robots' emotional conveyance by motion more challenging. Because recognition of conspecifics and enemies is ecologically important, the human visual system is capable of perceiving minute changes in facial expression and subtle body movement cues. In particular, the human visual system is very sensitive to the movements of human shaped entity. We can even tell the sex of a person from their movements alone [14]. Johansson's demonstration of biological motion is well known [15]. Humans' eyes are keenly aware of noise of vibrations among robot motions that are unintended by the operator and try to make sense of the mood/emotions, even though it just mere unintentional fluctuation.

In order to achieve detailed operation of various emotional motions, more advanced control method than the traditional Proportional-Integral-Differential (PID) control is required. Because of its intuitive parameter tunability, the PID control is still dominant in general industry. However, with regard to robot motion control, due to its nonlinear dynamic nature, a fixed PID gain can only guarantee its control performance in a single, fixed motion. While the emotional motions are often not fixed gestures and these velocity and acceleration are varied according to the degree of the emotion intensity. Among the advanced control theory, we focus on the Model Predictive Control/-Controller (MPC) that utilize dynamic model of the robots and predict its behavior to optimize future input. MPC has adaptability to the different robot postures/trajectory but require detailed modeling and computationally expensive. Especially, the technique to embed nonlinear dynamics to the model such as state space model is needed for implementation of MPC to the robots.

Moreover, to be more accepted by society, just sharing emotion is not enough for robots. Most research on emotional robots treat emotion conveyance as the main task, so the robots cannot do any other productive tasks such as manipulating objects. People's expectations of robots are becoming clearer, and the image of a convenient existence to do work is generally an important element in the image of robots. The reality is that household robots such as Vector and Kuri, which are only for sharing emotions, are forced to withdraw from the market even though these companies had been viewed as promising [16]. This trend implies that, in order for a robot to be accepted, it should have both a productive function and social function.

Thus, we need a motion controller that enables robot to execute designed motions precisely in order to communicate emotion with human in higher level. In this thesis, application of MPC is considered as the subject of the research. Moreover, the motion control system that allow robot to conduct productive task and social functionality at same time would be beneficial for social robots.

1.2 Objective

In this thesis, three distinct objective is presented regarding fore-mentioned motivations and the attempt to achieve those objective is presented in following chapters as primary work. First two objectives are about model identification and predictive control for robotic system. last one objective is about developing motion control system that enable to execute a productive task and social

task (emotional expression) at same time.

The first objective is to design a predictive control adaptation scheme based on the Generalized Predictive Control/Controller (GPC) using the robot dynamic and geometrical identification with optimal exciting motions. Acquiring model of controlled system (reference model) is essential for MPC and the controller needs to adapt new model if the reference model changed. Robot arms that perform productive tasks attach various tools to their end-effectors and that cause model error between the reference model in the controller and actual robot model. In order to achieve highly accurate position control with various tool, Adaptive Generalized Predictive Control/Controller (AGPC) was developed. AGPC uses dynamic parameter information such as the mass of each link and the position of the center of gravity, identified through optimal exciting motion data, and not only update reference model but also update control parameter of MPC based on the stability index. The AGPC were evaluated using the 7 DoF KUKA LWR arm.

Furthermore, we attempt to apply GPC to robots with an approach that globally linearizes the nonlinear dynamics of the robot, whereas the aforementioned AGPC utilized local linearization of the referenced model. We also propose a method to identify the dynamical information of a robot by analyzing the linearly represented dynamical system of the robot. This thesis presents an example of the linearization of the robot manipulator's nonlinear dynamics using Koopman Operator (KP), and introduces practical issues to be considered when applying KP to robotic systems. The basic idea of KP is to lift a non-linear function to a higher dimensional space by extending its explanatory variables, and these new variables are often referred to as "observables". In this higher dimension space, nonlinear functions can be described in a linear form. Using this, GPC was designed and aimed high tracking performances for a wide range of motions compared to traditional Proportional-Differential (PD) control. Also, We attempted to ascertain information about the mass by comparing a set of linear models identified with different masses grasped.

Finally, we develop a motion system that performs tasks while expressing emotions using the kinematic redundancy of robot arms. A system for performing emotional actions using redundant DoF while performing industrial tasks has not yet been demonstrated in the social robot field. Although Claret et al. have shown an example of a humanoid (Pepper) that utilize the redundant degrees of freedom to express emotion, its main task was a waving gesture and emotion conveyance, while productive task were not implemented [10]. In contrast, the target of our study is a robot arm that is specialized in productive tasks. Furthermore, we will mention the concerns of work-oriented robots as they work in the real world, expressing their emotions through their motions. The system is implemented and evaluated using an upper body humanoid (Kawada Robotics Nextage-Open).

1.3 Outline of the Thesis

This dissertation is structured into six sections, which can be outlined as follows:

- This Chapter 1 is introductory. it gives the motivation and context addressed in this dissertation and states objectives of author's doctoral works.
- Chapter 2 presents the state of the arts that related to the primal works in this dissertation.
- Chapter 3 presents first primal work that is *Adaptive generalized predictive controller and Cartesian force control for robot arm using dynamics and geometric identification.* Section 3.2 describes the kinematics/dynamic model for KUKA Light weight robot manipulator. Section 3.3 introduces the dynamic parameter method that is developed by co-authors of [17]. Section 3.4 describes the proposed AGPC and other controllers that is used in the control experiments. Section 3.5 describes dynamic/geometric parameter identification result of the painting roller and experimental setup for painting task in V-rep simulation. Section 3.6 presents results of painting task with proposed AGPC and comparison between other controllers. Also, discussion regarding to those results are presented in the section. Section 3.7 sum up the Chapter 3.

- Chapter 4 presents second primal work that is *Linearizing Robotic Manipulator's Dynamics Using Koopman Operator and Applying Generalized Predictive Control*. Section 4.3 describes the concept of Koopman operator for controlled dynamic systems. Section 4.4 introduces the designing method of GPC with state space model. Section 4.5 describes the open question about linearizing robot arm's dynamcs using Koopman operator and propose a solution. Section 4.6 present the experimental setup and shows results that compare proposed method and state of arts. Section 4.7 gives discussion regarding the results and present potential of presented method for dynamic/geometric parameter identification. Section **??** concludes the Chapter 4.
- Chapter 5 presents third primal work that is *Robotic Manipulator Expressing Emotions as Sub-Task Using Kinematic Redundancy*. Section 5.2 describes the kinematics model for redundant robot manipulators. Section 5.3 introduces the proposed emotion conveyance method. Section 5.4 describes the implementation of the method on a real robot, questionnaire contents, and experiment. Section 5.5 and 5.6 present the experimental results and the discussion, respectively. Section 5.7 concludes the Chapter 5.
- Finally, Chapter 6 presents the conclusions of the thesis, outlining the main contributions and directions for further research regarding limitations of presented works.

Chapter 2

Related Works

SUMMARY This section introduces previous studies related to the primal work in this paper and presents the positioning of this study. MPC for robotic systems, dynamic parameter identification, and Koopman operator theory in robotic systems are presented. Also, exsamples of HRI and social robotics, expressive motion by robotics, exploiting kinematic redundancy and finally, method for evaluating expressive movements and emotion are presented.

2.1 Model Predictive Control (MPC) for Robotic Systems

MPC [18] is a control method that determines control inputs while solving a finite-time optimal control problem at each time. MPC was developed in the 1970s as a control algorithm for systems with relatively slow control loops but large system delays, such as oil refineries and chemical plants. At that time, MPC was not uniformly called MPC, but was called Dynamic Matrix Control or Model Predictive Heuristic Control. Later, with improvements in the computational performance of controllers and advances in software (solvers) for solving optimization problems, the application of MPC has expanded to real-time control systems with high-speed control loops such as robotic systems.

MPC is also called Receding Horizon Control in which control inputs are determined by solving an optimal control problem in a finite interval at each time. Since the control input is determined by the state at each time, this method is characterized as both optimal control and feedback control. Besides the better control performance, MPC could provide two more advantages compared to conventional PID control. First, it is possible to systematically handle constraint conditions on inputs and outputs. In real situations, often there are constraint conditions on the inputs and outputs of the controlled system. For example, in the case of a robot arm, they are the limits of acceleration, the maximum value of velocity in terms of safety, and the maximum joint angles. Also, there is almost no input or output without constraint conditions, such as the maximum value of torque, voltage. In model predictive control, such constraint can be taken into account in optimization using the Lagrange's undetermined multiplier method, etc. [19], so that precise control can be performed with accurate consideration of the constraint conditions.

MPC is a control system that combines a model of controlled system and an optimization algorithm and it can be classified into two types depending on the system model and optimization algorithm used. The first is when the system model is a linear model and the constraints can be expressed in terms of convex functions or convex spaces. In this case, the MPC control problem can be solved by a convex optimization algorithm. Such an MPC is called a linear MPC, and the term MPC generally refers to this linear MPC. Another case is when the controlled system is non-linear. While linear MPC can find solutions quickly by using convex optimization techniques, Nonlinear MPC requires the use of nonlinear optimization to find solutions, which can be more computationally expensive than convex optimization, and the optimal solution may be initial value dependent. However, compared to linear MPC, it is possible to handle nonlinear models as they are in real, and in some cases, it may perform better than linear MPC [20].

In the past, researchers have worked in two areas to realize the MPC advantage on robotic systems by linearization of the model. About the optimization, improving efficiency of computation of optimization problem is one of the main motivation of research about MPC. When applied to robot control, for example, control inputs must be generated by iteratively solving non-linear optimization problems of appropriate size at intervals of several tens of milliseconds and it is computationally heavy. Therefore, it is difficult to use non-linear optimization for control systems that require fast response. Although, as computer performance has improved in recent years and this may not be a problem in some cases. Yet, it is still necessary to use high-speed logic for optimization which is generally for linear system. For real-time control using MPC, it is important to efficiently solve the optimal control problem, and various methods have been proposed for this purpose[20].

About the linearization, to apply MPC to a robot, a linear model must be designed. As fast optimization technique is often require system to be linear, the object to be controlled by MPC must be described by a linear model. If the controlled system is very complex or nonlinear, this linear model can be difficult to create. Robot arm's linearized model is basically designed by physics and geometric relations. For example, Taji et al. described a dynamic system in which a dual-armed two-link robot arm handles a single object in a state-space model, including friction. Since the model was a time-varying continuous-time state-space model, the changes in the state were assumed to be minute during the short prediction interval, and all the equations of motion, etc., were linearly approximated by fixing the inertia matrix, etc., to the values of the initial conditions [21]. Besides linearly approximated model, approximation by First-order Taylor expansion is also used [20],[22]. While these methods are locally linearize the system dynamics, strict linearization using feedback linearization is possible such as Computed Torque Control [23]. On the other hand, there are also methods to generate a linear model by neural networks or machine learning with various inputs and outputs [24, 25].

Sometimes, the model description is devised to ensure robustness to system change of conditions and to model errors and noise that arise during linearization and measurements. Robust MPC is an algorithm that predicts a set of states (Reachability Set) containing errors in the model, rather than a single point whose predictions are computed from the model, and controls the entire set to satisfy the constraints [24]. and Stochastic MPC is a more stochastic approach to MPC. The prediction of a state may be described by a stochastic model, or the cost function may be specified in terms of the expected value of the stochastic model [26].

There are also studies in which the system stability and robustness against model errors and noise are acquired by appropriately adjusting the MPC control parameters. For example, the extended horizon adaptive control (EHAC) proposed by B.E. Ydstie [27], or the extended prediction self-adaptive control (EPSAC) proposed by Keyser and Van Cauwenberghe were developed [28]. In the EHAC control, the adaptation of the control law depends on the system parameters. The controller adapts the final prediction horizon and fixes the optimal weighting parameter (λ). The parameters are updated over the whole process. The EPSAC updates the prediction horizon, weighting factor, and filter polynomials [29, 30]. However, the on-line adaptation affects the structure of the predictor and control, which may lead to unstable motions and the coefficients of the prediction equation cannot be obtained at certain points in time. Moreover, the controller produces a reduced functionality to calculate the control effort, and an extension of the horizon results in a delayed velocity response [20, 31].

GPC is a controller in the MPC category [32, 33]. It was developed by Clarke [2], and it is now widely used in industry and academia [34, 35]. Moreover, it was successfully implemented in several industrial applications with good performance and adequate robustness [32, 34, 18]. M. Makarov et al. [36] applied GPC to an anthropomorphic robotic arm, and it demonstrated a superior tracking and robustness to that of PID controllers. Furthermore, GPC can handle several control problems [37] for a wide range of plants with a significant number of design variables, which have to be specified by the user based on prior knowledge of the plant and control objectives.

2.2 Dynamic Parameter Identification

Dynamic parameter identification methods are becoming increasingly important in the development of model-based controllers. Precise dynamic parameters are required because many control methods are sensitive to their values. Especially in high-speed operation, sensitivity to dynamic parameter uncertainty is high. Atkeson et al. [38] proposed the estimation of inertial parameters of rigid-body loads and links, demonstrated with PUMA 600 robot. Gautier et al. [39] proposed a dynamic identification method from only the torque data. Grotjahn et al. [40] used the two-step approach to perform the friction and rigid body identification of robot dynamics. The focus of several studies has been the identification of the serial manipulator Segment Inertial Parameters (SIP) [41], [42], [43], [44], [45]. Identification method can be categorized into two group: on-line identification and off-line identification methods.

In the off-line procedure, all the input–output data is collected prior to the computation of parameters and there is no computation time limit. There are three main off-line methods for robot arm. The first is done by physical experiments in which the robot is disassembled to isolate each link and conducted experiments to obtain inertial parameters. For example, the mass could be evaluated directly and the center-of-mass of the parts could be obtained by examining counterbalanced points of the link. The inertia tensor could be obtained through pendular motions [46]. However, to conduct physical experiments of each parts, the special measuring devices are usually needed and the characteristics of joints are neglected. Furthermore, practically speaking, physical experiments should be done by the manufacture of the robot before assembling.

Second off-line method is Computer Aided Design (CAD) techniques. This approach uses geometric and material characteristics which can be found in CAD data. Robotics CAD/CAM packages often provide tools to calculate the inertia parameters from 3-dimensional models. However, the precision of the material/geometric model in CAD system determines the accuracy of the estimated parameter. Due to the manufacturing error of the link, the CAD model is not identical with the real part as manufacturing error exsists and friction parameters are not predictable from CAD drawing. Therefore, the accuracy of the estimated parameter values is effected.

The third off-line identification method is based on the analysis of the "input/output" behavior of the robot on some planned excited motion. It use mathematical model of robot dynamics to compare difference between a the real variables and its model with estimated dynamic parameters. This method was found to be the best in terms of ease of experimentation and precision of the identification by Khalil et al. [47]. Also [48] identified the dynamic parameters of payload of the robot. The least-square and maximum likelihood estimation methods are the most popular approaches. Furthermore, the accuracy of SIP estimation is related to the maximization of information, i.e. the design of optimal exciting trajectories [49], [50], [51], [52].

Contrary to the off-line technique, the on-line method conduct real-time updates of the dynamic parameters during robot operations. This method finds values of the parameters in the mathematical model of a system from on-line measured data [53]. One of the way to incorporate identified parameter to the control on time is adapting control algorithms. the method has been investigated extensively to estimate or adjust the dynamic parameter values used in the control on-line. For the case of parallel robots, problems such as workspace constraints and the complexity of the dynamic equations are reduced with adaptive control algorithms. Then based on the Jacobian matrix, Burdet and Codourey [54] projected the force and torque of each moving part onto the Cartesian space.

Another important trend about parameter identification is using neural networks. Neural networks have received considerable attention in the dynamic parameter identification fields and Control in general. It has been demonstrated that neural networks can effectively identify non-linear dynamics [55]. The identified parameters are represented as the weights of the nodes. Neural network can realize the on-line identification since the trained network output real-time parameter value fast as real-time data are given. Jiang et al. [56] proposed an identification approach using neural network to compensate uncertainty of dynamics.

2.3 Koopman Operator Theory on Robotics Systems

Koopman operator is a concept related to data-driven modeling of dynamics, which can describe nonlinear dynamical systems in a linear form. Moreover, unlike modeling with neural networks, it is theoretically a strict linearization, and semantic analysis results can be obtained. Above all, the major advantage is that nonlinear dynamical systems are represented by state-space models. Discoveries made by data-driven methods are now revolutionizing the way complex systems are modeled, predicted, and controlled. Modern scientific and engineering problems of increasing complexity cannot be addressed by empirical modeling or knowledge of basic physics alone. More and more researchers are adopting data-driven approaches to complex systems as diverse as turbulence, the brain, climate, epidemiology, finance, and robotics. These complex systems are typically nonlinear, dynamic, multi-scale in space/time, and high-dimensional, but have dominant underlying patterns that are necessary for the ultimate goal of prediction, estimation, and control. Modern mathematical methods and the explosion of computational performance have enabled us to tackle challenges that were previously unfeasible.

Here, the concept called "Koopman Operator" (KP) is particularly interesting [57, 58]. The KP is a linear operator defined for nonlinear dynamical systems and represents the time evolution of the observed quantity (output function in control engineering) of the dynamical system. This operator is a linear operator that preserves the information of the original nonlinear dynamical system and it allows mapping dynamic constraints into a linear dynamical system in a higher dimensional state space. Its basic idea is to lift non-linear function to a higher dimensional space by extending its explanatory variables that are often referred as "observables". In this higher dimension, nonlinear functions can be described in a linear form. Here, how the observables should be augmented depends on the dynamical system's property and needs to be chosen with consideration, such as which kind of observables and how many should be[59].

The KP have only recently begun to attract attention in the field of robotics, and only a few applications exist. For example, Bruder et al. applied to soft robotic arm [60], and Abraham showed an example of a serial manipulator controlled with KP [61]. However, the performance of the linearized model was not discussed in detail for the robot arm case, and the control result by LQR left room for improvement. In particular, it is still unclear how to extend the state vector in a way that is suitable for describing the dynamics of a robot arm.

Linear dynamic models derived from KP enable the application of linear control methods and hold great promise in the field of control engineering. In particular, the adaptability of KP with Model Predictive Control (MPC) [62], is pointed out by Korda [63] and Igarashi et al. [64]. In fact, there are several works that show numerical examples of linearization of the non-linear systems and control by linear control theory like LQR [65, 66] and MPC [63]. Furthermore, control examples with linearized model by KP can be found in vehicle control [67], fluid dynamics [68], and energy grid management [69].

Another similar direction in model-based control is to learn robot dynamics models through machine learning and apply them to control [59]. For example, in [70], the nonlinear robot dynamics is represented as a mixture of local linear models, learned from data, and used for control. Although the concept of representing nonlinear dynamics with a myriad of linear combinations is similar to the KP, it differs from the method using machine learning in that the analysis of the model through the KP in this study is obtaining a state-space model of the system and can be applied to state-space model analysis methods and control design that were established in control engineering.

2.4 HRI and Social Robotics

In HRI research, robot control and human perception have been studied in two main research directions: physical (pHRI) and social (sHRI). Our study falls under sHRI. Much of pHRI research is focused on robots, developing control frameworks that treat humans as extraneous noise, external

forces, or another "robot" [71][72][73].

In contrast, sHRI research is focused on human perception. Social human-robot interactions without physical contact have often been investigated through specially designed experiments. Researchers have been trying to analyze mental states, comfort, sense of safety, sociability, and general perceptions of users. Such mental measures are often acquired using post-experimental questionnaires [74][75][76][77].

Weiss et al.[74] used the HRP-2 humanoid to investigate whether the general attitude of people who are skeptical towards robotics changed after interacting with the robot. Our study uses the Negative Attitude Towards Robot Scale (see section c)) used in their research. Kamide et al.[76] aimed to discover the basic factors for determining the "Anshin" of humanoids from the viewpoint of potential users. "Anshin" is the Japanese concept of subjective well-being toward life with artificial products. According to the factor analysis in their research, the five factors of Anshin are comfort, performance, peace of mind, controllability, and robot-likeness. Existing sHRI studies have also attempted to give robots emotional intelligence and investigate the human perception of emotional robots as mentioned in the following section.

2.5 Expressive Motions by Robots

Expressing emotions through body motions has been studied in the fields of animation and computer graphics for decades, though it was not until the beginning of 21th century that this was studied in robotics. Kulic and Croft [78] presented the first study of this kind, in which they found that robot motions may cause users to feel anxious. They also pointed out the need to design robot motions so that the interaction is more comfortable between the user and the robot. Saerbeck and Bartneck [79] proposed a method that effectively transmits different emotional states to the user by varying the velocity and acceleration of the robot. The proposed solution has been implemented in a Roomba robot and further shown to be extendible to other robots.

Beck et al. [9] proposed a method for generating emotional expressions in robots by interpolating between robot configurations which are associated with specific emotional expressions. The emotions are selected from the Circumplex model of emotions [80]. This model plays an important role in mathematically defining emotions as points in a two-dimensional space, allowing each points to correspond to different emotions. The PAD model [3] is also widely used as a tool to express emotions numerically (see section 5.3.2).

An approach has also been taken to add an offset to the known gesture behavior in terms of position and its velocity. For instance, the method proposed by Lim et al. [11] varies the final position and velocity of the base gesture on the basis of the intensity and type of emotion the humanoid robot wishes to convey. In their study, the amount of offset is varied on the basis of the features extracted from the speech signal. Similarly, Nakagawa et al.'s study [81] showed a correspondence between emotion and offset based on the Circumplex model. By adding this offset to the intermediate configuration and velocity of a given trajectory, emotional nuances are added to the trajectory. Thus, the trajectory can be parametrized in two dimensions to express the intended emotion.

Claret et al. [10] proposed using the extra degrees of freedom left over from the robot's kinematic redundancy to make the robot express emotions. This concept is based on a technique called null space control [82] (see section 5.2.2), which uses the extra degrees of freedom to perform an additional task as a sub-task even when the robot is performing its main task. This approach was applied to the humanoid robot Pepper, which was given the gesture of waving as its main task, and an emotional movement component was given as a sub-task. With this method, the robot was able to express emotions simultaneously expressed while performing the main task of hand waving. Our study examines emotional expression while performing a productive task through manipulators, which was not mentioned in the previous study. We also introduce a method for

effectively regulating the emotional movements expressed within the redundant degree of freedom.

2.6 Exploiting Kinematic Redundancy

Liégeois [83] and Klein and Huang [84] pointed out that if the number of task-required dimensions is lower than the number of robot joints, the robot is considered as redundant and it is possible to program multiple tasks, regardless of whether the robot structure is redundant. The most common method of exploiting robot redundancy is the null space of the robot's Jacobian. In this method, the potential function is projected onto the kernel of the main task. In other words, only the space of joint speeds that can be executed without affecting the main task is made available to the sub-task [83] [85]. In this approach, the first task is guaranteed to be executed, and the second task is executed only if sufficient degrees of freedom remain in the system, thus avoiding inter-task conflicts.

There are several examples of implementing secondary tasks using the Jacobian null space approach. For instance, Liégeois implemented joint limit avoidance [83], and Nemec and Zlajpah demonstrated singularity avoidance [86]. In addition, Maciejewski and Klein developed a self-collision avoidance system [87]. Hollerbach and Suh implemented torque minimization [88], and Baerlocher and Boulic demonstrated the simultaneous positioning of multiple end-effectors [89]. In a subsequent study, they demonstrated center of mass positioning for humanoids [90]. Peng and Adachi, added compliant control to the null space of the 3 Dof robot manipulator [91]. On the other hand, the work of Claret et al. [10] is the only example of a study in which emotional expression was set as a sub-task.

2.7 Evaluating Expressive Movements and Emotion

How to evaluate the physical and psychological movement of people and objects is an important concern that is fundamental to the study of emotional movement. There are many study that aim is quantify the degree of expressiveness of human motion and emotion.

About evaluation of expressive motion, there are several way to decompose human motion/gesture. For exsample, Laban Effort System [92], which invented in the 1940's by Laban as part of a dance notation indicating the how of a motion. The Laban Effort System describe motion by four-dimensional state space of Time, Weight, Space and Flow. Also the specific movement guidelines can be paraphrased as Press, Writhe, Glide, Float, Punch, Slash, Dab and Flick, respectively. This motion notation system is used not only for dance training, but also robot motion evaluation. Knight [8] have studied about emotional motion of robot with low degree of freedom using Laban system and showed even robot with low degree of freedom can effectively differentiate generated emotional motions.

Also, there is data set that contain gesture and the label describing its mood/emotion. For example, The Geneva Multimodal Emotion Portrayal (GEMEP) corpus [93] is emotion portrayals to tune and validate the emotional gesture/behavior. The GEMEP corpus was developed by the University of Geneva. It consists of more than 7,000 video of emotional motions. It represent 18 emotions in total and acted by 10 actors. Glowinski et al. [94] utilize GEMEP corpus to presents a framework for analysis of affective behavior starting with a reduced amount of visual information related to human upper-body movements.

Rather than existing gesture, it is pointed out that the mood/emotion of motion is conveyed by motion feature such as jerkiness (smoothness), kinetic energy, spatial extent, gaze and etc. In Saerbeck and Bartneck's work [79], a emotional motion pattern generation method for robot is presented. The proposed method differentiate each motion pattern by varying the amount of velocity and acceleration of the robot trajectory. This work showed that it is possible to transmit different emotional states to the user by adjusting motion feature of velocity and acceleration. Although the proposed solution has been implemented in a robot with low degree of freedom (Roomba), it

further shown to be extendible to other robots.

Besides the motion evaluation, also methods of evaluation/quantification of emotional state is crucial for social robot study. To express emotion by number, there is the Circumplex Model of Affect that is a model of emotions representation developed by James Russell [80]. In this method, emotions are represented as points in a circular space of two dimensions. These two dimensions are arousal and pleasure ,corresponding to the vertical and horizontal axis. There is 3 dimensional space model of emotion called Pleasure-Arousal-Dominance (PAD) model by Mehrabian and Russell [3]. The PAD model decompose an emotion into three elements: Arousal level, Pleasure level, and Dominance level. To measure emotions, the Self-Assessment Manikin (SAM) by Lang [5] can be used. The SAM scale is a visual questionnaire composed of three scales. Each scale is corresponding to a dimension of the PAD space. Lang conducted a comparative analysis of measurement results from both the SAM and Mehrabian and Russell's PAD scales. The correlation coefficients between the two scales were 0.94 for pleasure, 0.94 for arousal, and 0.66 for dominance [95].

Chapter 3

Adaptive generalized predictive controller and Cartesian force control for robot arm using dynamics and geometric identification

SUMMARY In the first study, an adaptive generalized predictive control (AGPC) was designed using kinematic parameters estimated by dynamic identification of the robot arm to allow the robot to perform tasks in response to changes in the grasped object at its end-effector. The performance of AGPC was compared with PID control, which is still dominant in industry, and generalized predictive control (GPC) that is for systems linearized by feedback linearization. The results showed that AGPC exhibited the best position control and external force control performance based on the parameters estimated from the dynamic identification, even when the size and mass of the grasped object changed.

3.1 Introduction

Robotic serial manipulators are now widely used for manufacture in various industrial fields. [96, 97]. Industrial manipulators are typically controlled by Proportional-Integral-Differential (PID) controllers. Often its control parameters are tuned based on a fixed robot model and fixed series of motion [98, 99]. Contrarily to the manufacture, in the case of HRI, the task/motion pattern is not always the same because robots are required to adapt its motion according to the reaction of people and context of the interaction. Moreover, in case of the robotic manipulator, it could be equipped various object to handle many kind of tasks. Therefore, the simple usage of the PID control is not suitable for HRI in general.

Besides the position control, force control/collision handling is also required for robot to operate in general environment [100]. A force controller is typically based on force sensors, which are expensive, fragile, and require frequent calibrations. Alternatively, external forces can be estimated by using accurate dynamic and kinematic models of the robotic arm. With joint position sensors and torque sensors, force control can be performed by exploiting robot's dynamic and kinematic models and estimating external forces [101]. Thus, dynamic and kinematic modeling is important for robots, and accurate models give robots to possibility for advanced HRI.

How to adapt to new reference model for MPC is an open question. Robotic arms are capable of using various types of tools. However, the dynamic characteristics of the robot vary when the robot grasp a different tool. Especially when the mass of the tool is too heavy to ignore relative to the robot's own mass. This situation will be seen more often in the future for small robotic arms used for HRI. The additional mass and length at the end-effector may result in an inaccurate

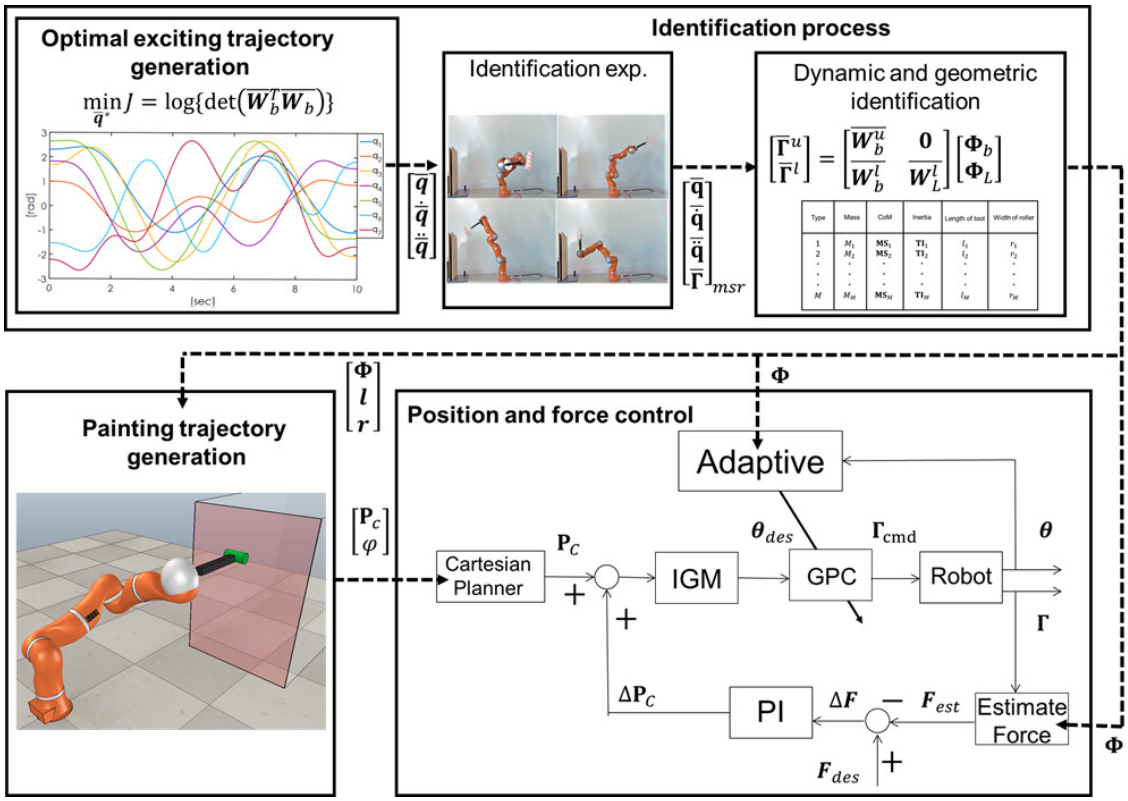

Figure 3.1: Overview of the proposed framework for adaptive predictive controllers based on dynamic and geometric identification with optimal exciting motion.

control result if the robot is only operated by controller which is deigned based on the old model and have no adaptability to the new model, such as a case of conventional PID controller. Here, the author combined Dynamic Parameter Identification (DPI) and GPC to solve this problem.

In this chapter, an adaptive predictive position-force control framework (AGPC) that uses geometric and dynamic identification was proposed. The GPC was adapted to the new reference model by adjusting control parameter λ acording to the newly identified dynamic parameters. The control parameter λ represents wight on the change of future input torque and directly modifies the cost function, which updates the law of control. This method can be used for any serial manipulator equipped with joint torque sensors or equivalent sensors such as motor-current measurement. This method enable the manipulators to accurately perform various tasks using different tools, without loss of performance. The adaptive position and force controllers were designed and evaluated for the KUKA LWR (KLWR) robotic arm. A wall painting task using different types of painting rollers is presented in this work.

An overview of the framework proposed in this paper is presented in Fig. 3.1. The explanation of notation is following; $q(\theta), \dot{q}, \ddot{q}, \Gamma$ denote joint position, velocity, acceleration, and torque. $\boldsymbol{W}$ represent regressor matrix. $\boldsymbol{\Phi}$ is a vector of dynamic parameters compose of SIP, Center of Pressure, and mass. l, r represent length and radius of the painting roller. $\mathbf{P}_c, \varphi$ denote the motion plan in the Cartesian space and swivel angle plan respectively. In section 3.2, the geometric and dynamic models of the KLWR are described. Section 3.3 presents the parameter identification of the robot using optimal exciting trajectories. Section 3.4 presents the development of the robot control system with position and force controls, and the application of the adaptation with the novel prediction scheme. In section 3.5, the experimental setup is presented, followed by the results and discussion in section 3.6.

Let it reminded that the contents of section 3.3 is a credit to the co-authors of [17], and for the

authors' credit, I have introduced a system that optimally incorporates the new robot information obtained by DPI into the reference model of the predictive controller.

3.2 Modeling of KUKA LWR

3.2.1 Geometric Model

In this study, the control command was organized by task space control where the desired end-effector motion in the task space is converted to joint angles using inverse geometric model. Therefore, an inverse geometric model of the KLWR was required.

The KUKA LWR (KLWR) is a manipulator with 7 rotational joints (Fig. 3.2). The KLWR has a load-to-weight ratio of approximately 1:1 and the total mass of the arm is approximately 17 kg with a 1.2 m work space [99]. Each link is equipped an angle position and torque sensor. Using the modified Denavit-Hartenberg (MDH) parameters (**Table 1**), the forward kinematics model was calculated using specialized software "Symoro+" [102]. The segment lengths L_1, L_2, L_3 were set using the available CAD data [41].

Table 3.1: Modified DH parameters of the KUKA LWR

i	a_i	μ_i	σ_i	α_i	d_i	θ_i	r_i
1	0	1	0	0	0	θ_1	L_1
2	1	1	0	$\pi/2$	0	θ_2	0
3	2	1	0	-$\pi/2$	0	θ_3	L_2
4	3	1	0	-$\pi/2$	0	θ_4	0
5	4	1	0	$\pi/2$	0	θ_5	L_3
6	5	1	0	$\pi/2$	0	θ_6	0
7	6	1	0	-$\pi/2$	0	θ_7	0

Contrary to the foreword kinematic model, inverse geometric model require additional effort to compute for the case of 7 DoF robot. The inverse geometric model is generally calculated using the Paul method [102]. However, given that the KLWR has 7 DoFs and only 6 Dof is required to describe a posture in 3D space, the robot is structurally redundant, meaning there are infinite options for realizing one end-effector posture.

To solve the redundancy problem in the robot, a swivel angle φ was added to the model [103] shown in Fig. 3.3. The swivel angle φ uniquely represents the position of the elbow joint (4th joint) $\mathbf{P}_e$ on a circle perpendicular to the vector of the wrist joint (6th joint) from the shoulder joint (2nd joint) $[(\mathbf{P}_s\mathbf{P}_w)]$. In Fig. 3.3, $\mathbf{P}_s, \mathbf{P}_e, \mathbf{P}_w$ define the position of the shoulder, elbow, and wrist of the robot, respectively. To mathematically describe this circle, the unit vector $\widehat{\mathbf{n}}$ of the direction from the shoulder to the wrist was defined. A local coordinate system was created from unit vectors $\widehat{\mathbf{u}}$ and $\widehat{\mathbf{v}}$. It can be noted that $\widehat{\mathbf{u}}$ was arbitrarily set to be the projection of a user-defined vector $\mathbf{b}$. The mathematical description of $\widehat{\mathbf{u}}$, $\widehat{\mathbf{v}}$, and $\widehat{\mathbf{n}}$ is found in [103].

$$\begin{aligned} \hat{\mathbf{n}} &= \frac{\mathbf{P_W}}{\|\mathbf{P_W}\|} \\ \hat{\mathbf{u}} &= \frac{\hat{\mathbf{b}} - (\hat{\mathbf{b}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}}{\left\|\hat{\mathbf{b}} - (\hat{\mathbf{b}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}\right\|} \\ \hat{\mathbf{v}} &= \hat{\mathbf{n}} \times \hat{\mathbf{u}} \end{aligned} \tag{3.1}$$

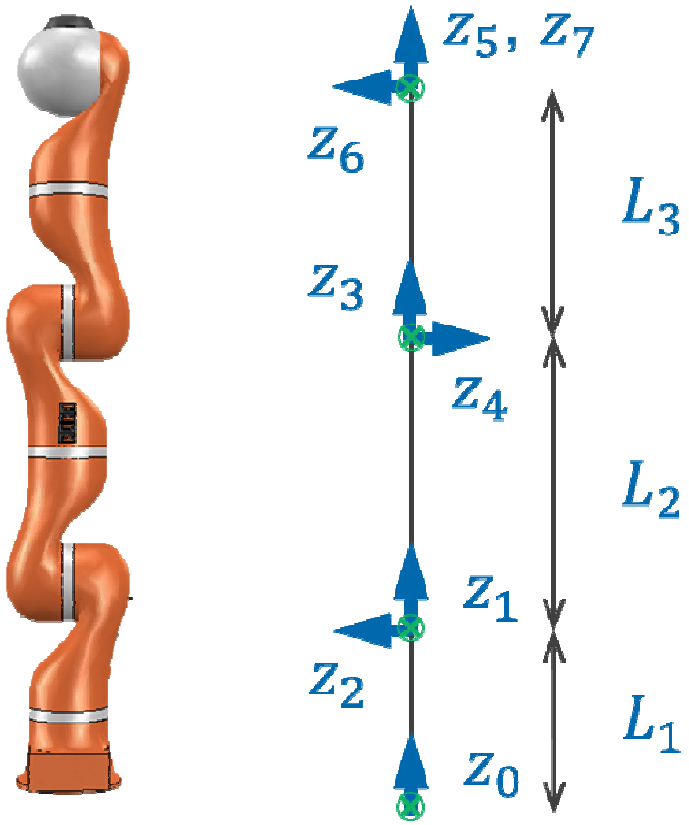

Figure 3.2: Frames and segments definitions for the KUKA LWR used in the MDH in Table 3.1 [1]

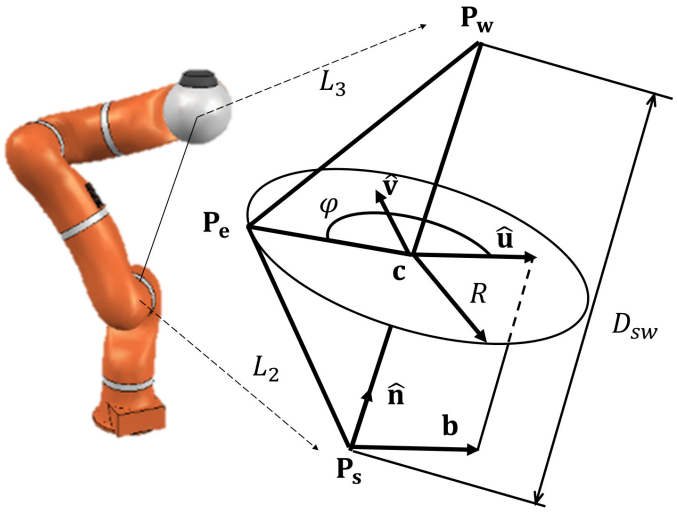

Figure 3.3: Conceptual diagram of swivel angle φ

By using these unit vectors, the position $\mathbf{P}_e$ of the elbow joint can be expressed by the swivel angle φ as follows.

$$\begin{aligned} cos(\alpha) &= \tfrac{L_1^2+L_3^2-L_2^2}{2L_1L_3} \\ \mathbf{c} &= L_1\hat{\mathbf{n}}\, cos(\alpha) \\ R &= L_1\, sin(\alpha) \\ \mathbf{P_c} &= \mathbf{c} + R(cos(\varphi)\hat{\mathbf{u}} + sin(\varphi)\hat{\mathbf{v}}) \end{aligned} \tag{3.2}$$

The Paul method was then implemented to obtain the resolution of the inverse geometric model. By using the elbow joint position $\mathbf{P}_e$ and wrist joint position $\mathbf{P}_w$ obtained by swivel angle, the inverse geometry problem can be solved and the joint angles can be uniquely obtained. detailed explanation can be found in [104, 105]. Here is brief explanation of the Paul method. Let $\mathbf{U}_{e0}$ denotes the known transformation matrices from the base to the elbow joint.

$$\begin{aligned} \mathbf{U_{e0}} &= {}^0\,\mathbf{T}_{n_e} \\ &= {}^0\,\mathbf{T}_1\,{}^1\mathbf{T}_2 \cdots {}^{n_e-1}\,\mathbf{T}_{n_e} \\ &= \begin{bmatrix} sx_e & nx_e & ax_e & Px_e \\ sy_e & ny_e & ay_e & Py_e \\ sz_e & nz_e & az_e & Pz_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3.3}$$

The transformation matrix ${}^{n-1}\mathbf{T}_n$ is the transformation matrix from the $n-1$ th frame to the n th frame. Using the Paul method, a transformation matrix for a particular links can be derived using a known transformation matrix. The Paul method can generate a new expression by multiplying ${}^{n}\mathbf{T}_{n-1}$ from the left. By multiplying Eq. (3.3) by ${}^1\mathbf{T}_0, {}^2\,\mathbf{T}_1, ..., {}^{n_e-2}\,\mathbf{T}_{n_e-1}$ from the left, we obtain the following equation.

$$\begin{aligned} \mathbf{U_{e0}} &= {}^0\,\mathbf{T}_1\,{}^1\mathbf{T}_2 \cdots {}^{n_e-1}\,\mathbf{T}_{n_e} \\ \mathbf{U_{e1}} &= {}^1\,\mathbf{T}_2 \cdots {}^{n_e-1}\,\mathbf{T}_{n_e} \\ &\vdots \\ \mathbf{U_{e n_e-1}} &= {}^{n_e-1}\,\mathbf{T}_{n_e} \end{aligned} \tag{3.4}$$

Once the transformation matrix for a particular links was obtained, the trigonometric functions of its elements were analyzed based on Shimizu's research [104] to calculate the joint angles. swivel angle provides the elbow joint posture relative to the desired end-effector position, therefore, the Paul method can be used with redundant manipulators.

3.2.2 Dynamic Model

A dynamic model is a model that represents the behavior of a robot when a certain control input (torque) is applied to the robot, while an inverse dynamic model, conversely, represents the torque required to achieve a specific robot behavior. The inverse dynamic model is needed in this study for applying feedback linearization and for evaluation of DPI. The inverse dynamic model of the KLWR was calculated through Symoro+ which is using Newton-Euler equations. The inverse dynamic model provides the joint torques vector $\mathbf{\Gamma}$ that realize a certain joint position, velocity and

acceleration [102].

$$\begin{aligned} \boldsymbol{\Gamma} &= \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \boldsymbol{\Gamma}_f + \boldsymbol{\Gamma}_{ext} \\ \boldsymbol{\Gamma}_f &= \mathrm{diag}(\dot{\mathbf{q}})\mathbf{FV} + \mathrm{diag}(\mathrm{sign}(\dot{\mathbf{q}}))\mathbf{FC} \end{aligned} \tag{3.5}$$

Where $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ represents the (7×1) vectors of the joint angles, velocities, and accelerations; $\mathbf{M}(\mathbf{q})$ (7×7) represents the robot tensor of the inertia; $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ (7×1) represents the Coriolis and centrifugal terms; and $\mathbf{G}(\mathbf{q})$ (7×1) is the gravitational term. Moreover, $\boldsymbol{\Gamma}_{ext}$ is the joint torques vector, which represents the influence of the external forces; and $\boldsymbol{\Gamma}_f$ (7×1) is the vector of the friction torques due to Coulomb and viscous-friction effects. Furthermore, $\mathbf{FC} = \begin{bmatrix} FC_1 & ... & FC_{NJ} \end{bmatrix}^T$ (7×1) is the vector containing the Coulomb's friction coefficients, and $\mathbf{FV} = \begin{bmatrix} FV_1 & ... & FV_{NJ} \end{bmatrix}^T$ (7×1) is the vector containing the viscous friction coefficients.

Here, interestingly, the equation of motion Eq. (3.5) can be expressed in linear form with respect to the dynamic (inertial) parameters in the joint frames such as SIP, Center of Mass (CoM), and mass [42] as follows:

$$\mathbf{W}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\Phi} = \boldsymbol{\Gamma} \tag{3.6}$$

where $\mathbf{W}$ (7×84) is the regressor matrix of the chains, $\boldsymbol{\Phi}$ (84×1) is the vector of standard inertial parameters to be identified. In detail, $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_1^T & \boldsymbol{\Phi}_{NL}^T & \mathbf{FC}^T & \mathbf{FV}^T \end{bmatrix}^T$. For each link i, 10 inertial parameters can be expressed in the joint frame $\boldsymbol{\Phi}_i = [M_i \ \mathbf{MS}_i^T \ \mathbf{TI}_i^T]^T$, where M_i is the mass, $\mathbf{MS}_i = \begin{bmatrix} MSX_i & MSY_i & MSZ_i \end{bmatrix}^T$ is the three-dimensional vector of the CoM, and the six-dimensional vector $\mathbf{TI}_i = \begin{bmatrix} XX_i & YY_i & ZZ_i & XY_i & XZ_i & YZ_i \end{bmatrix}^T$ that gathers the elements of the 3×3 tensor of inertia.

3.3 Identification of Robot Parameters

Generally, mechanical parameters such as mass, center of gravity, and moment of inertia of each link can be calculated for a robot by using CAD data. In reality, however, the mechanical model obtained from CAD data alone is not sufficient, because elements not included in the CAD data, such as cables and other electrical components, also affect the robot dynamics. Therefore, it is necessary to identify the mechanical model of the robot. In general, white-box modeling is considered to be the most effective method for identifying the mechanical model of a robot, because a mathematical model can be constructed based on physical laws. Therefore, the identification of the robot dynamics means the identification of the mechanical parameters consisting of mass, center of gravity, and moment of inertia.

The identification process is done with three steps.

1. The generation of the optimal exciting trajectory.
2. The identification of the robot inertial parameters.
3. The end-effector tool identification.

the identification method that is using the optimal exciting motion is described in [1]. In his study, painting roller is chosen as the implementation example. Painting rollers are tools used to paint walls and require proper force and positioning accuracy for the task. Following his example, the painting roller was similarly treated as an implementation example in this study.

From the Eq. (3.6), we can obtain the dynamical parameter $\mathbf{\Phi}$. However, in general, the regressor matrix $\mathbf{W}$ is not a full-rank matrix. Therefore, it is not possible to obtain the dynamical parameters by using the least-squares method directly. Therefore, it is necessary to use a transformed form of the dynamical parameters called the minimum dynamical parameters [43, 44]. The minimum dynamical parameters $\mathbf{\Phi}_b$ are the dynamical parameters with the minimum number of dimensions necessary to represent the entire dynamical model. Therefore, using the minimum dynamical parameters, Eq. (3.6) can be rewritten as follows.

$$\mathbf{W\Phi} = \mathbf{W}_b\mathbf{\Phi}_b = \mathbf{\Gamma} \tag{3.7}$$

When the obtained motion data is n samples, it can be written as follows.

$$\begin{bmatrix} \mathbf{W}_b(1) \\ \vdots \\ \mathbf{W}_b(n) \end{bmatrix} \mathbf{\Phi}_b = \begin{bmatrix} \mathbf{\Gamma}(1) \\ \vdots \\ \mathbf{\Gamma}(n) \end{bmatrix} \tag{3.8}$$
$$\rightarrow\ \bar{\mathbf{W}}_b\mathbf{\Phi}_b = \bar{\mathbf{\Gamma}}$$

The regressor matrix $\mathbf{W}$ is depend on the structure of the robot and delivered by Symoro+, and the symbolic approach of computing the base parameters is based on determining the independent elements of the energy functions of each link [102]. By applying the weighted least-squares method [106] on the Eq. (3.8), the minimum dynamical parameters $\mathbf{\Phi}_b$ can be uniquely identified as shown below.

$$\mathbf{\Phi}_b^* = (\bar{\mathbf{W}}_\mathbf{b}^\mathbf{T}\mathbf{P}\bar{\mathbf{W}}_\mathbf{b})^{-1}\mathbf{P}\bar{\mathbf{W}}_\mathbf{b}^\mathbf{T}\bar{\mathbf{\Gamma}}$$
$$\mathbf{P} = \begin{bmatrix} \mathbf{S}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{S}_{N_j} \end{bmatrix} \tag{3.9}$$
$$\mathbf{S}_j = \frac{\mathbf{I}_{N_e}}{\sigma_j}$$
$$\sigma_j = \frac{\|\bar{\mathbf{\Gamma}}_j - \bar{\mathbf{W}}_b^j\mathbf{\Phi}_b^j\|}{N_e - N_B^j}$$

Where, N_e, N_B^j is the number of samples and number of minimum dynamical parameters for j th link respectively. $\mathbf{I}_{N_e}$ is identity matrix that size of $N_e \times N_e$. $\bar{\mathbf{\Gamma}}_j, \bar{\mathbf{W}}_b^j$, and $\mathbf{\Phi}_b^j$ is the parts corresponding to j th link. By using this weight matrix $\mathbf{P}$, the least-squares method can be implemented to account for differences in the magnitude of joint torques.

About the optimal exciting trajectories, in conducting identification experiments, it is a very important issue to decide what kind of motion to use for identification. The motion that guarantees the identification accuracy is called Optimal Exciting Trajectory (OET), and such a characteristic is called excitation. This OET is known to reduce the following cost function J_d in previous studies [1].

$$J_d = -log\{det(\mathbf{W}_b^T\ \mathbf{W}_b)\} \tag{3.10}$$

The trajectories were parameterized by Q-Spline method and optimized to minimize the cost function of the Eq. (3.10) using quadratic programming (QP). Here, the motion was optimized so that the joint angles, velocities, accelerations, and input torques were within limits, and self-collisions were avoided.

In general, the minimum dynamical parameters $\mathbf{\Phi_b}$ are sufficient to represent the dynamic model. However, complete dynamic parameter $\mathbf{\Phi}$ is needed for adapting reference model of predictive controllers. Therefore, it is necessary to identify the total dynamic parameters by QP. In order to identify the total dynamic parameters, as in the case of the minimum dynamic parameters, the measured values of each joint angle and torque, as well as the total dynamic parameters from CAD data to be used as a reference, are required. The cost function for QP is shown below.

$$\begin{gathered} J_\Phi = w_1 \left\| \mathbf{W}_p\mathbf{\Phi} - \mathbf{\Gamma}_p \right\|^2 + w_2 \left\| \mathbf{\Phi} - \mathbf{\Phi}_{CAD} \right\|^2 \\ \mathbf{W}_P = \mathbf{P}\mathbf{W}_B \\ \mathbf{\Gamma}_p = \mathbf{P}\mathbf{\Gamma} \end{gathered} \tag{3.11}$$

Where $\mathbf{\Phi}_{CAD}$ is total dynamic parameters from CAD data. w_1 and w_2 are weights for the first and second terms respectively.

3.3.1 End-Effector Tool Identification

The robotic arm can attach various tools to the end-effector to perform tasks. Moreover, the physical change of the end-effector generates geometric and dynamic changes in the system. The geometric parameters of the tool were identified using its SIP, as described in [1, 45]. The dynamic identification of the end-effector was achieved using two sets of grouped equations which represent the trajectories without and with the end-effector presented by [107].

$$\begin{bmatrix} \overline{\mathbf{\Gamma}}^u \\ \overline{\mathbf{\Gamma}}^l \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{W}}_b^u & 0 \\ \overline{\mathbf{W}}_b^l & \overline{\mathbf{W}}_L^l \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_b \\ \mathbf{\Phi}_L \end{bmatrix} \tag{3.12}$$

Here, $\overline{\mathbf{\Gamma}}^u$, $\overline{\mathbf{\Gamma}}^l$, $\overline{\mathbf{W}}_b^u$, $\overline{\mathbf{W}}_b^l$ represent the measured joint torques vectors and minimum regressor matrices that correspond to the KLWR is unloaded and loaded cases, respectively.

The geometric identification of end-effector is based on the relationship between the dynamic parameters and geometric parameters [48]. In this study, the length and width of a paint roller are identified using the look-up table. The table contains dynamic and geometric parameters for different sizes of painting rollers. A principal component analysis (PCA) of the dynamic parameters is conducted for the classification of the tool at end-effector [108]. In other words, after identifying the dynamic parameters of the grasped tool that are not known, the error norm of all the dynamic parameters in the lookup table is calculated, and the one with the smallest value is considered as the painting tool being grasped. Then, the geometric model of the robot is updated with the corresponding geometric parameters.

In this work, the main task was the painting of flat surfaces with smooth and continuous trajectories. In particular, the robot manipulator traced a painting trajectory in the shape of "W" on the wall using painting rollers attached to the end-effector. The size of the "W" trajectory was

Type	Mass	CoM	Inertia	Length of tool	Width of roller
1	M_1	$\mathbf{MS}_1$	$\mathbf{TI}_1$	l_1	r_1
2	M_2	$\mathbf{MS}_2$	$\mathbf{TI}_2$	l_2	r_2
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
M	M_M	$\mathbf{MS}_M$	$\mathbf{TI}_M$	l_M	r_M

Figure 3.4: Example of look-up table

approximately 0.2 m × 0.2 m. The painting roller was applied to the wall with a constant force of 30 N. In this work, two painting rollers (big and small) specified in Table 3.4 were used. The trajectory for the painting task was generated in the Cartesian space by the optimization process, ensuring a smooth trajectory and proper initialization of painting task. The Cartesian trajectory applied in this study was developed by T. Katsumata et al. [1].

The optimization for the motion trajectory minimizes a cost function constituted by the norm of the robot joint jerks and the duration of the desired trajectory. Simultaneously, the optimization maximizes the work area (in the case of a painting surface by the horizontal and vertical distances traveled by the roller) during the motion. Thus it is ensuring dynamic and kinematic viability in the robot. The trajectory is defined in 6 DoFs; $P_C = \begin{bmatrix} \overline{P}_X & \overline{P}_Y & \overline{P}_Z & \overline{\theta}_X & \overline{\theta}_Y & \overline{\theta}_Z \end{bmatrix}^T$. Here, $\overline{\mathbf{P}}_{X,Y,Z}$ are 3D position of the end-effector and $\theta_{\mathbf{X},\mathbf{Y},\mathbf{Z}}$ are the orientations represented by Euler angle. The path planning optimization problem can be expressed as the solution of $\mathbf{X}_{opt} = \begin{bmatrix} T_c & \varphi & Z_u & Z_l & Y_0 & \Delta Y \end{bmatrix} \in \Re^6$, where the T_c represents the trajectory duration time, φ is the swivel angle, Z_u and Z_l are the upper and lower vertical positions of the roller, respectively. And Y_0 is the initial horizontal roller position, and ΔY is the horizontal distance traveled by the roller. The velocity and acceleration at the point of turns were set as zero. The trajectory was interpolated using 5^{th} order B-splines. Through this parameterization of the trajectory, the adaptation of robot motion to the different tools is automated.

These aforementioned work are developed by co-authors of previous work [17] and the author's primal work is to unified those DPI and geometric adaptation method into predictive control system and formed one adaptive robotic manipulator motion operation scheme. Now the controller have to adapt these newly identified parameters. In particular, GPC and AGPC are based on the dynamic models, and updating the parameters of the robot is a crucial task for the controllers if the characteristics of the robot varies with different types of rollers attached to its end-effector. In the next section, the proposed AGPC that exploits the identified results is presented.

3.4 Robot Control System (Position and Force Controllers)

The control system for painting wall with applying 30 N force was designed according to the scheme as shown in Fig. 3.6. The painting motion $\mathbf{P}_c$ is generated by planner described in section 3.3.1. The Cartesin trajectory of the end-effector is converted into joint angle trajectories $\boldsymbol{q}_{des}$ using inverse geometric model (IGM) described in section 3.2. The controller receive desired joint angle and generate input torque $\boldsymbol{\Gamma}_{con}$ to realize the target joint angle trajectory. The position/torque sensors inside of the joints return the current joint angles $\boldsymbol{q}$ and torque $\boldsymbol{\Gamma}$ working on the joints. For the force control, reaction force on the wall is estimated based on the torque information and compensation variance of the end-effector position $\Delta\mathbf{P}_c$ is computed using proportional-integral (PI) control method and added to the next Cartesian trajectory command. The position controllers were developed using the proportional-Differential (PD) with gravitational compensator,

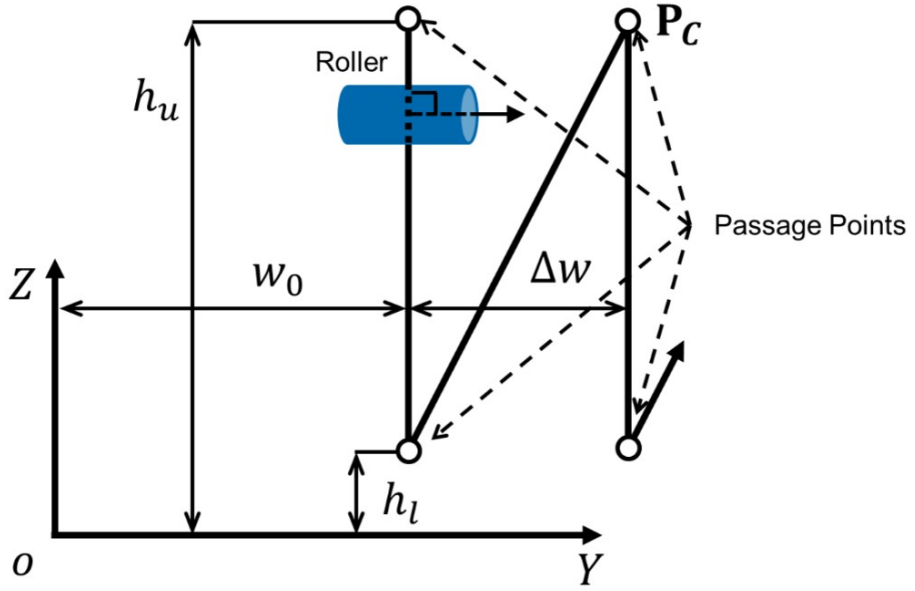

Figure 3.5: Parameterization of the trajectory of the painting task

GPC, and AGPC. Moreover, the AGPC and GPC make use of the models identified in the previous sections to adapt its reference model.

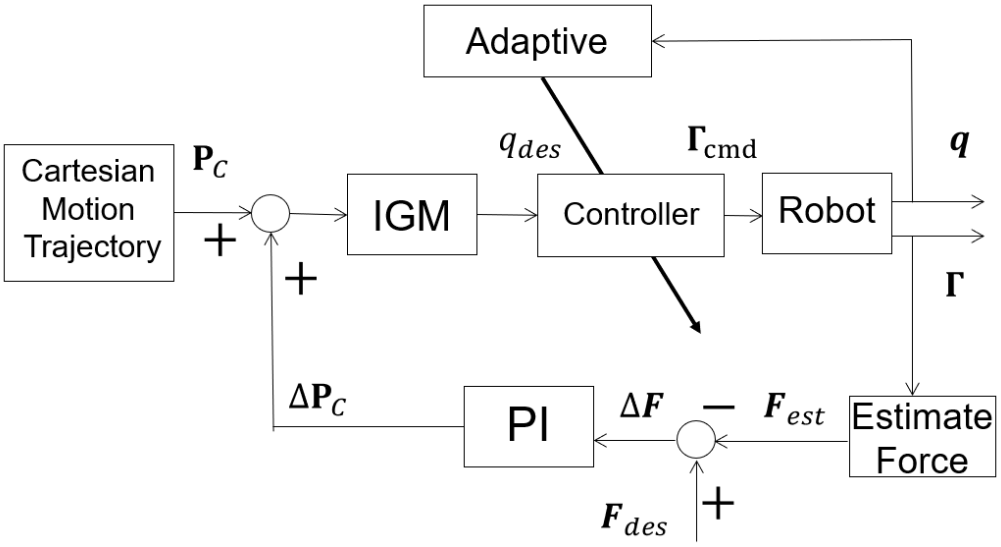

Figure 3.6: Adaptive control system for robot force and position control based on fast dynamic and geometric identification

3.4.1 External Force Estimation

Applying the DPI to the robotic manipulator and of its end-effector, the external forces (reaction force from the wall) could be estimated from the measured joint torques using the following equations.

$$\begin{aligned} \mathbf{\Gamma} &= \widehat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \widehat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) + \widehat{\mathbf{G}}(\mathbf{q}) + \mathbf{\Gamma}_{ext} \\ &\rightarrow \mathbf{\Gamma} = \mathbf{\Gamma}_{dyn} + \mathbf{\Gamma}_{ext} \\ \mathbf{F}_{est} &= (\mathbf{J}^T)^+ \mathbf{\Gamma}_{ext} = (\mathbf{J}^T)^+ (\mathbf{\Gamma} - \mathbf{\Gamma}_{dyn}) \end{aligned} \tag{3.13}$$

where $\mathbf{J}$ is a (6×7) robot Jacobian matrix and superscript + denotes pseudo inverse. $\widehat{\mathbf{M}}(\mathbf{q})$ (7×7) is the identified robot inertia matrix, $\widehat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})$ (7×1) is the vector that contains the identified Coriolis and centrifugal terms. $\mathbf{\Gamma}$ and $\mathbf{\Gamma}_{dyn}$ are the measured and estimated joint torques, respectively. $\mathbf{F}_{est}$ (6×1) is the estimated vector of the external forces in Cartesian space. The symbolic model of each dynamical components are generated through Symoro+. It is symbolic because one easily adapt to new dynamic parameters once the robot setting have changed.

3.4.2 Force Control

The control scheme is composed of two embedded control loops inspired by [109]. The outer loop controls the external forces, and the inner loop controls the pose of the Cartesian end-effector as shown in Fig. 3.6. The additional displacement command to the end-effector in 3D position, is given by:

$$\Delta P_c = \mathbf{K}_p^f (\mathbf{F}_d - \mathbf{F}_{est}) + \mathbf{K}_i^f \int_{T_0}^{T} (\mathbf{F}_d - \mathbf{F}_{est}) \tag{3.14}$$

where ΔP_C is the additional displacement reference signal in Cartesian space. $\mathbf{F}_d$ is the desired external force. $\mathbf{K}_p^f$ and $\mathbf{K}_i^f$ are the proportional and the integral gains, respectively. And T_0 and T represent the initial time and current time, respectively. The gains are set as follows by refereeing previous work of Katsumata [108].

$$\mathbf{K}_p^f = diag(1.0 \times 10^{-4}, 0, 0)\ , \quad \mathbf{K}_i^f = diag(2.0 \times 10^{-3}, 0, 0) \tag{3.15}$$

Here, $diag(x)$ denotes the diagonal matrix with diagonal elements x.

3.4.3 Position Control

Besides the predictive controllers, several traditional control techniques were implemented for the comparison. First, a PD control with a gravitational compensator was presented. Second, a PD control with feedforward compensation was explained. followed by GPC, and AGPC is proposed.

a) PD Control & Gravitational Compensator

The PD control with a gravitational compensator was designed by tuning the gains as a classical joint space PD controller [110] with P gain [500, 1000, 300, 500, 100, 100, 0.5] and D gain [15, 15, 10, 15, 5, 5, 0.01] for links 1-7, respectively.

Table 3.2: Gain values for the PD controller for KLWR

i-th link	1	2	3	4	5	6	7
K_p	500	1000	300	500	100	100	0.5
K_d	15	15	10	15	5	5	0.01

The gravitational torque was computed using the dynamic parameters identified in the previous part. Given that the dynamic models of the robot and painting tool were already identified, the PD control could be expressed as:

$$\Gamma_d = \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_d(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \widehat{\mathbf{G}}(\mathbf{q}) \tag{3.16}$$

where $\mathbf{K}_p$ and $\mathbf{K}_d$ are $(N_J \times N_J)$ diagonal matrices for the experimentally-tuned proportional and derivative gains. $\mathbf{q}_d$ (7×1) and $\dot{\mathbf{q}}_d$ (7×1) are the vectors of the desired and measured joint angles, respectively. $\dot{\mathbf{q}}_d$ and $\dot{\mathbf{q}}$ were obtained by the first-order backward difference with respect to $\mathbf{q}_d$ and $\mathbf{q}$. And $\widehat{\mathbf{G}}(\mathbf{q})$ $(N_J \times 1)$ is the vector of gravitational terms obtained using the identified inertial parameters.

b) PD Control & Feedforward

The PD control with feedforward was applied, as expressed in below. Typically, the plant with previous determined dynamics and a predefined trajectory was controlled using feedback and feedforward controls. The feedforward torques were computed using the identified dynamic parameters of the robot. P gains and D gains were tuned using the same values of the PD gravitational compensator control. $\dot{\mathbf{q}}_d$ was obtained by the first-order backward difference with respect to $\mathbf{q}_d$.

$$\mathbf{\Gamma}_d = \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_d(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \widehat{\mathbf{M}}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \widehat{\mathbf{C}}(\mathbf{q}_d, \dot{\mathbf{q}}_d) + \widehat{\mathbf{G}}(\mathbf{q}_d) \tag{3.17}$$

c) Generalized Predictive Control (GPC)

The GPC scheme was applied to the robot arm using feedback linearization. GPC is a method of model predictive control (MPC) proposed by Larke et al. in 1987 [111]. GPC is expected to improve the performance of the position control with various condition such as different posture and velocity. Before describing the GPC design, the feedback linearization method is presented. In this study, the same feedback linearization technique as that in the common case of computed torque control (CTC) was used [23] to calculate suitable models for the predictive controllers. GPC is linear model predictive control, therefore, non-linear system needed to be described in linear model in order to be applied GPC. The robotic arm dynamics could be linearized and decoupled by nonlinear feedback, when input is set as below with new input control vector $\boldsymbol{w}$.

$$\mathbf{\Gamma} = \widehat{\mathbf{M}}(\mathbf{q})\mathbf{w} + \widehat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) + \widehat{\mathbf{G}}(\mathbf{q}) \tag{3.18}$$

$\widehat{\mathbf{M}}(\mathbf{q})$, $\widehat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})$, and $\widehat{\mathbf{G}}(\mathbf{q})$ are the estimates of $\mathbf{M}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, $\mathbf{G}(\mathbf{q})$, respectively; and $\mathbf{w}$ is the new input control vector. Assuming that $\widehat{\mathbf{M}}(\mathbf{q}) = \mathbf{M}(\mathbf{q})$, $\widehat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, and $\widehat{\mathbf{G}}(\mathbf{q}) = \mathbf{G}(\mathbf{q})$ the problem is reduced to a linear and decoupled double-integrators system, where n is the number of DoFs of the robot [47].

$$\ddot{\mathbf{q}} = \mathbf{w} \tag{3.19}$$

This Eq. (3.19) shows that each link is independent and angler acceleration is linearly proportional to new input $\boldsymbol{w}$. Thus, linear control theory such as the GPC can be applied. Predictive control can be summarized as follows [112, 33].

- The definition of a numerical model of the system to predict future behavior.
- The minimization of a quadratic cost function over a finite future horizon using future predicted errors.
- Applying the first value of the computed optimal input torque to the system.
- the iteration of the entire procedure for the next sampling period, according to the "receding horizon" strategy [2]

For the GPC, the plant was modeled using Controlled Auto-Regressive Integrated Moving Average (CARIMA) model which is shown in Eq (3.20). Even though CARIMA model is single-input-single-output (SISO) model, thanks to the feedback linearization, the original dynamics is decoupled and individual link can treated as SISO system.

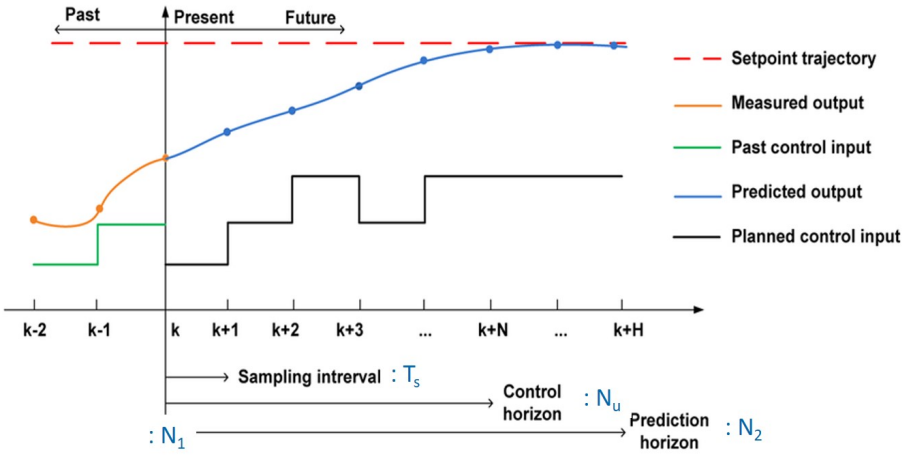

Figure 3.7: Concept of receding horizon strategy [2]

$$
\begin{aligned}
A(z^{-1})y(t+1) &= B(z^{-1})u(t) + \frac{C(z^{-1})}{\Delta}\xi(t) \\
A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_{n_a} z^{-n_a} \\
B(z^{-1}) &= 1 + b_1 z^{-1} + b_2 z^{-2} + ... + b_{n_b} z^{-n_b} \\
C(z^{-1}) &= 1 + c_1 z^{-1} + c_2 z^{-2} + ... + c_{n_c} z^{-n_c}
\end{aligned}
\tag{3.20}
$$

Where z^{-1} is a one-step backward operator, and $\Delta = 1 - z^{-1}$. $A(z^{-1})$ and $B(z^{-1})$ polynomials were obtained by discretizing the linearized model of Eq. (3.19), and $C(z^{-1})$ is the polynomial of coefficients that represent the white noise in the system. $u(t)$ and $y(t)$ are the plant input and output respectively, and $\xi(t)$ is a centered Gaussian white noise. This model gives one step ahead prediction. The reason why CARIMA model is useful is that it incorporates a disturbance estimation. Therefore, it can give unbiased prediction in the steady-state, irrespective of some model uncertainty.

Traditionally, to obtain the prediction equation, Diophantine equation is solved. However, the solution of Diophantine equation is somewhat obscure so the matrix notation is used here following [113]. The ideal condition has no noise so let us $\xi(t) = 0$, $C(z^{-1}) = 1$. Let us rewrite Eq. (3.20) as below.

$$
\begin{aligned}
& A_d(z^{-1})y(t) = B_d(z^{-1})\Delta u(t) \\
\rightarrow \quad & y(t+1) + A_1 y(t) + ... + A_{n_a} y_{t-n_a+1} = b_1 \Delta u(t) + ... + b_{n_b} \Delta u(t - n_b + 1)
\end{aligned}
\tag{3.21}
$$

Where

$$
\begin{aligned}
A_d(z^{-1}) &= \Delta A(z^{-1}) = 1 + A_1 z^{-1} + ... + A_{n_a} z^{-n_a} \\
B_d(z^{-1}) &= b_1 z^{-1} + b_2 z^{-2} + ... + b_{n_b} z^{-n_b}
\end{aligned}
\tag{3.22}
$$

And series of one step predictions towards $N_2 - th$ step can be expressed using matrix.

$$\mathbf{C}_A \begin{bmatrix} y(t+1) \\ \vdots \\ y_(t+N_2) \end{bmatrix} + \mathbf{H}_A \begin{bmatrix} y(t) \\ \vdots \\ y(t-(n_a+1)) \end{bmatrix} = \mathbf{C}_b \begin{bmatrix} \Delta u(t) \\ \vdots \\ \Delta u_(t+N_2-1) \end{bmatrix} + \mathbf{H}_b \begin{bmatrix} \Delta u(t-1) \\ \vdots \\ \Delta u(t-(n_b+1)) \end{bmatrix}$$

$$\rightarrow \quad \mathbf{C}_A \widehat{\mathbf{y}} + \mathbf{H}_A \mathbf{y} = \mathbf{C}_b \tilde{\mathbf{u}} + \mathbf{H}_b \mathbf{u}$$

$$\rightarrow \quad \widehat{\mathbf{y}} = \mathbf{C_A^{-1} C_b} \tilde{\mathbf{u}} + \mathbf{C_A^{-1} H_b u} - \mathbf{C_A^{-1} H_A y} \tag{3.23}$$

Where

$$\mathbf{C}_A = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ A_1 & 1 & 0 & \dots & 0 \\ A_2 & A_1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ A_{N_2-1} & A_{N_2-2} & A_{N_2-3} & \dots & 1 \end{bmatrix}$$

$$\mathbf{H}_A = \begin{bmatrix} A_1 & A_2 & \dots & \dots & \dots & A_{n_a} \\ A_2 & A_3 & \dots & \dots & A_{n_a} & 0 \\ \vdots & \vdots & \vdots & \iddots & \vdots & \vdots \\ A_{N_2} & \dots & A_{n_a} & 0 & \dots & 0 \end{bmatrix}$$

$$\mathbf{C}_b = \begin{bmatrix} b_1 & 0 & 0 & \dots & 0 \\ b_2 & b_1 & 0 & \dots & 0 \\ b_3 & b_2 & b_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ b_{N_2} & b_{N_2-1} & b_{N_2-2} & \dots & b_1 \end{bmatrix} \tag{3.24}$$

$$\mathbf{H}_b = \begin{bmatrix} b_2 & b_3 & \dots & \dots & \dots & b_{n_b} \\ b_3 & b_4 & \dots & \dots & b_{n_b} & 0 \\ \vdots & \vdots & \vdots & \iddots & \vdots & \vdots \\ b_{N_2+1} & \dots & b_{n_b} & 0 & \dots & 0 \end{bmatrix}$$

$$\tilde{\mathbf{u}} = [\Delta u(t), \cdots, \Delta u(t+N_u-1)]^T$$

$$\widehat{\mathbf{y}} = [\widehat{y}(t+N_1), \cdots, \widehat{y}(t+N_2)]^T$$

The optimal predictor Eq. (3.25) was configured with the matrix about the past states, present, and future actions which are represented by **if**, **ih**, and **G**, respectively.

$$\widehat{\mathbf{y}} = \mathbf{if}\, \mathbf{y} + \mathbf{ih}\, \mathbf{u} + \mathbf{G}\, \tilde{\mathbf{u}}$$

$$\rightarrow \quad \widehat{\mathbf{y}} = \mathbf{H_f} + \mathbf{G}\, \tilde{\mathbf{u}} \tag{3.25}$$

Where

$$\mathbf{if} = \mathbf{C_A^{-1} H_A}$$

$$\mathbf{ih} = \mathbf{C_A^{-1} H_b}$$

$$\mathbf{G} = \mathbf{C_A^{-1} C_b} \tag{3.26}$$

$$\mathbf{H_f} = \mathbf{if}\, \mathbf{y} + \mathbf{ih}\, \mathbf{u}$$

Each element of the **G** matrix represents the dynamic response of the system. The control signal Δu was obtained by minimizing of the quadratic cost function shown in Eq. (3.27);

$$J = \sum_{j=N_1}^{N_2}([\hat{y}(t+j|t) - w(t+j)]^2) + \sum_{j=N_1}^{N_u}(\lambda(j)[\Delta u(t+j-1)]^2) \tag{3.27}$$

Where N_1 and N_2 define the range of the prediction output, N_u defines the control horizon, λ is a control weighting factor, *r(k)* is the reference value, *y(k)* is the prediction output value obtained by solving the Diophantine equation, and Δu is the control signal. Moreover, Eq. (3.27) can be expressed by a vector with the following form:

$$J = (\widehat{\mathbf{y}}(t) - \mathbf{r}(t))^T(\widehat{\mathbf{y}}(t) - \mathbf{r}(t)) + \Delta\mathbf{U}(t)^T\Lambda\Delta\mathbf{U}(t) \tag{3.28}$$

Where

$$\begin{aligned} \mathbf{r} &= [r(t+N_1), \cdots, r(t+N_2)]^T \\ \Lambda &= diag[\lambda(1), \cdots, \lambda(N_u)] \\ \Delta\mathbf{U}(t) &= [\Delta u(t), \cdots, \Delta u(t+N_u-1)]^T \end{aligned} \tag{3.29}$$

By substituting Eq. (3.25) into Eq. (3.28) and differentiating the equation with respect to $\Delta\mathbf{U}(t)$ then organize each term with $dJ = 0$, the control law $\tilde{u}_{opt}$ that minimizes the evaluation function in Eq. (3.30) was obtained as follows:

$$\tilde{u}_{opt} = \mathbf{N}(\mathbf{r} - \mathbf{H_f}) \tag{3.30}$$

Where

$$\begin{aligned} \tilde{u}_{opt} &= [\Delta u(t)_{opt}, \cdots, \Delta u(t+N_u-1)_{opt}]^T \\ \mathbf{N} &= [\mathbf{G}^T\mathbf{G} + \Lambda]^{-1}\mathbf{G}^T \end{aligned} \tag{3.31}$$

The receding horizon principle assumes that only the first value of the optimal control series $\tilde{u}_{opt}$ is applied, thus for the next step, the procedure is repeated.

The design was therefore implemented adjusting T_S, N_1, N_2, N_u and λ to satisfy the required input/output behavior, i.e., fastest response that is consistent with the stability requirements. With this control strategy, a two-DoF reference signal tracking (RST) controller was obtained, and the procedure is described in [114], [115]. The resulting GPC was then synthesized as a two-DoF controller, as shown in Eq. (3.32) [116, 114].

$$S(z^{-1})\Delta u(t) = -R(z^{-1})y(t) + T(z)r(t) \tag{3.32}$$

The controller was calculated using synthesized polynomials $R(z^{-1})$, $S(z^{-1})$ and $T(z^{-1})$ which is shown in Fig. 3.8.

d) Adaptive GPC

The AGPC is proposed and implemented for position control, and the structure is presented in Fig. 3.8. In the real-time robot manipulation case, it may be difficult to obtain reliable angular acceleration information because the measurement noise on the joint angle is amplified when

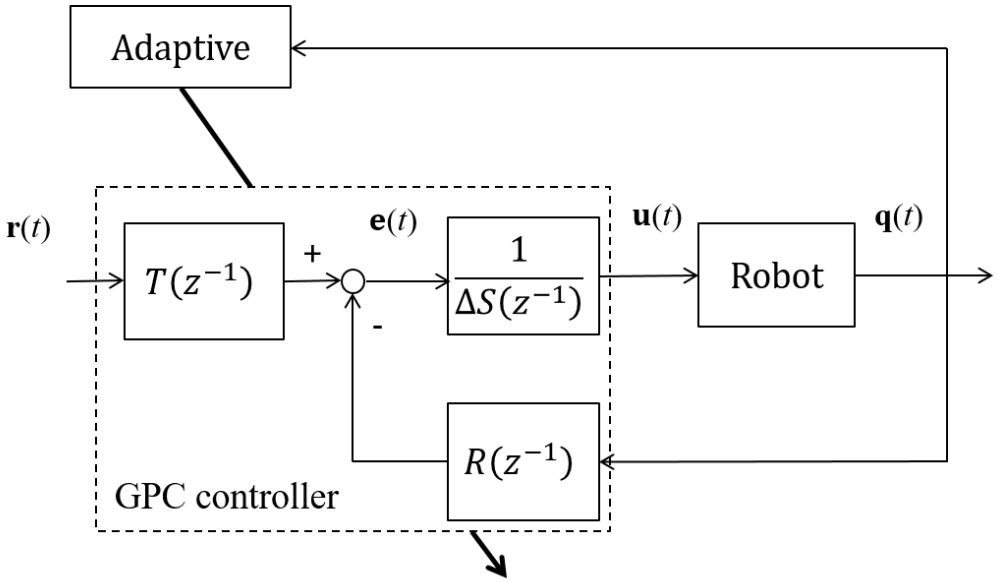

Figure 3.8: Scheme for the AGPC

second-order derivative is applied to it. Without accurate acceleration information, advanced control techniques such as CTC could not be applied to the system. Consequently, the adaptation of the GPC was designed using the dynamic information of the robot. According to Khalil and Dombre [47], the approximate dynamic model of robot arm links can be expressed as a linear second order differential equation:

$$\Gamma_j = m_j \ddot{q}_j + F_{vj} \dot{q}_j + \gamma_j \tag{3.33}$$

Where m_j=M_{jjmax} is the maximum magnitude of element M_{jj} of the inertial matrix of the robot, γ_j represents a disturbance torque, and F_{vj} is the coefficient of viscous friction. The GPC is then designed based on the model Eq. (3.33) as same as Eq. (3.19). Contrarily to the Eq. (3.19), the inertial characteristics of the robot m_j vary in accordance with the motions of the robot in Eq. (3.33). Thus, therefore, to cope with model error and stabilize the GPC, updating polynomials $R(z^{-1})$, $S(z^{-1})$, and $T(z^{-1})$ was attempted. It was achieved by adjusting the GPC design parameter λ and final prediction horizon N_2.

The controller directly adapts the control parameters; weight for control variation λ and the predictive window for the control design N_2, and that leads to the adaptation of controller polynomials $R(z^{-1})$, $S(z^{-1})$, and $T(z^{-1})$. The optimal values for these control parameters are determined based on the stability analysis of gain margin and phase margin. The analysis is done by piloting bode diagram of the control loop and converging it to "block diagram" which is shown in Fig. 3.9 and Fig. 3.10 [116]. The phase margin is required to be larger than 45°, and the gain margin is required to be larger than 6 dB, to be selected as the optimal controller. This process validated robustness criteria, which involves checking the direct and complementary sensibilities in Eq. (3.34). Bode and black analyses reveal the relationship between the measurement of the direct sensitivity σ_d, and the complementary sensitivity σ_c. The first represents the relationship between the output and the noise measurement, and the second represents the relationship between the output and the perturbation in the robot. The stability analysis is conducted every 0.1 s to determine the optimal λ for the GPC.

$$\sigma_d = \frac{A(z^{-1})\Delta S(z^{-1})}{A(z^{-1})\Delta S(z^{-1})+(z^{-1})B(z^{-1})R(z^{-1})}$$
$$\sigma_c = \frac{(z^{-1})B(z^{-1})R(z^{-1})}{A(z^{-1})\Delta S(z^{-1})+(z^{-1})B(z^{-1})R(z^{-1})} \tag{3.34}$$

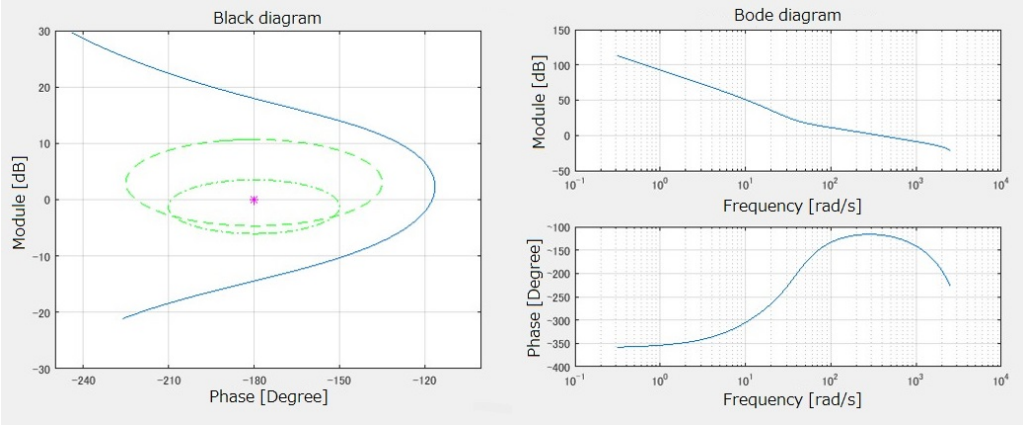

Figure 3.9: Stability analysis based on the gain/phase margin (stable). The diagram about the phase of the Bode diagram and the diagram about the gain are integrated into one diagram as a black diagram, which displays the gain margin and phase margin in an easy-to-understand manner. The larger ellipses in green represent a phase margin of 45 degrees and a gain margin of 6 decibels.

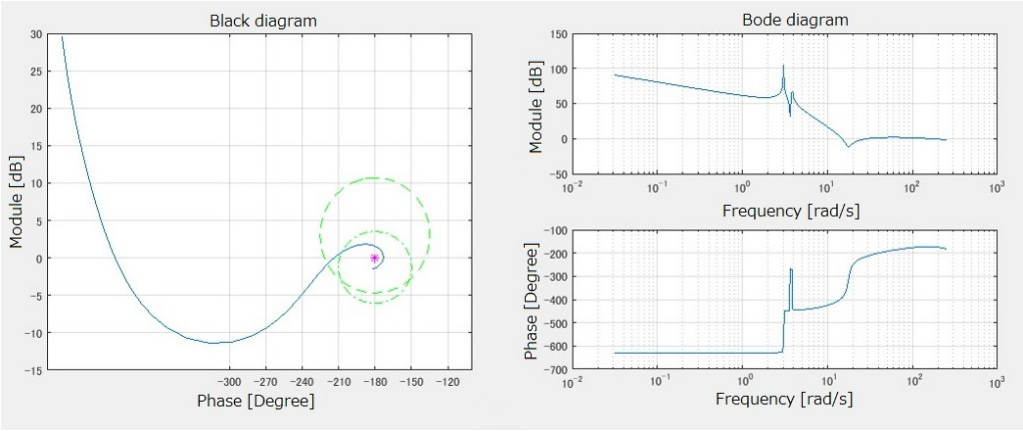

Figure 3.10: Stability analysis based on the gain/phase margin (unstable). The diagram about the phase of the Bode diagram and the diagram about the gain are integrated into one diagram as a black diagram, which displays the gain margin and phase margin in an easy-to-understand manner. The larger ellipses in green represent a phase margin of 45 degrees and a gain margin of 6 decibels.

The optimal λ is selected based on the λ that yields the best stability response. The corresponding λ is selected as the new optimal λ until the next time-period. The algorithm is set to select the best λ from 20 sets in the range of $0.001 \sim 5.0 \times 10^6$ as shown in Table 3.3.

Besides λ, according to the analysis by Ogata [117], the final predict horizon N_2 is preferred to be small, as it has a direct influence on the computational cost and delay of the system response. On the other hand, a larger N_2 generally results in an increased system stability. In this study, the initial value of N_2 was set as a minimum with tolerable value of the gain and phase margins. If the λ cannot provide system stability, the algorithm immediately changes the final window of

Table 3.3: Choices of λ for GPC design

#	1	2	3	4	5
λ	1.0×10^{-3}	5.0×10^{-3}	1.0×10^{-2}	5.0×10^{-2}	1.0×10^{-1}
#	6	7	8	9	10
λ	5.0×10^{-1}	1.0	5.0	1.0×10^{1}	5.0×10^{1}
#	11	12	13	14	15
λ	1.0×10^{2}	5.0×10^{2}	1.0×10^{3}	5.0×10^{3}	1.0×10^{4}
#	16	17	18	19	20
λ	5.0×10^{4}	1.0×10^{5}	5.0×10^{5}	1.0×10^{6}	5.0×10^{6}

predictor N_2 to bigger value, and a new optimal λ is searched.

The adaptive algorithm is summarized below.

1. Calculate the inertial matrix of the robot at the initial pose using the angle information and identified dynamic parameters. Then build the new robot models for each link using the dynamic model Eq. (3.33) of the joints.

2. To determine the first optimal GPC, the prediction horizon N_1, N_2, the control horizon N_u, and the sample time T_s is chosen based on the referring Ogata's research [117].

3. By using 20 choices of λ in Table 3.3, configure GPCs and conduct stability analysis using black diagram Fig. 3.9 to chose best GPC.

4. Execute the control process (Eq. (3.30)) with best GPC polynomials $R_0(z^{-1})$, $S_0(z^{-1})$, $T_0(z^{-1})$ as initial configuration.

5. Every 0.1 second, the new approximated dynamic model of joints Eq. (3.33) at the current pose is calculated to update the reference model.

6. Decide optimal control parameter λ for the new reference model by repeating the step 3 and if the stability is not sufficient, increase prediction horizon N_2.

7. Return to the step 5 and Contrite.

The stability analysis is conducted over the entire duration of the task, and the optimal λ is selected by the algorithm above. If the algorithm cannot determine the optimal λ that can yield a sufficient system, it immediately changes the final window of the predictor N_2 and starts searching for a new λ.

The main differences between the GPC + feedback linearization and AGPC are the presence of the linearizer and the adaptive method used to develop the on-line optimization for the model error. The GPC + feedback linearization uses a strict linearizer to reduce the controlled system, as shown in Eq. (3.19). However, the strict linearizer is sensitive to model errors and jerky acceleration signal. Thus, these elements affects the entire linearizer, and the GPC decrease the control performance. On the other hand, the AGPC is designed to be stable at all times by analyzing the system responses throughout the duration of the task. Moreover, it guarantees a certain range of robustness against model errors and noisy inputs. Thus, the AGPC is less sensitive to the model errors than the GPC. In addition, the AGPC is an auto tuning system; thus, the user is not required to determine optimal set of design control parameters such as in the case of the GPC + feedback linearization.

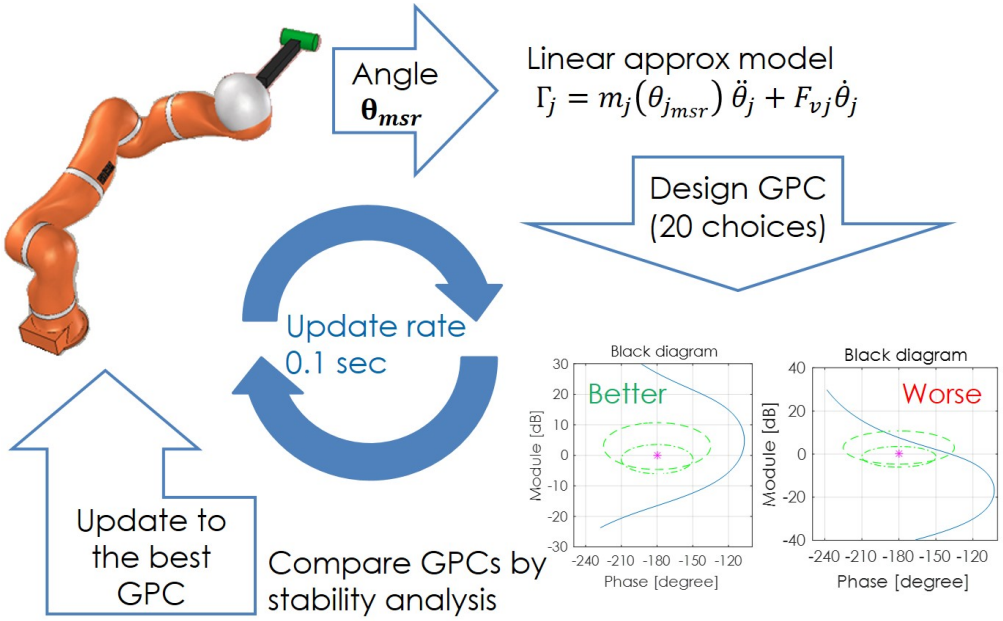

Figure 3.11: Concept of AGPC

3.5 Experimental Setup

3.5.1 Parameter Identification of the Robot and Tools

The position and force controllers with aforementioned controllers were compared using V-REP simulation platforms with KUKA LWR real experimental identification data [1]. For the identification, the KUKA LWR excuted OET and thus, robot and tool's dynamic/geometric parameters obtained following the procedure described in section 3.3. Fig. 3.12 presents the KUKA environment when the exciting motion was executed for the identification of the parameters. The identification and control were validated using two different types of painting rollers. Table 3.4 presents the physical parameters of these tools, and Table 3.5 presents the identified parameters. As Table 3.5 presents ,it can be considered the two identified masses are closed to their reference one. The parameters were downsized using PCA and stored in a look-up table with associated geometric parameters. Thus, the robot could recognize the painting roller attached to the end-effector. These values were obtained in the previous experimental identification developed by the co-authors of this paper [1]. The root mean square error (RMSE) between measured torque and estimated torque with identified dynamic parameter was 0.63 Nm [1].

Table 3.4: Physical parameters for the painting roller tools

Name of the roller	*Length* [m]	*Width* [m]	*Weight* [kg]
Big roller	0.35	0.20	0.2
Small roller	0.33	0.10	0.1

The KUKA LWR used in this study had a mass of 12 kg, and the masses of the rollers were 100 $\sim$ 200 g. Although the differences between the masses of the rollers were relatively small, and a difference of 100 g could affect the control system if it attempts to conduct precise and high-speed tasks. It should be noted that the system successfully identified the physical parameters of the rollers and differentiate between the different rollers, although the mass difference between the rollers was only 0.1 kg. Then, the identified values were used to implement the model predictive control such as the GPC and AGPC, and the computation of the gravitational compensation torque for the PD + Grv control. The masses and dimensions of the rollers were linked in the look-up

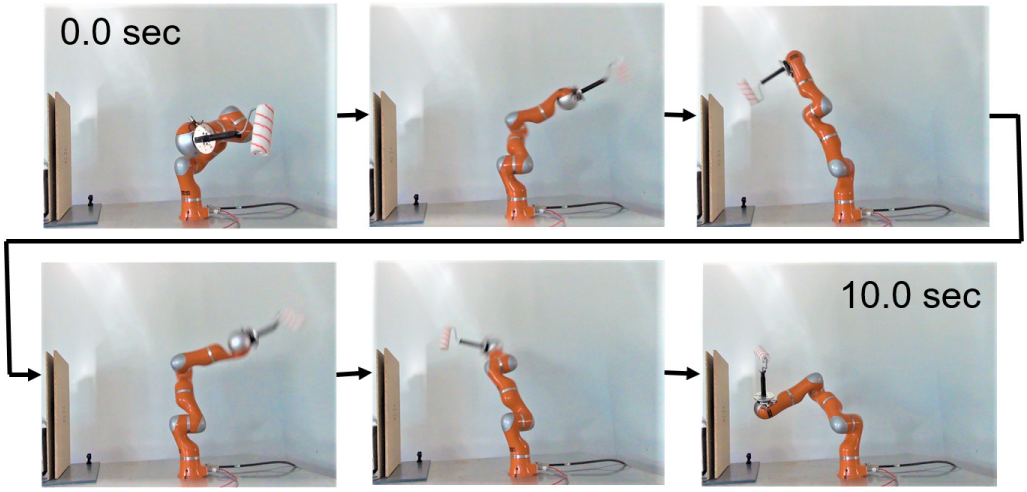

Figure 3.12: Robot motion for dynamic identification with optimal exciting trajectories [1]

Table 3.5: Identified inertial parameters for big $\mathbf{\Phi}_{br}$ and small $\mathbf{\Phi}_{sr}$ rollers [1] (see section 3.3 for the definition of each symbol). $\%\sigma_{\Phi_r}$ is relative standard deviation.

Parameter	$\mathbf{\Phi}_{br}$	$\%\sigma_{\Phi_{br}}$	$\mathbf{\Phi}_{sr}$	$\%\sigma_{\Phi_{sr}}$
M [kg]	$2.4.10^{-1}$	0.5	$9.7x10^{-2}$	1.3
MSX [kgm]	$-3.9.10^{-3}$	8.1	$4.5.10^{-5}$	28.4
MSY [kgm]	$-1.5.10^{-5}$	21.0	$7.2.10^{-5}$	25.4
MSZ [kgm]	$6.5.10^{-2}$	0.6	$2.4.10^{-2}$	1.4
XX [kgm^2]	$2.0.10^{-2}$	1.9	$7.4.10^{-3}$	4.4
YY [kgm^2]	$2.1.10^{-2}$	2.0	$7.1.10^{-3}$	5.1
ZZ [kgm^2]	$2.1.10^{-3}$	14.9	$1.5.10^{-4}$	31.0
XY [kgm^2]	$5.6.10^{-4}$	56.6	$2.0.10^{-4}$	102.8
XZ [kgm^2]	$6.3.10^{-4}$	41.5	$-4.2.10^{-4}$	54.7
YZ [kgm^2]	$8.3.10^{-5}$	49.9	$2.1.10^{-4}$	47.5

table Fig. (3.4) and updated for the final robot models.

3.5.2 Control Experiments

Following tasks are executed to compare the performance of the proposed AGPC and other controllers explained in section 3.4.3.

1. Swinging motion of the end-effector for 3 seconds using AGPC and GPC without adaptation.
2. Painting task with the big roller and small roller.

About the swinging control experiment, the control performance of AGPC and GPC in case of fast motion is compared for several trajectories. Each controller was running with rate of 1.0 kHz with V-Rep simulation environment. The V-Rep is connected to MATLAB through SDK and provide robot's state such as joint current joint information and reaction of robot to the input torque with 1.0 kHz. Note that AGPC adaptation process rate is 10 Hz. The duration is 3.0 seconds and the velocity and acceleration at beginning and ending point of each motion is set as zero and the trajectory is smoothly interpolated as Fig. 3.13 and Fig. 3.14 shows an example.

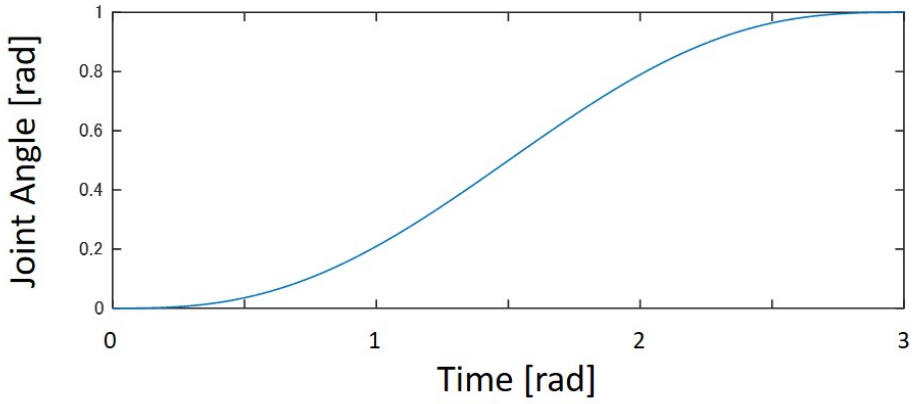

Figure 3.13: Example of the joint space trajectory for swinging motion

The second control experiment is to compare proposed AGPC and other controllers by means of force control and position control accuracy. The task is drawing W shape on the wall with applying force of 30 N as showin in Fig 3.15 and Fig 3.16. Each controller was running with rate of 1.0 kHz with V-Rep simulation environment. The V-Rep is connected to MATLAB through SDK and provide robot's state such as joint current joint information and reaction of robot to the input torque with 1.0 kHz.

3.6 Results and Discussion

3.6.1 Swinging Motion with AGPC and GPC Without Adaptation

To ensure the proper function of the AGPC, simple motion control was conducted using the AGPC in the V-REP environment. Fig. 3.17 - Fig. 3.22 is showing 3D trajectory of the end-effector trajectory where the duration is 3 seconds, and its joint space trajectory. The reference trajectories are plotted in dotted black line, and AGPC and GPC is plotted in red and blue solid line respectively.

These figures reveal that the AGPC could stably control the LWR and follow the reference with minimal error. Because the range of motion is large for the robot, as shown here, the controller

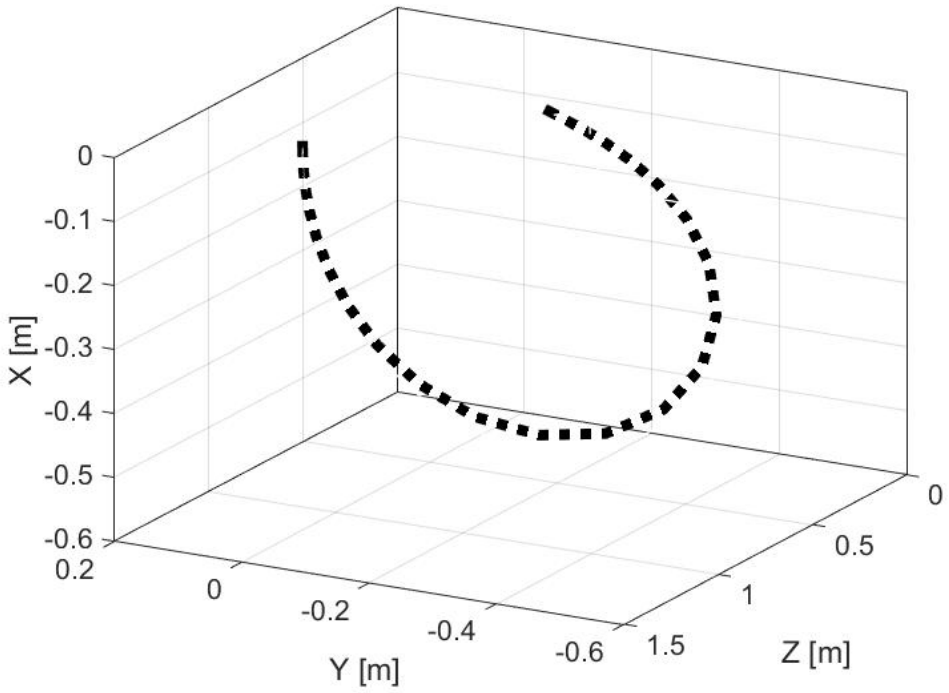

Figure 3.14: Example of the Cartesian space trajectory for swinging motion

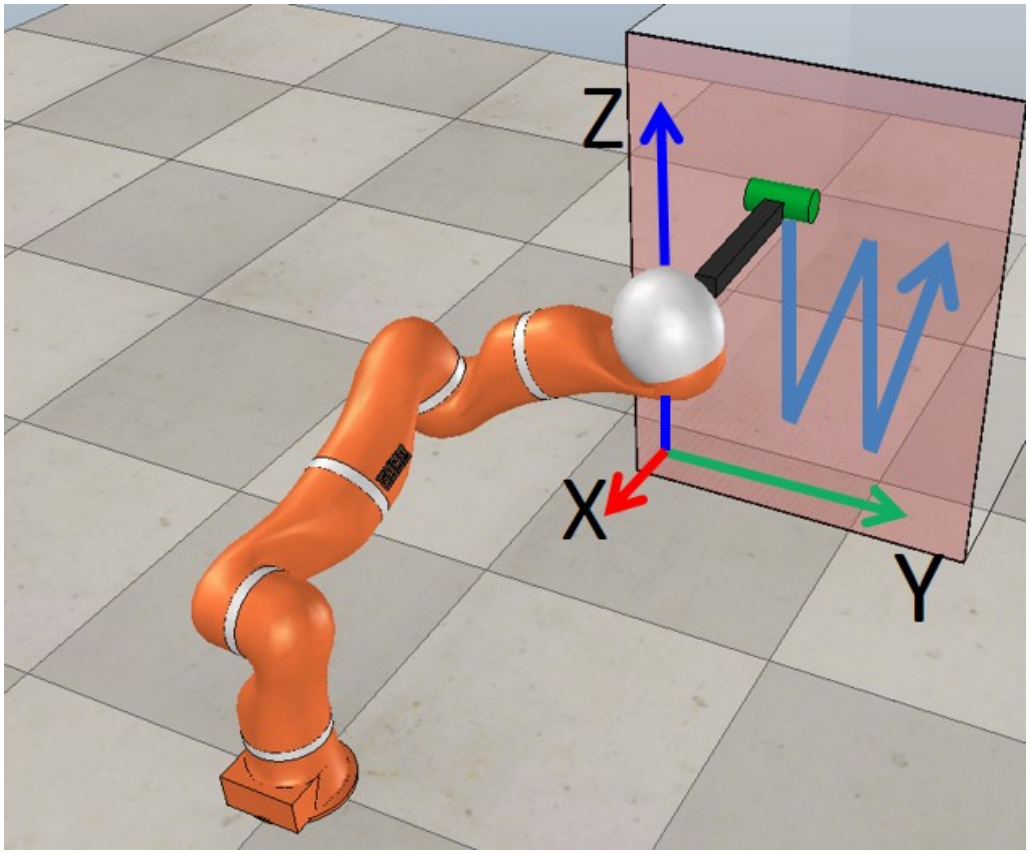

Figure 3.15: Painting task simulation in V-Rep simulator

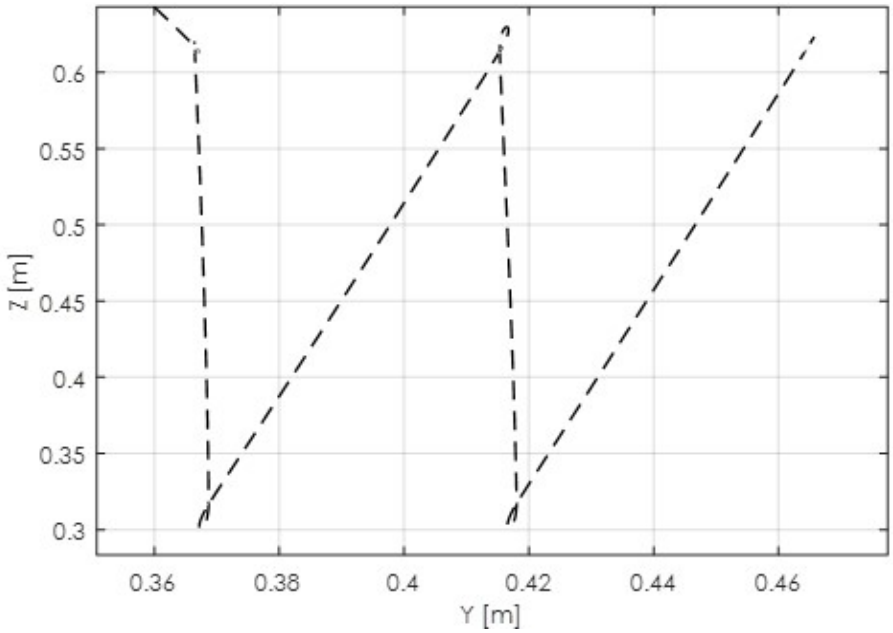

Figure 3.16: Reference task for the painting task

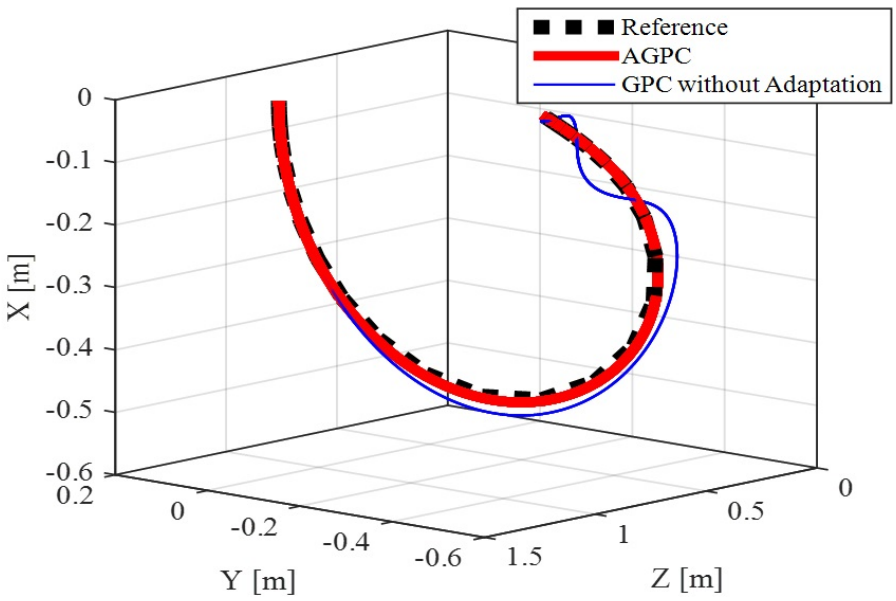

Figure 3.17: 3D trajectory of the end-effector with AGPC controller with simple trajectory #1

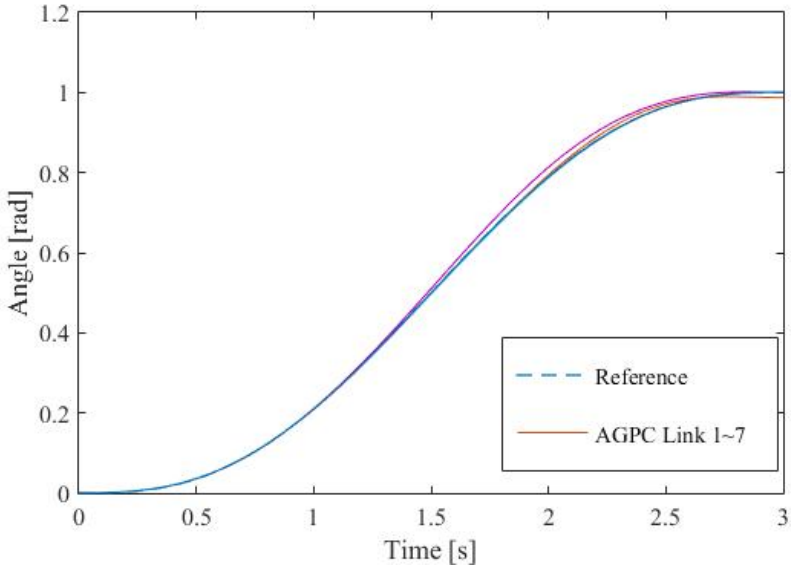

Figure 3.18: Joint space trajectory result with AGPC controller with simple trajectory #1

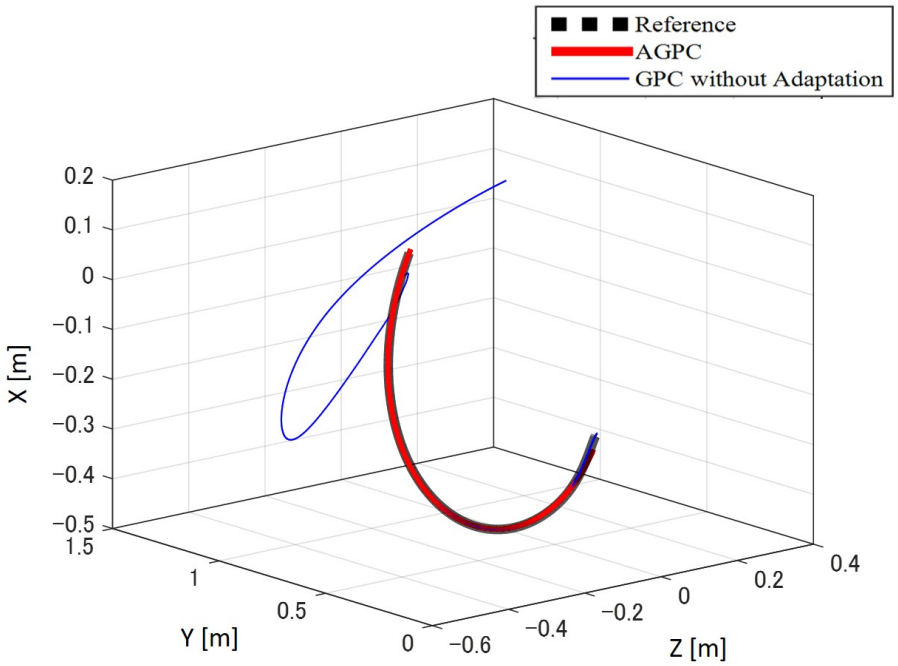

Figure 3.19: 3D trajectory of the end-effector with AGPC controller with simple trajectory #2

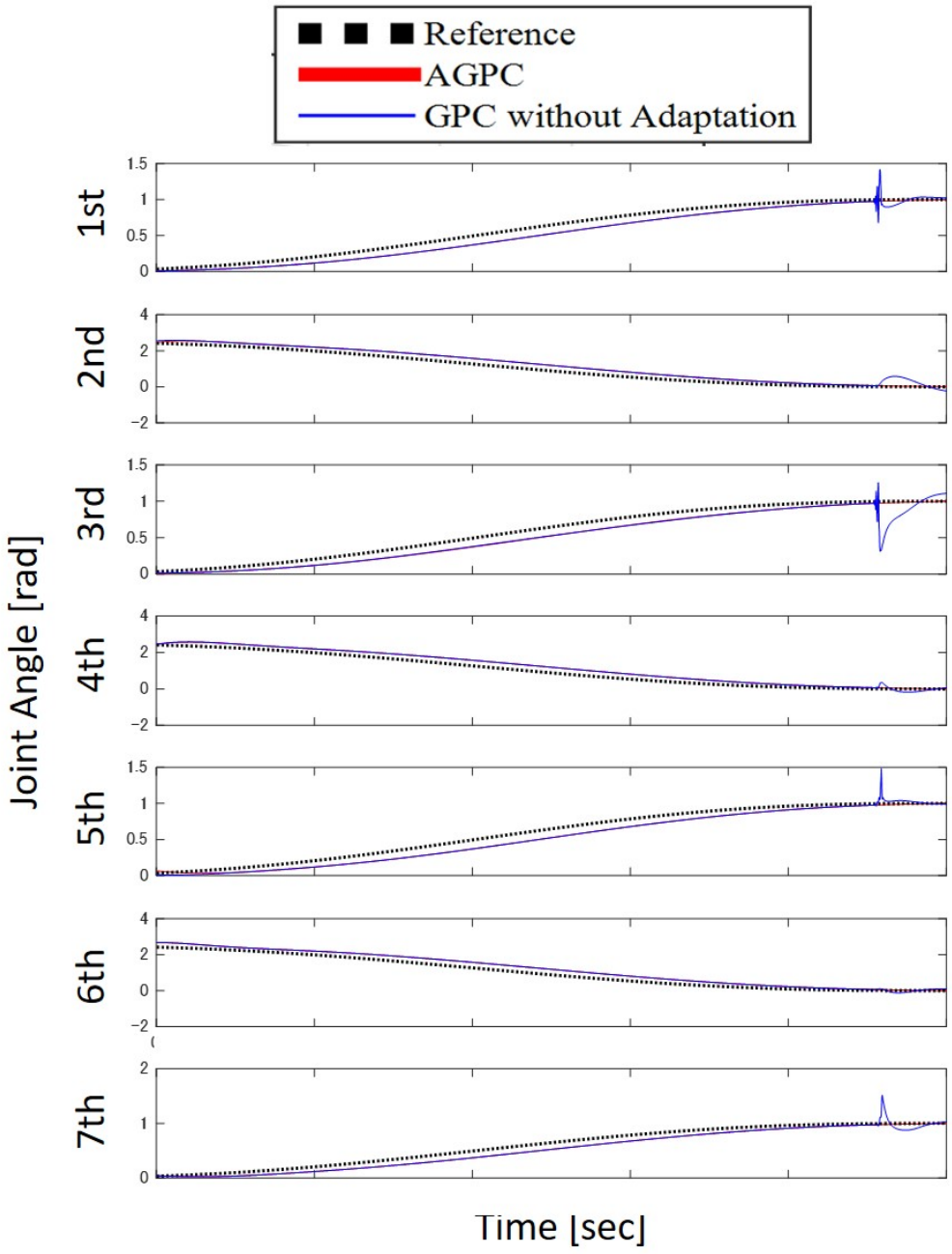

Figure 3.20: Joint space trajectory result with AGPC controller with simple trajectory #2

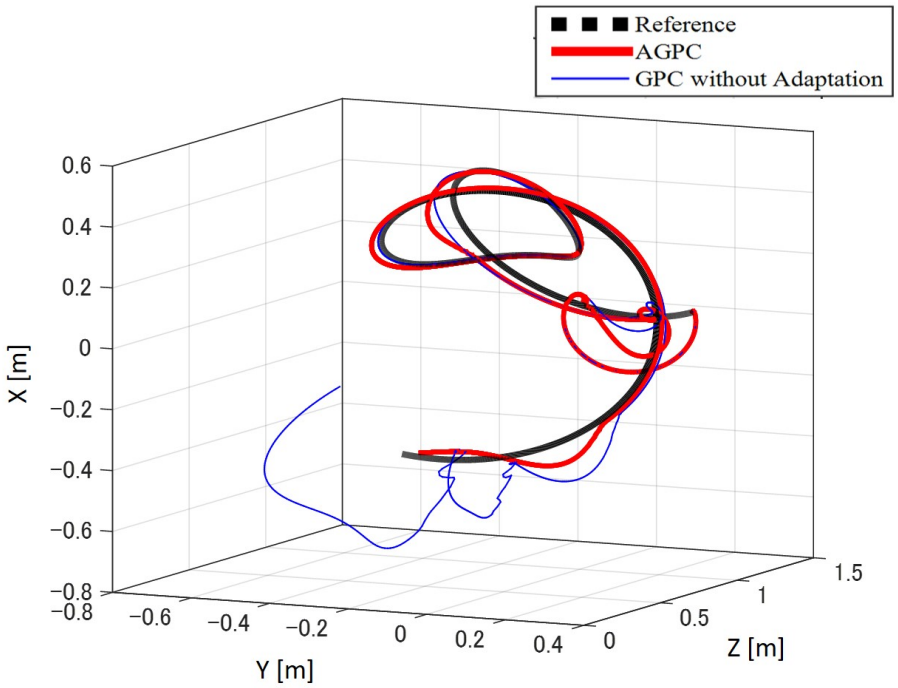

Figure 3.21: 3D trajectory of the end-effector with AGPC controller with complex trajectory

is expected to operate stably over a wide range of motion. Moreover, Fig. 3.23 presents an time series change of the polynomial $R_{adp}(z^{-1})$ for the case of trajectoy #1 (Fig. 3.17, Fig. 3.18), and how it varied in accordance with changes in the posture of the robot.

3.6.2 Painting Task Results

Fig. 3.24 - Fig. 3.27 show that the GPC and AGPC controllers yielded accurate tracking, whereas the PD control + gravitational compensation and feedforward + PD control deviated from the desired trajectory, as the painting motion progressed. There was approximately 5 mm of a constant position error in the tracking control with the controllers with PD in the both cases of the big roller and small roller. The real-time duration of each trajectory was 50 s and 46 s for the large roller and small roller, respectively.

Given that the PD and feedforward control do not support integral action, the robot could not handle the position error caused by the external force at the end-effector. On the other hand, the GPC supports integral action due to the nature of its structure, successfully managed the disturbance of external force and exhibited a significant improvement of tracking ability.

Although the system demonstrated a sufficient force control and tracking control performance, a constant position error of approximately 5 mm was presented in each case, as mentioned above. The error may be suitable for the painting task over a large area. However, the system would be required to reduce the position error if the complexity of the task and precision requirements increased such as writing small letters or locate the tool into the reduced place. Therefore, other controllers with the adaptive methods for position error were proposed, i.e., the GPC, AGPC, and feedforward - PD control using the optimal trajectory and force control strategy discussed in the previous sections.

Furthermore, the results reveal that the GPC with feedback linearization demonstrates a good performance due to the fine identified inertial parameters used to develop the model of the robot. Fig. 3.28 presents the RMS position error when the controllers were implemented. The GPC

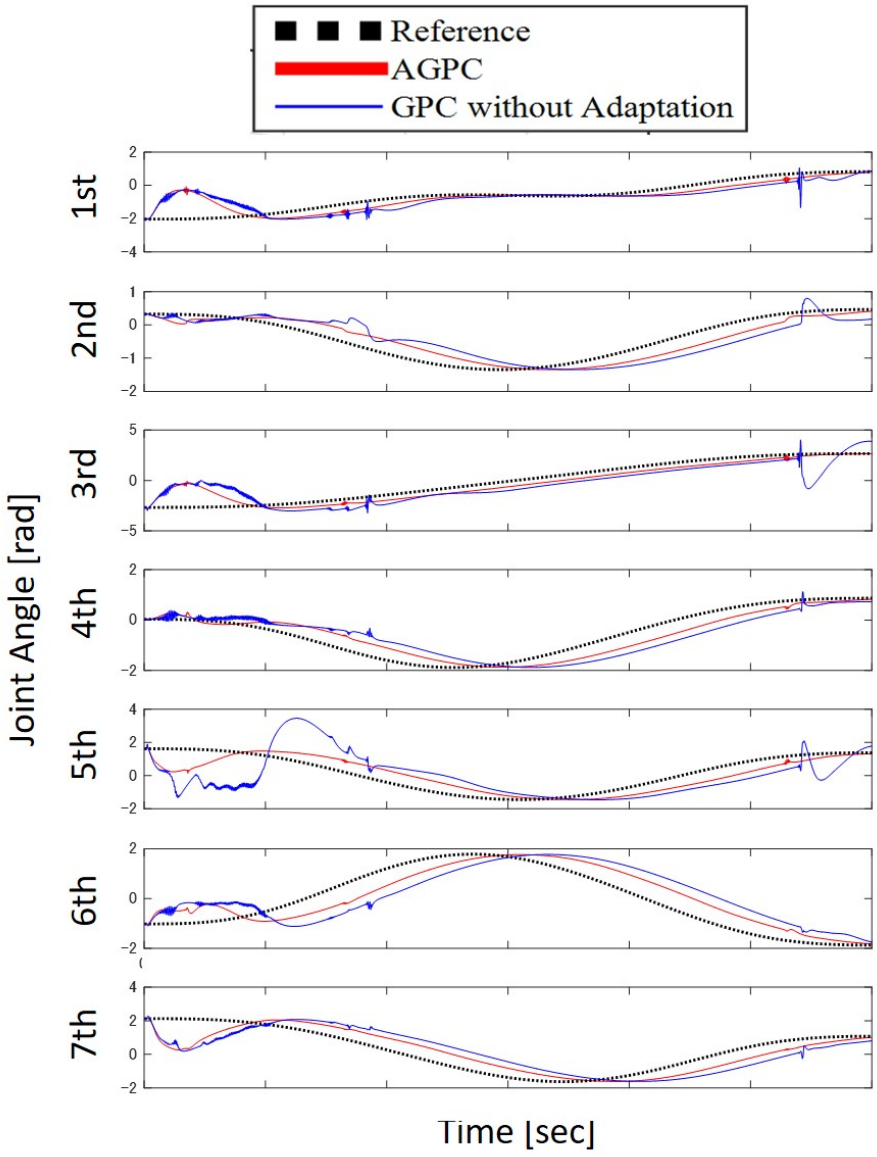

Figure 3.22: Joint space trajectory result with AGPC controller with complex trajectory

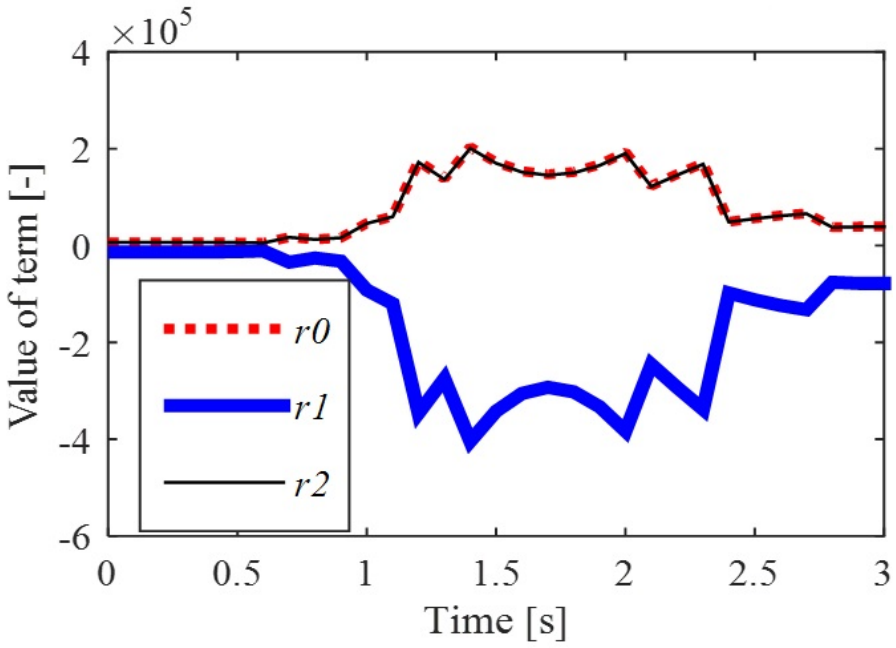

Figure 3.23: Transition of $R(z^{-1})$ polynomial terms r_0, r_1, and r_2 with respect to the new optimal λ. It represents the controller is changing/adapting its internal value to cope with non-linearity of the robot arm dynamics.

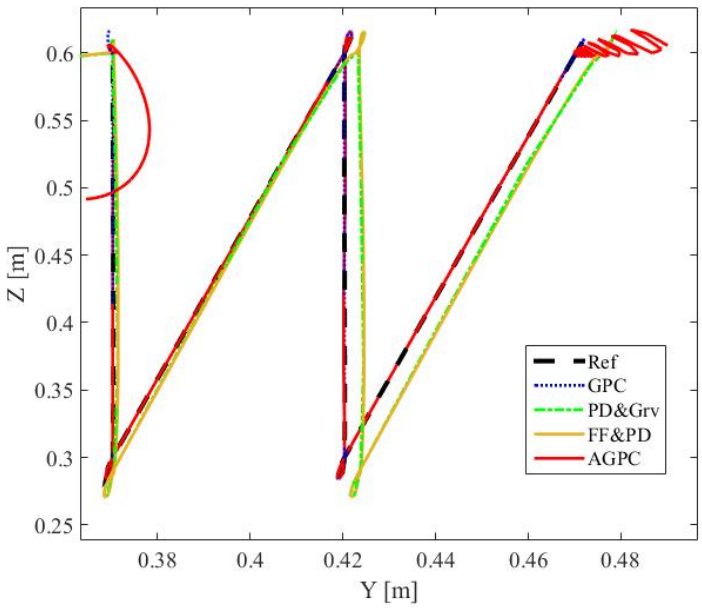

Figure 3.24: Results of painting task with GPC, AGPC, and PD+ for small roller

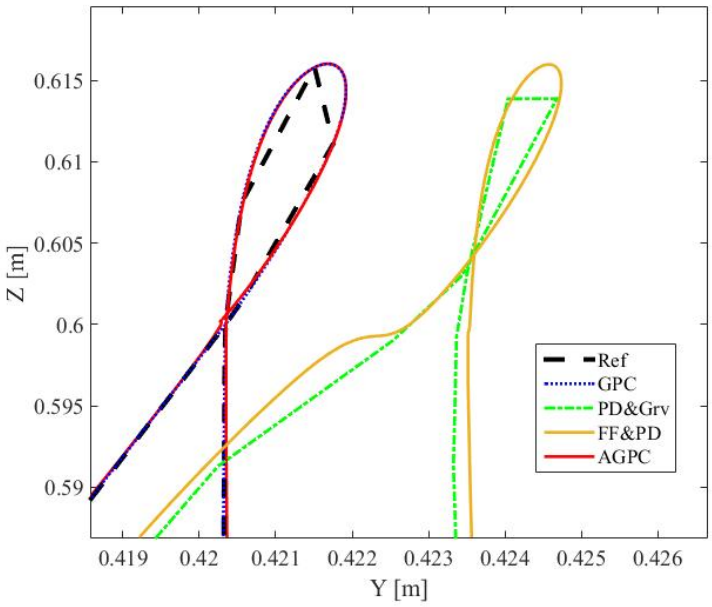

Figure 3.25: Zoom results of painting task for small roller with GPC, AGPC, and PD+

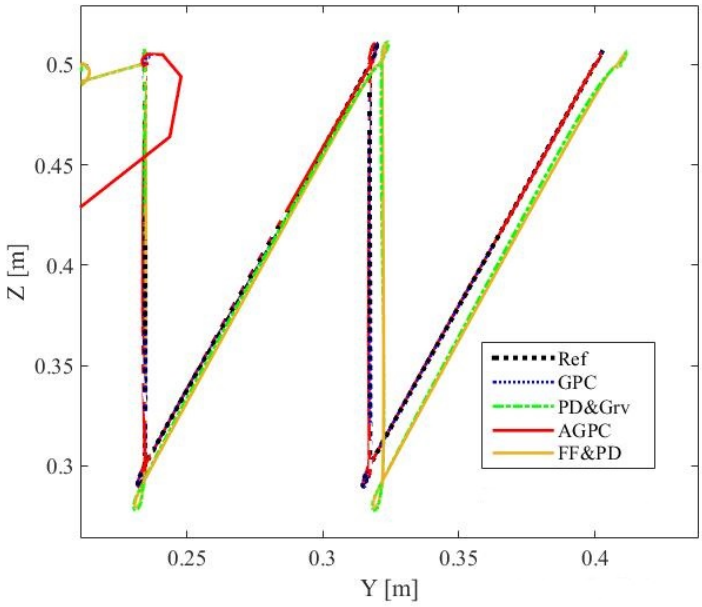

Figure 3.26: Results of painting task with GPC, AGPC, and PD+ for big roller

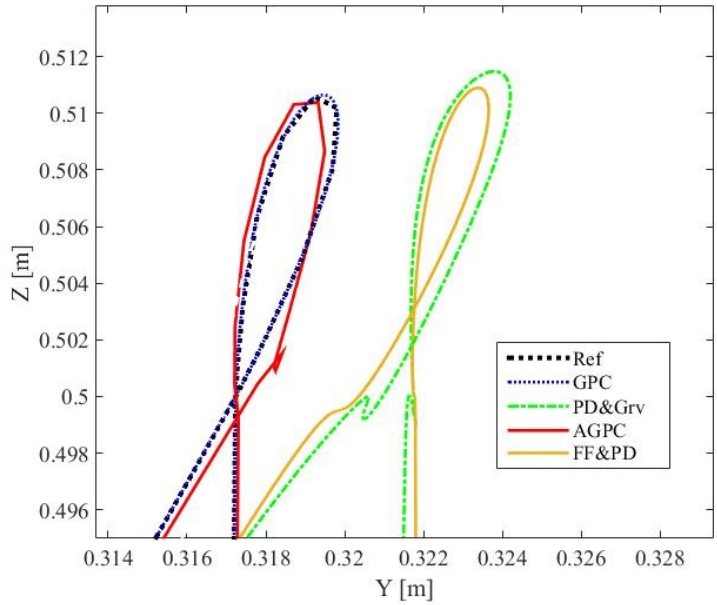

Figure 3.27: Zoom results of painting task for big roller with GPC, AGPC, and PD+

and AGPC yielded a minimal position error when compared with that of PD with gravitational compensation. The results for the GPC and AGPC were 0.1 mm and 0.21 mm with the small roller, and 0.22 mm and 0.62 mm with the big roller in the end-effector, respectively.

As shown in Fig. 3.29 and Fig. 3.30, the GPC and AGPC yielded better force control results due to their fast reaction capabilities. The AGPC presented a better control result than the PD control with gravitational compensation. Moreover, the AGPC re-adjusted according to the posture of the robot arm on-line. In addition, in the case of the GPC with feedback linearization, the AGPC can work without to direct affect of the acceleration feedback, which could contain a noise in the real situation. Thus, the AGPC is more suitable for robots in planning and motion tasks.

3.6.3 Considerations of the Controllers

The GPC seems have the best performance in this study and hare is the discussion about it. The GPC implemented in this study utilize the feedback linearization, and was optimized by tuning the parameter λ, prediction windows N_1-N_2, and control horizon N_u, as discussed in Section c). By adjusting the parameters, GPC response speed, and convergence speed/position, the stability and robustness sensibilities changed. Given that the controlled system is strictly linearized, and that its controlled MIMO system is decoupled and does not change over the range of motion, a fixed GPC (fixed values of design parameters) is applicable to this system, and the values of the optimal parameters depend on the motion task.

There is more possibility in robot arm with GPC. By combining GPC + feedback linearization and on-line system identification, another AGPC can be introduced. Using on-line system identification, the system could detect whether the robot is in high-speed motion or interacting with the external environment. In this case, the high response speed of the GPC should be designed by adjusting the design parameters. Moreover, it is possible that using on-line system identification to design the feedback linearizer on-line strictly linearizes the robot system. This system identifies the dynamic parameters of the robot on-line and simultaneously constructs a feedback linearizer using the identified values. Eventually, the optimal linearizer is obtained by evaluating the tracking error. However, it is difficult to set the initial parameters for the linearizer and the GPC, which

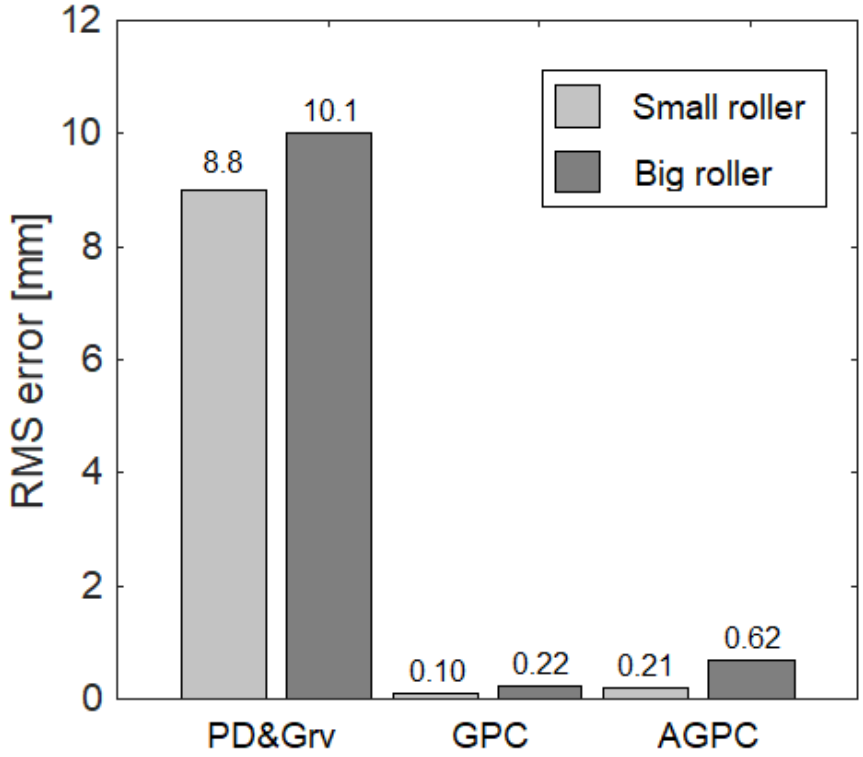

Figure 3.28: RMS position error with GPC, AGPC, and PD+ controllers

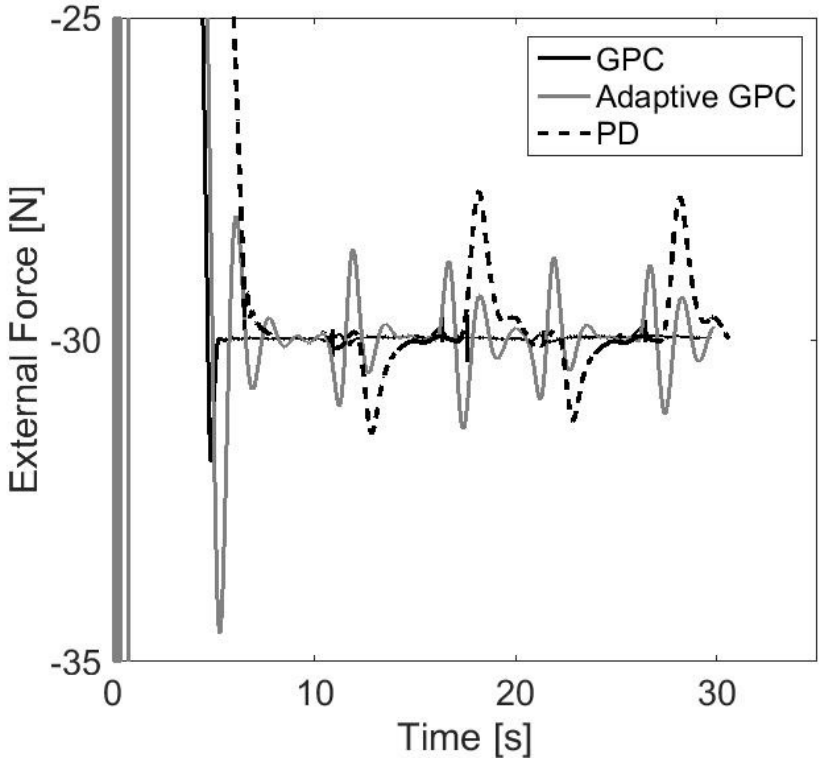

Figure 3.29: External Force with GPC, AGPC, and PD+ controllers

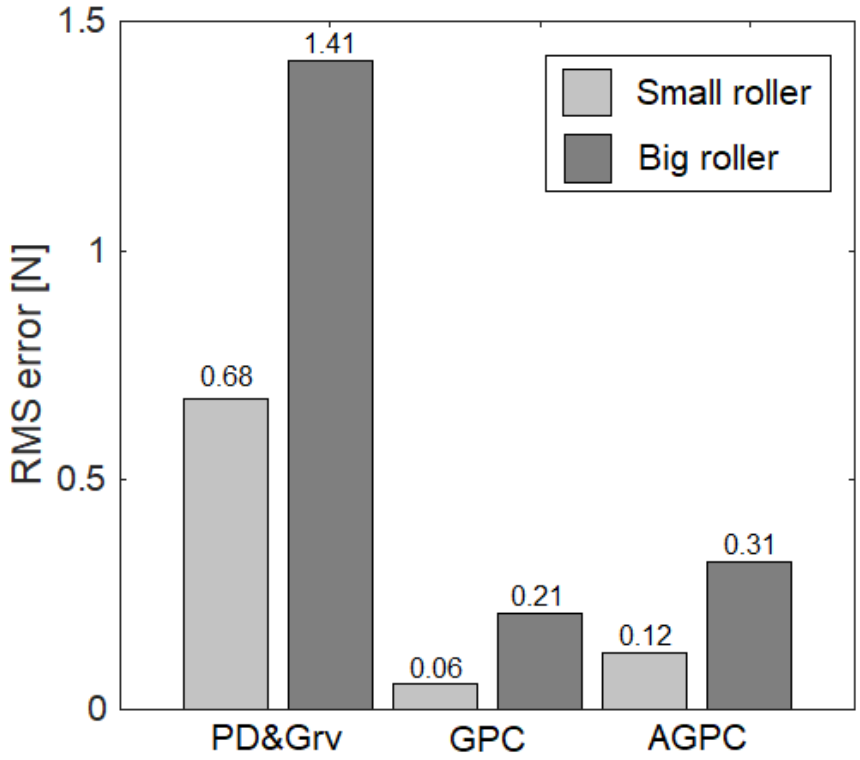

Figure 3.30: RMS force error with GPC, AGPC, and PD+ controllers

provide a certain level of stability from the start of the motion.

The computational cost was calculated by the program for different types of rollers, to validate the GPC, AGPC, and PD+ controllers. The computational cost was verified by measuring the allocated memory, % of CPU, and time consumption. The results are summarized in Table 3.6.

Table 3.6: Computational cost using the GPC, AGPC, and PD+ during the painting task

Controller	$Memory[GB]$	$\%CPU$	$CPUtime[s]$
AGPC (small roller)	0.993	27	0.043
AGPC (big roller)	0.628	21	0.051
GPC (small roller)	0.950	23	0.032
GPC (big roller)	0.599	21	0.040
PD+ (small roller)	0.628	18	0.037
PD+ (big roller)	0.601	17	0.038

The table above presents CPU time of each controller for one cycle. As expected, it shows that the AGPC has the highest computational cost. This is because the AGPC conducts optimizations and stability analyses, in addition to the calculation of the control torques. In this study, the sampling time for the calculation was 1 ms; approximately 43 seconds was required to calculate 1 second of motion. However, this type of on-line adaptive controller can be further enhanced with potential computational characteristics and optimize scripts to decrease the duration of the control tasks.

3.7 Conclusion

In this study, a novel robot control adaptation based on predictive controllers and the fast dynamic and geometric identification of base parameters in a force control framework was developed.

The proposed methods were applied to the KUKA lightweight robot. During the roller painting task, the external forces applied to the end-effector could be estimated and controlled without a force sensor mounted at the end-effector, due to the accurate model identification.

Moreover, the controllers were developed using the identification performed with the exciting motion of 10 s, which was generated by an optimization. The proposed adaptive predictive controllers were compared with other controllers such as the PD control with gravitational compensation, feedforward with PD control, and GPC. The AGPC updated itself during the motion and successfully controlled the robot with good stability by exploiting the fine identified SIPs. The GPC and AGPC demonstrated excellent control performances such as a 90% reduction in the position errors when compared with the results of the previous experiment. Moreover, a novel tuning design with robust control will be developed in the future to enhance the AGPC. Furthermore, the future work will extend this method to a significantly larger number variables, rollers, and tools.

Although GPC + feedback linearization demonstrated an improved tracking ability and lower computation cost, AGP exhibited a superior control performance. Moreover, it confirmed the potential of the GPC adaptation in the application with the on-line robotic arm during the task. The performance of the proposed AGPC therefore verified the effectiveness of the GPC is solving the robotic arm control problem using a simple model such as in (22), and demonstrated the robustness and applicability of the GPC scheme.

3.8 Publications

The following list contains the published and submitted papers regarding this doctoral work.

Journals

- S. Hagane, L.K. Rincon-Ardila, T. Katsumata, V. Bonnet, P. Fraisse, G. Venture, Adaptive Generalized Predictive Controller and Cartesian Force Control for Robot Arm Using Dynamics and Geometric Identification, *Journal of Robotics and Mechatronics*, volume 30(6), pp.927-942, December 2018, ISSN: 1883-8049, DOI: 10.20965/jrm.2018.p0927

Conference

- T. Katsumata, S. Hagane, B. Navaro, P. Fraisse, G. Venture, Adaptive cartesian force control using payload dynamics and geometric identification, The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec2017), Fukushima Japan, Session 1P1-H11, November 2017, ISSN: 2424-3124, DOI: 10.1299/jsmermd.2017.1P1-H11
- S. Hagane, L. Rincon, T. Katsumata, G. Venture, Adaptive Generalized Predictive Control for 7 DoF Robot Arm, The Proceedings of 23th Robotics Symposia, Shizuoka Japan, Session 5C1, pp 324-329, March 2018

Chapter 4

Linearizing Robotic Manipulator's Dynamics Using Koopman Operator and Applying Generalized Predictive Control

SUMMARY the second study is about generalized predictive control, but here, the main focus is on linearizing the dynamical system of a robot using the concept of the Koopman operator. Using this concept of the Koopman operator, a nonlinear dynamical system of a robot can be represented by a linear time-invariant state-space model. This allows us to apply advanced linear control laws, including generalized predictive control, to the robot arm. Furthermore, since the computational cost is lower than that of predictive control based on rigorous modeling, high control frequency can be achieved in a real environment. Here, a 7-DOF robot is simplified to a 3-DOF robot, and the Koopman operator is identified using a method inspired by the robot arm dynamics model. A state-space model was extracted from the identified Koopman operator, and its accuracy was compared with that of the state-space model using Koopman operators in previous studies. We also designed a generalized predictive controller for the obtained linear state-space model of the robot and compared its control performance with that of a conventional PD controller. The results show that the GPC with Koopman operators maintains constant control performance over a wide range of postures, speeds, and work ranges compared to the PD controller with fixed gains.

4.1 Introduction

In recent years, with the ability to handle large-scale time series data, there has been an attempt to mathematically extract the essence of complex real-world phenomena from observed values in order to advance understanding or control them. However, often these real-world phenomena have non-linear characteristics and controllers for non-linear dynamic systems often require complex modeling and a rough approximation of the system. On the other hand, linear control theory is well established more than non-linear control, and techniques to linearize non-linear dynamics have been researched so that linear control theory can be applied to non-linear systems. Attempts have been made to consider the system locally linear by approximating it in the neighborhood of the reference point by Taylor Expansion, or globally linearize by state feedback linearization using state estimation observers [22]. However, the former can only guarantee linear control performance locally, and the latter is significantly affected by model errors, making it unsuitable for non-linear controllers such as model-based control.

Here, the concept called "Koopman Operator" (KP) is particularly interesting [57, 58]. The KP is a linear operator defined for nonlinear dynamical systems and represents the time evolution of the observed quantity (output function in control engineering) of the dynamical system. This operator

is a linear operator that preserves the information of the original nonlinear dynamical system and it allows mapping dynamic constraints into a linear dynamical system in a higher dimensional state space. Its basic idea is to lift non-linear function to a higher dimensional space by extending its explanatory variables that are often referred as "observables". In this higher dimension, nonlinear functions can be described in a linear form. Here, how the observables should be augmented depends on the dynamical system's property and needs to be chosen with consideration, such as which kind of observables and how many should be[59].

Linear dynamic models derived from KP enable the application of linear control methods and hold great promise in the field of control engineering. In particular, the adaptability of KP with Model Predictive Control (MPC) [62], is pointed out by Korda [63] and Igarashi et al. [64]. The motivation to apply MPC exists especially in the field of robotics. This is because predictive control can accommodate sequential changes in future target trajectories and can be used for intelligent motion control, where the robot generates and executes unique motions at a moment's notice. However, robot dynamics are nonlinear, and MPCs that include nonlinear models suffer from high computational costs and difficulties in generalizing the design [63]. On the other hand, if the dynamic system of the robot could be represented by a linear equation of state, the computational cost would be small enough for practical systems, and it would be possible to apply not only the well-established linear control theory such as Linear Quadratic Regulator (LQR), but also Generalized Predictive Control (GPC).

Other well establish control method for robot arm is computed torque contol (CTC) method [23]. It utilize robot's dynamic and geometric information to conduct feedback linearization and reduce the control system to linear and decoupled. However, applicable system for this method is limited to the legit body structure and require joint acceleration information which likely to contain some noise. Moreover, dynamic parameter identification require joint torque information which is not always available. On the other hand, this data driven linearization by KP enable any structure to acquire linearized model without dynamic parameter identification. In the future, the structure of robot will be more diverse and acquiring traditional form of dynamic model will be not simple. For example, soft robot that doesn't have joints and bio-inspired joint with complex actuator is not easy for CTC method to adapt contrary to the presented method.

This chapter presents an example of the application of the KP for a robot manipulator which has a nonlinear dynamics, and introduces practical issues to be considered when applying KP to robotic systems. Also, it presents working candidates for new state vectors for manipulators. In this study, the optimal expansion scheme was acquired by expanding the robot dynamics equation and using its monorails. The KP was then identified by the least-squares minimization, and an approximate linear state-space model of the robot was identified. Using this, GPC was designed and its tracking performances for a wide range of motions are compared with traditional PD control. Also, the linear model from observables choice of previous research and ours are compared.

4.2 Linear Predictor

First, the concept of linear predictor for non-linear dynamics is explained here. A discrete-time nonlinear controlled dynamical system is

$$x_{t+1} = f(x_t, u_t) \tag{4.1}$$

where $x \in \mathbb{R}^n$ is the state of the system, $u \in \mathbb{R}^m$ is the control input, subscripts t indicates time steps and f is the mapping.

We need the predictor that takes initial condition x_0 and the control inputs $\{u_0, u_1, ...\}$ and outputs the trajectory of Eq.(4.1) along certain time steps. In particular, the form of state space model which are suitable for linear control design is required. The discrete state space model is

$$\begin{aligned} z_{t+1} &= \mathbf{A}z_t + \mathbf{B}u_t \\ \hat{x}_{t+1} &= \mathbf{C}z_{t+1} \end{aligned} \tag{4.2}$$

where $z_t \in \mathbb{R}^N$, and $\hat{x}$ is predicted value of x. Hare, N is generally $N \gg n$. $A \in \mathbb{R}^{N\times N}$ is the state matrix, $B \in \mathbb{R}^{N\times m}$ is the control matrix, and $C \in \mathbb{R}^{n\times N}$ is the output matrix. The initial condition of the predictor Eq.(4.2) is

$$z_0 = \Psi(x_0) := \begin{bmatrix} \psi_1(x_0) \\ \vdots \\ \psi_N(x_0) \end{bmatrix} \tag{4.3}$$

where x_0 is the initial value of states in Eq.(4.1), and $\psi_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, ...N$, are lifting functions and importantly, it is user specified functions. The state z_t is referred as lifted stets or observables.

The idea is that if the true trajectory of x_t generated by Eq.(4.1) and the predicted trajectory from Eq.(4.2) are close within a sufficient time interval, the optimal controller based on Eq.(4.2) should be close to the optimal controller for Eq.(4.1).

4.3 Koopman Operator

We start by recalling the Koopman operator (KP) approach for the analysis of an uncontrolled dynamical system.

$$x_{t+1} = f(x_t) \tag{4.4}$$

The KP is a linear operator defined for nonlinear dynamical systems and represents the time evolution of observables in the dynamical system. Let $\mathcal{F}$ denotes the space that the observables Ψ in Eq.(4.3) can take, $\mathcal{F}$ has as elements not only state variables but also functions, such as their squares. Therefore, $\mathcal{F}$ is a space of infinite dimension. Now, when $N = \infty$, the operator $\mathfrak{K} : \mathcal{F} \to \mathcal{F}$ that evolves from an observable $\Psi(x_t)$ to a new observable $\Psi(x_{t+1})$ is defined as follows.

$$(\mathfrak{K}\Psi)(x_t) := \Psi(x_{t+1}) \tag{4.5}$$

The time evolution of the observables Ψ under the mapping f in Eq.(4.4) can then be expressed as follows.

$$\begin{aligned} \Psi(x_t) = \Psi(f(x_{t-1},)) &= (\mathfrak{K}\Psi)(x_{t-1}) + r_{t-1} \\ &= (\mathfrak{K}^2\Psi)(x_{t-2}) + r_{t-1} + r_{t-2} = ... \\ &= (\mathfrak{K}^t\Psi)(x_0) + \textstyle\sum_{i=0}^{t} r_i \end{aligned} \tag{4.6}$$

This $\mathfrak{K}$ is called the Koopman operator [57, 58] and is a linear operator that represents the time evolution of an observable under a nonlinear mapping f. And $r_i, i = 0...t$ is the transition error. In principle, as the size of operator $N \times N$ goes to $\infty \times \infty$, the transition error r_t goes to zero [118, 119]. However, it is sometimes possible to find $N < \infty$ such that $\sum_{i=0}^{t} r_i \approx 0$ [120].

4.3.1 Koopman Operator for Controlled Systems

There are several ways of generalizing the KP to controlled systems [63, 121, 122].

The KP can include an input u that affects the transition of observables. Consider the observables functions that include the control input as shown below.

$$\Psi(x_t, u_t) := \begin{bmatrix} z_t \\ u_t \end{bmatrix} = \begin{bmatrix} \psi_1(x_t) \\ \vdots \\ \psi_N(x_t) \\ u_t \end{bmatrix} \tag{4.7}$$

The KP $\mathfrak{K} \in \mathbb{R}^{(N+m)\times(N+m)}$ associated to Eq.(4.7) is defined by

$$\begin{gathered} (\mathfrak{K}\Psi)(x_t, u_t) := \Psi(x_{t+1}, u_{t+1}) \\ \mathfrak{K} = \begin{bmatrix} \mathfrak{K}_x & \mathfrak{K}_u \\ \cdot & \cdot \end{bmatrix} \end{gathered} \tag{4.8}$$

Here, $\mathfrak{K}_x \in \mathbb{R}^{N\times N}$ is the part that represents the impact of the previous lifted state on the next states, and $\mathfrak{K}_u \in \mathbb{R}^{N\times m}$ represents the contribution of input to the evolution of the lifted state.

Finally, those submatrices of the KP $\mathfrak{K}_x, \mathfrak{K}_u$ are used to obtain the linear predictor in Eq.(4.2).

$$\begin{gathered} \Psi(x_{t+1}, u_{t+1}) = \mathfrak{K}\Psi(x_t, u_t) \\ \Downarrow \\ z_{t+1} = \mathfrak{K}_x z_t + \mathfrak{K}_u u_t \\ \hat{x}_{t+1} = \mathbf{C} z_{t+1} \end{gathered} \tag{4.9}$$

Note that finding the output matrix $\mathbf{C}$ is simple when the observables explicitly include the state variable. In that case, the output matrix will be an index matrix. And from now on, we will proceed with the discussion assuming that the observables explicitly include the state variable that is the object of control.

Also note that, it is generally not possible that the trajectory of this linear predictor Eq.(4.9) matches the trajectory of the original nonlinear system Eq.(4.1) for all times. Nevertheless, these predictors can be used for MPC that use finite-time predictions, as long as the predictions are accurate for a sufficient time interval.

4.3.2 Approximation of Koopman Operator

When enough data $D = \{x_i, u_i\}_{i=0}^{M}$ and $M > N$ is provided, we can approximate KP $\mathfrak{K}$ using least-squares minimization over the parameters of $\mathfrak{K}$ [61, 123].

$$\min_{\mathfrak{K}} \frac{1}{2} \sum_{i=0}^{M-1} \|\Psi(x_{t+1}) - \mathfrak{K}\Psi(x_t)\|^2 \tag{4.10}$$

Given that Eq.(4.10) is convex in $\mathfrak{K}$, the solution is given below.

$$\mathfrak{K} = \mathbf{G_1} \mathbf{G_2}^{+} \tag{4.11}$$

Where

$$\begin{aligned}\mathbf{G_1} &= \frac{1}{M}\sum_{i=0}^{M-1}\Psi(x_{i+1})\Psi(x_i)^T \\ \mathbf{G_2} &= \frac{1}{M}\sum_{i=0}^{M-1}\Psi(x_i)\Psi(x_i)^T\end{aligned} \tag{4.12}$$

In here, + denotes the Moore-Penrose pseudoinverse. The continuous time operator $\mathfrak{L}$ is then given by

$$\mathfrak{L} = \frac{1}{T_s} log(\mathfrak{K}) \tag{4.13}$$

Where T_s is the time interval for each time step.

4.4 Generalized Predictive Control

Generalized Predictive Control (GPC) is a method of model predictive control (MPC) proposed by Larke et al. in 1987 [111]. And it is compose of

1. Derivation of output prediction equation using model of control target.
2. Determination of the optimal control rule based on the prediction equation.
3. Application of the Receding Horizon Method

Where, the Receding Horizon Method means that the time interval in which the range of output prediction and the optimal control law are considered is successively shifted forward in time. Specifically, only the first few samples (usually one sample) of the optimal control inputs are applied to the control target at each time step.

Various methods have been proposed for model predictive control, while GPC is a method that uses the CARIMAX model (Controlled Auto-Regressive Integrated Moving Average model) or state-space model for output prediction and derivation of the optimal control input. GPC is a systematic arrangement and generalization of MPC within the framework of control theory [124].

Here, we consider an m-input n-output linear time-invariant plant with a state-space model in Eq.(4.2). In this case, the control objective is to make the system output follow the step-like target value stably. Therefore, in order to include an integrator in the control system and to follow the target value without steady-state deviation, an expanded system that includes an integrator in the plant of Eq.(4.2) is constructed.

$$\begin{aligned}z_{e_{t+1}} &= \mathbf{A}_e z_{e_t} + \mathbf{B}_e \Delta u_t \\ \hat{x}_t &= \mathbf{C}_e z_{e_t}\end{aligned} \tag{4.14}$$

where

$$\begin{aligned}\mathbf{A}_e &:= \begin{bmatrix}\mathbf{A} & \mathbf{B} \\ \mathbf{0}_{m\times n} & \mathbf{I}_m\end{bmatrix},\ \mathbf{B}_e := \begin{bmatrix}\mathbf{B} \\ \mathbf{I}_m\end{bmatrix} \\ \mathbf{C}_e &:= \begin{bmatrix}\mathbf{C} & \mathbf{0}_{n\times m}\end{bmatrix},\ z_{e_t} := \begin{bmatrix}z_t \\ u_t\end{bmatrix} \\ \Delta &= 1 - \xi^{-1}\end{aligned} \tag{4.15}$$

Note that ξ^{-1} is an operator representing a one-sample delay. The output prediction equation for j step ahead can be derived by repeating Eq.(4.14) as follows.

$$\hat{x}_{t+j} = \mathbf{C}_e \mathbf{A}_e^j z_{e_t} + \sum_{i=0}^{j-1} \mathbf{C}_e \mathbf{A}_e^{j-i-1} \mathbf{B}_e \Delta u_{t+i} \tag{4.16}$$

In GPC, the following linear-quadratic form of the cost function is defined to derive the optimal control input.

$$J = (\hat{X}_t - X_{r_t})^T \mathbf{Q} (\hat{X}_t - X_{r_t}) + \Delta U_t^T \mathbf{R} \Delta U_t \tag{4.17}$$

where

$$\begin{array}{c} \hat{X}_t^T := [\hat{x}_{t+N_1}^T, ..., \hat{x}_{t+N_2}^T] \\ \Delta U_t^T := [\Delta u_t^T, ..., \Delta u_{t+N_u-1}^T] \\ X_{r_t}^T := [x_{r_{t+N_1}}^T, ..., x_{r_{t+N_2}}^T] \end{array} \tag{4.18}$$

$\mathbf{Q}$ and $\mathbf{R}$ are the weight matrices for the prediction tracking error and the input increment, respectively, both semi-positive definite matrices. N_1, N_2, and N_u are the minimum, maximum, and control prediction ranges, respectively. x_{r_t} is the target value at time t.

Substituting the prediction $\hat{X}$ in Eq.(4.16) into the cost function J in Eq.(4.17), it can be expressed in the quadratic form of ΔU_t. Therefore, the optimal ΔU_t can be obtained from the equation in which the partial derivative of the evaluation function J with respect to ΔU_t is a zero vector.

$$\Delta U_t = (\mathbf{G}_s^T \mathbf{Q} \mathbf{G}_s + \mathbf{R})^{-1} \mathbf{G}_s^T \mathbf{Q} (X_{r_t} - H_s x_t) \tag{4.19}$$

In here, when a matrix of l rows and m columns is counted as one block, $\mathbf{G}_s$ is a matrix of $N_2 - N_1 + 1$ block rows and N_u block columns, whose (i_{th}, j_{th}) block components are as follows

$$\mathbf{G}_s(i,j) = \begin{cases} \mathbf{C}_e \mathbf{A}_e^{N_1+i-j-1} \mathbf{B}_e, & N_1 + i - j - 1 \geq 0 \\ \mathbf{0}_{n \times m}, & N_1 + i - j - 1 < 0 \end{cases} \tag{4.20}$$

$$\mathbf{H}_e = \begin{bmatrix} \mathbf{C}_e \mathbf{A}_e^{N_1} \\ \mathbf{C}_e \mathbf{A}_e^{N_1+1} \\ \vdots \\ \mathbf{C}_e \mathbf{A}_e^{N_2} \end{bmatrix} \tag{4.21}$$

Even though, this ΔU_t gives control inputs for the time step from t to $t + N_u - 1$, those control will be open-loop control, since the state of the controlled object is only taken into account at the current time. Therefore, only the first block of ΔU_t is applied to the actual control target.

$$\Delta u_t = [\mathbf{I}_m, 0, ..., 0]\Delta U_t \tag{4.22}$$

The control input after time $t+1$ can also be obtained by the same procedure as the derivation of the control input at time t.

4.5 Finding Optimal Observables for Manipulator's Dynamics

This section introduces the problems to be considered when actually applying the KP to a linearization of robot arm's dynamics. The primary concern when introducing the KP is the setting of the observables. For the robot arm, Ian Abraham and Todd D. Murphey show examples of state quantities actually used [61]. They used 7 Dof serial link robotic arm " Sawyer". They used distinctive method, in which the state of adjacent joints are multiplied to obtain observables, is here called the $n-th$ order product method.

$$\begin{gathered} z(x) = [x^T, 1, q_1q_2, q_2q_3, ..., q_6^3q_7^3, \dot{q}_1\dot{q}_2, ..., \dot{q}_6^3\dot{q}_7^3] \in \mathbb{R}^{51} \\ x = [q_1, ..., q_7, \dot{q}_1, ..., \dot{q}_7]^T \end{gathered} \tag{4.23}$$

Where, $q_i, \dot{q}_i$, $i = 0, ..., 7$ are $i-th$ joint position and angular velocity. Whether this choice of observables is optimal for describing robot arm's dynamics is still a matter of debate. If the amount of observables is increased without limit, the state matrix of the resulting state-space model is no longer a full-rank matrix, and redundancy is created in the computation of the inverse matrix. This has a negative impact on the calculation of optimal control inputs in model-based control such as GPC. Furthermore, including excessively large matrices in the model increases the computational cost and prevents quick prediction and control. Hence, it is ideal to choose functions and state variables that are appropriate for that dynamical system and minimize the size of the observables.

4.5.1 Referring Robot Dynamic Equation

To define optimal observables, the robot arm's dynamical equation is used. The idea is that the monomial that appears when the dynamics of the robot are truly expressed in the form of a linear combination, should be the ideal choice of observables, and its size should be optimal without excess or deficiency. For the k-link serial link robot arm, the dynamic equation for joint angular acceleration and its monomial expansion can be expressed as below.

$$\begin{gathered} \ddot{q} = \mathbf{M}_{(q)}^{-1}(\Gamma - \mathbf{H}_{(q,\dot{q})} - fric_{(q,\dot{q})}) \\ \Downarrow \text{Expand} \\ \begin{cases} \ddot{q}_1 = c_{01} + c_{11}g_{1(q,\dot{q})} +, ..., c_{N1}g_{N(q,\dot{q})} \\ \vdots \\ \ddot{q}_k = c_{0k} + c_{1k}g_{1(q,\dot{q})} +, ..., c_{Nk}g_{N(q,\dot{q})} \end{cases} \end{gathered} \tag{4.24}$$

Where, $q, \dot{q}, \ddot{q}$ are vectors of joint position, angular velocity, and angular acceleration respectively. And Γ is input torque vector, $\mathbf{M}_{(q)}$ is inertia matrix, $\mathbf{H}_{q,(\dot{q})}$ is torque vector that is originated from gravity, centrifugal and Coriolis, $fric_{q,\dot{q}}$ is friction torque. For the expanded form, c_{ij}, $i = 0, ...N$, $j = 1, ..., k$ is coefficients for monomial $g_{i(q,\dot{q})}$.

Ideally, if this monomials $g_{i(q,\dot{q})}$ is included as an observable, the continuous-time operator that gives joint angular acceleration is obtained exactly. However, the monomials that actually appear in the equation include fractions, and the further it go to the link at the end-effector, the more complex the monomials becomes. For example, some of the monomials in equation Eq.(4.24) looks like below.

$$\begin{aligned} g_{ex1} &= \frac{sin^2(q_1)cos(q_2)}{\alpha(\beta - \gamma\, cos(q_1 - q_2))} \\ g_{ex2} &= \dot{q}_6 sin(q) \times \frac{1}{g_{ex1}} \end{aligned} \tag{4.25}$$

Thus, in the process of equation expansion of the dynamics Eq.(4.24), there is a form in which linearly combined monomials are included in the denominator. Furthermore, the further one goes to the equation about the end-effector, the more these complex monomial divisions appear, so the structure of the monomial becomes even more complex, with a very long linear combination in the denominator. Above all, the number of monomials grows exponentially, and the size of the observables becomes too large.

In order to overcome the complexity, concentrating on monomials that appear frequently in the Eq.(4.24), we extract them and employ them as observables. By doing so, we aimed to approximate the KP. In theory, the size of the KP is infinite-dimensional, and when applied to a real system, the resulting operator will be an approximation no matter what. The important question is which set of observables can reproduce the behavior of a nonlinear dynamical system with sufficient accuracy at a realistic size.

In this study, the dynamics calculation software SYMORO+ [102] was used to write down the dynamics equation for the robot arm. Also, in order to reduce computation intensity, 3 Dof serial link robot arm is treated as an example in this research. First, SYMORO+ outputs symbolically expressed dynamic equation Eq.(4.24) in MATLAB function file. Then, by counting operators such as $+, -, *, /$, the tendency of simple forms that appear in monomials is determined. As a result, the following configurations seem to be primal.

1. trigonometric functions of joint angles and their multiplication (e.g. $sin^{n_{s1}}(q_1)cos^{n_{c1}}(q_1)$, $sin^{n_{s2}}(q_2)cos^{n_{c2}}(q_2)$, $sin^{n_{s3}}(q_3)cos^{n_{c3}}(q_3)$)
2. multiplication of joint angular velocities
   (e.g. $\dot{q}_1^{n_a}\dot{q}_2^{n_b}\dot{q}_3^{n_c}$)
3. multiplication of 1 and 2

The number of monomials used for these multiplications (ex. $n_a + n_b + n_c$) is expressed in terms of "order" hereafter. Some of these orders may exceed 10, but their effects are so nonlinear that they behave like noise and their share in the overall dynamics is limited. The maximum order in each case was therefore defined as follows.

$$\begin{aligned} &1)\ max(\textstyle\sum_{i=1}^{3} n_{si} + n_{ci}) = 5 \\ &2)\ max(n_a + n_b + n_c) = 3 \end{aligned} \tag{4.26}$$

At this point, the total number of employed observables is 9243. The KP identified from these observables will be 9243×9243, which is too computationally expensive to incorporate into GPC. Furthermore, this KP is less likely to be full rank. Therefore, it is necessary to eliminate the unnecessary monomials from the observables in the Eq.(4.27).

$$
\begin{aligned}
z(x) = [1, x^T, \dot{q}_1\dot{q}_2, ..., \dot{q}_3^3, sin(q_1), ..., cos^5(q_3), \\
\dot{q}_1 sin(q_1), ..., q_3 cos^5 q_3] \in \mathbb{R}^{9243} \\
x = [q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3]^T
\end{aligned}
\tag{4.27}
$$

4.5.2 Extract Important Monomials

To determine unimportant monomials, we first approximated the KP using the observables in Eq.(4.27) Here, as described in Chapter 4.3.2, more than 9243 data points were required, so 10000 data points are provided with 10 Hz sampling. The motion was controlled with PD control with gains that were tuned based on Franka-ROS package. The motion consisted of random safe postures of the panda robot smoothly connected by spline interpolation. The duration between postures gradually varied between 1 and 10 seconds, and movements covering a wide range of velocities were recorded. The data was acquired through Gazebo simulator with ROS Melodic environment.

Table 4.1: Gains for PD Control

	Joints						
	1st	2nd	3rd	4th	5th	6th	7th
P gain	30	60	35	35	30	15	5
D gain	0.3	0.6	0.3	0.4	0.4	0.3	0.1

After KP were identified, the columns were taken one by one, their ranks were calculated, and the columns with rank failures were determined by repeating the process until the ranks fail. The optimal observables to be sought were then determined by excluding the monomials associated with that column as shown in Algorithm 1. In this process, low-rank matrix approximation method is used. This low-rank approximation is called a matrix sketch [125]. The matrix sketch only preserves important features of the original matrix, filtering unnecessary information out. The tolerance for relative approximation error of the matrix sketch is set to 0.1.

Algorithm 1 Calculate Optimal Observables

Ensure: $rank(\mathfrak{K}) = length(K)$
1: $K \Leftarrow \mathfrak{K}(:, 1)$
2: $i \Leftarrow 1$
3: **while** $rank(K) < rank(\mathfrak{K})$ **do**
4: $\quad i \Leftarrow i + 1$
5: $\quad$ **if** $rank(K) == length(K(1, :))$ **then**
6: $\quad\quad K \Leftarrow [K, \mathfrak{K}(:, i)]$
7: $\quad$ **else**
8: $\quad\quad K(:, end) \Leftarrow [\,]$
9: $\quad$ **end if**
10: **end while**

In Algorithm 1, K is a submatrix of KP $\mathfrak{K}$, extracting the part of the KP that concerns important monomials. By checking the column numbers of the remaining KP that is in K, against the order of the observables in Eq.(4.27), finalized observables are defined. The size of the observables is $z(x) \in \mathbb{R}^{99}$. The all monomials of the robot arm's dynamics and selected monomials by Algorithm 1 is available hare `https://github.com/shohei1536/Extracted_monomials`.

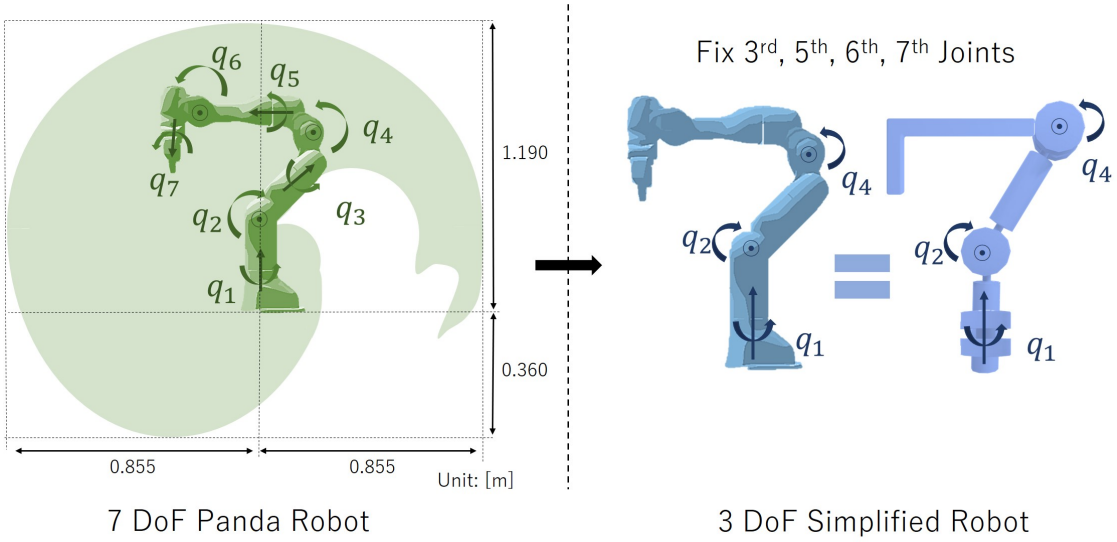

Figure 4.1: Panda robot and simplified robot

4.6 Control Experiments and Results

For the control experiments, Panda robot from Franka is used. Although it has 7 Dof, it is treated as 3 Dof robot by disabling $3, 5, 6, 7 - th$ joints. The experiment is done in the Gazebo simulation with Ubuntu18 and ROS Melodic environment. The experiment section has 3 parts.

1. Comparison of Performance of the Linear Predictor
2. Comparison of Performance Within a Task Space
3. Comparison of Adaptability to Different Motions

In 1), linear predictors from the proposed observable set and the linear predictors derived from observables used in the previous study were compared. Secondly, in 2) We compared the end-effector tracking performance in the task space for the GPC from the linear predictor in the previous study and our GPC and PD controller. Finally, in 3) tracking performance over a wider range of motion conditions (position, velocity, and acceleration) was compared in joint space.

4.6.1 Comparison of performance of the linear predictor

First, the KP $\mathfrak{K}$ that contains the linear predictor is shown in Fig.4.2 As explained at the very end of Section 4.5, the number of our observables is 99, so the KP identified from here takes the form of a matrix of size 99×99. From this matrix, the linear predictor that is explained in Eq.(4.2) was acquired as explained in Eq.(4.8) and Eq.(4.9). The KP was identified from 110000 data points that are 10 Hz recordings of PD-controlled motions. The motions were made as explained in section 4.5.2. To get motion data that contain a wide range of position, velocity, and acceleration, the duration time between postures was gradually varied between 1 and 10 seconds. Moreover, very slow motion where the duration between postures was 50s was also added to data for identification. Henceforth, we will refer to the KP, identified from these observables we have proposed, using the "Opt" adjunct in the following figures.

As can be seen from Fig.4.2, the diagonal components of the identified KP tended to take values close to 1.

The prediction of the robot's behavior using the state-space model obtained from the KP is shown in Fig.4.3. The figure is about the 4th joint, where the most nonlinear behavior appears among the 1st, 2nd, and 4th joints. The left figure shows the joint position and the right figure

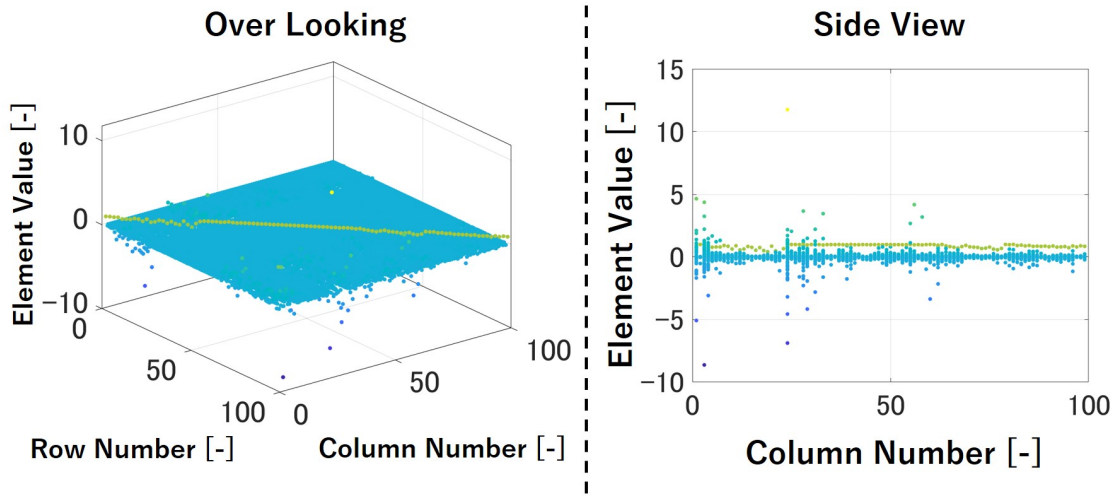

Figure 4.2: Identified Koopman operator $\mathfrak{K}$ from our observables. Its diagonal elements are highlighted in green and have distinct tendency where these value is almost one

shows the joint angler velocity. Predictions are shown for 90 steps (0.9 seconds) made at 100 [Hz]. Actual robot data are shown as solid lines and predictions are shown as dashed lines.

As mentioned at the end of section **??**, it is generally not possible that the trajectory of linear predictors matches the trajectory of the actual system for over an infinite period of time. On the other hand, these predictors are accurate for a prediction steps of more than 20 (0.2 seconds). Therefore, the predictor is sufficient for model base control such as GPC with prediction window of 0.2 seconds.

Besides our prediction with Opt observables, as a comparison, predictions from n-th order product methods were plotted (see (4.23)). Even though in the previous study of Ian et al., the robot's DoF was 7, the n-th order product method was applied to the 3 DoF by maintaining the principle of multiplying the state of adjacent joints. In addition to the observables of 3rd orders that was introduced in (4.23), the observables sets of 1st order, 2nd orders, and 4th order were also plotted in Fig.4.3. As Fig.4.3 shows, our predictor is slightly more accurate than the others, especially for the velocity prediction. For example, the average value of the mean absolute error (MAE) of joint position for 30 different trajectories is; our Opt predictor was $0.0368[rad]$ and for the 3rd order product predictor is $0.0374[rad]$, and MAE of joint angular velocity for our predictor was $0.0895[rad/s]$ and $0.1137[rad/s]$ for the 3rd order predictor.

4.6.2 Comparison of performance within a task space

The GPC was designed using the linear predictor obtained in the previous chapter, and its performance in the task space of the robot was compared to that of the PD controller. The task is to draw the shape of 8 horizontally five times using only 1st, 2nd, and 4th joints. Fig.4.4 shows the results with the time taken for one cycle set to 3 seconds and 10 seconds, respectively. The control parameters of GPC are set as $N_1 = 1, N_2 = 20, N_u = 10, R = diag(0.1), Q = diag(1.0)$. Here, $diag(x)$ denotes the diagonal matrix in which each elements have value of x.

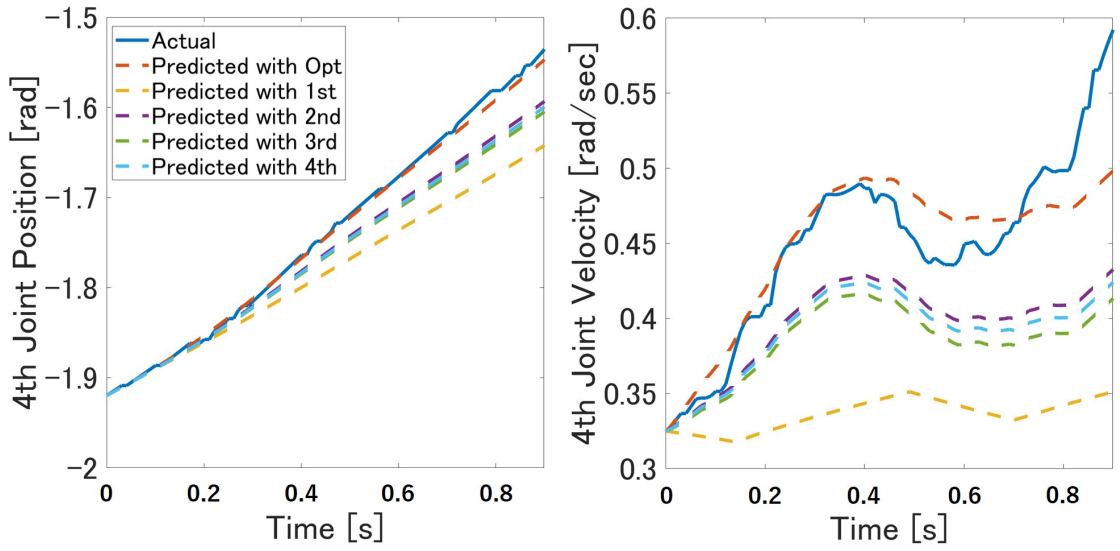

Figure 4.3: Comparison among linear predictors and actual robot arm's behavior. In the legend section, 1st, 2nd, 3rd, 4th denote the predictor with previously suggested observables sets. And Opt denotes the predictor with proposed observables.

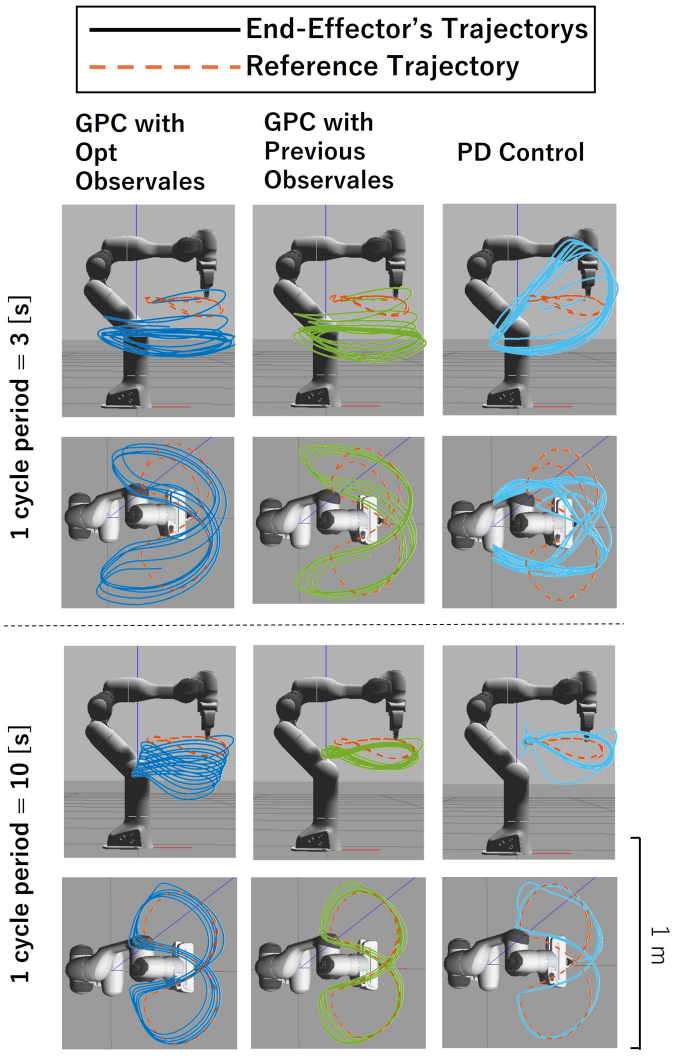

Figure 4.4: Control result of GPC and PD controller in task space. The duration of one cycle was 3 seconds and 10 seconds.

4.6.3 Comparison of adaptability to different motions

To see the control performance of GPC against wide range of joint position, velocity and acceleration, the exciting motion and RMSE in joint space was compared among our GPC, PD control and GPC from 3rd linear predictor. The motion was made as explained in section 4.5.2. The duration time between postures was set between 1 and 5 seconds. The measurement sampling rate was 100 Hz and duration was 30 s.

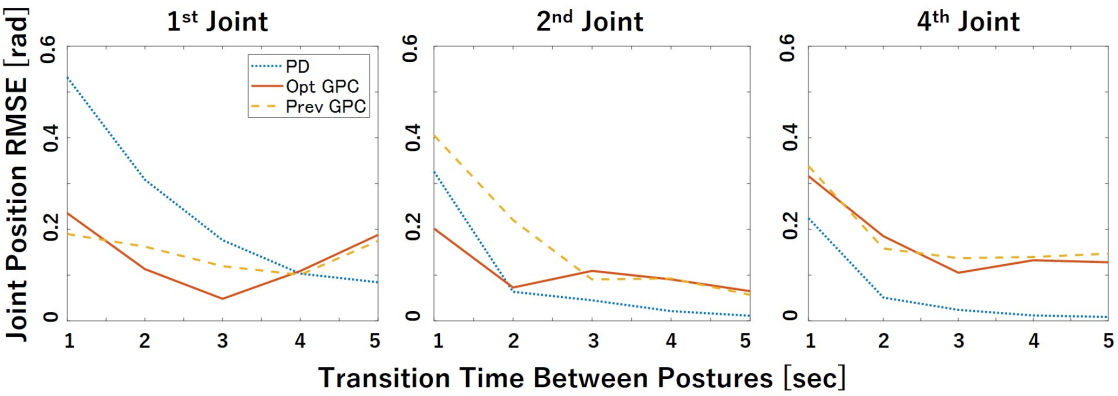

Figure 4.5: RMSE of joint position for 1st, 2nd and 4th joint in case of excited motions

4.7 Discussion

About section 4.6.1, our KP has distinct features, its diagonal elements have value of almost 1. This is representing the physical fact that the observables has continuity in transitioning to the next state. Each element of this matrix $\mathfrak{K}$ represents the degree of influence that the observables have on each of the observables' next stats. The final three rows of this matrix are larger than the other elements, but this is a meaningless byproduct, as they represent the degree of influence that the observables have on future changes in the input.

In the result of Fig.4.3, it should be noted that even the observables of 2nd and 3rd order that has size of only $\mathbb{R}^{18}$ and $\mathbb{R}^{22}$ can predict robot motion similarly to our linear predictor which have size of $\mathbb{R}^{99}$. On the other hand, as our predictor has better prediction, more complex monomials are included in the observables, the better the predictor might be.

About section 4.6.2, Even though our GPC could control robot stably, the result was not the best. Our GPC suffers from the effect of gravity and the end-effector was shifted to a lower position compare to the GPC with 3rd order observables in case of a slow task (Fig.4.4). This is thought to be because the larger the size of the state-space model matrix, the more room there is for model error, and the more the errors add up.

When we compare the fast task and slow task, GPC controllers keeps its performance, while PD controller lost its performance. This tendency can be seen in the result of section 4.6.3. For the 1st and 2nd joint, our GPC Could keep control performance better than PD control in the case of fast motions. This high responsiveness to the slackness of the target trajectory can be seen in the results of Fig.4.4. Here, the PD control is wobbly because the target speed changed abruptly at the start of the movement. On the other hand, GPC adapts appropriately to this abrupt change in target speed.

Model errors are mentioned here. In the proposed method, the number of monomials constituting the observables was carefully selected, resulting in the identification of approximate Koopman operators. As a result, the accuracy of the linear predictor deteriorated, resulting in control errors

as shown in the figure 4.4. Figure 4.6 shows the relationship between the different observables and the accuracy of the linear predictor. This is shown for the fourth joint of a 7-DOF robot. In the proposed method, the accuracy of the linear predictor decreases as the number of observables is reduced. On the other hand, the linear predictor in the previous study achieves better accuracy with fewer observations. It is believed that there is a way to select more suitable observables.

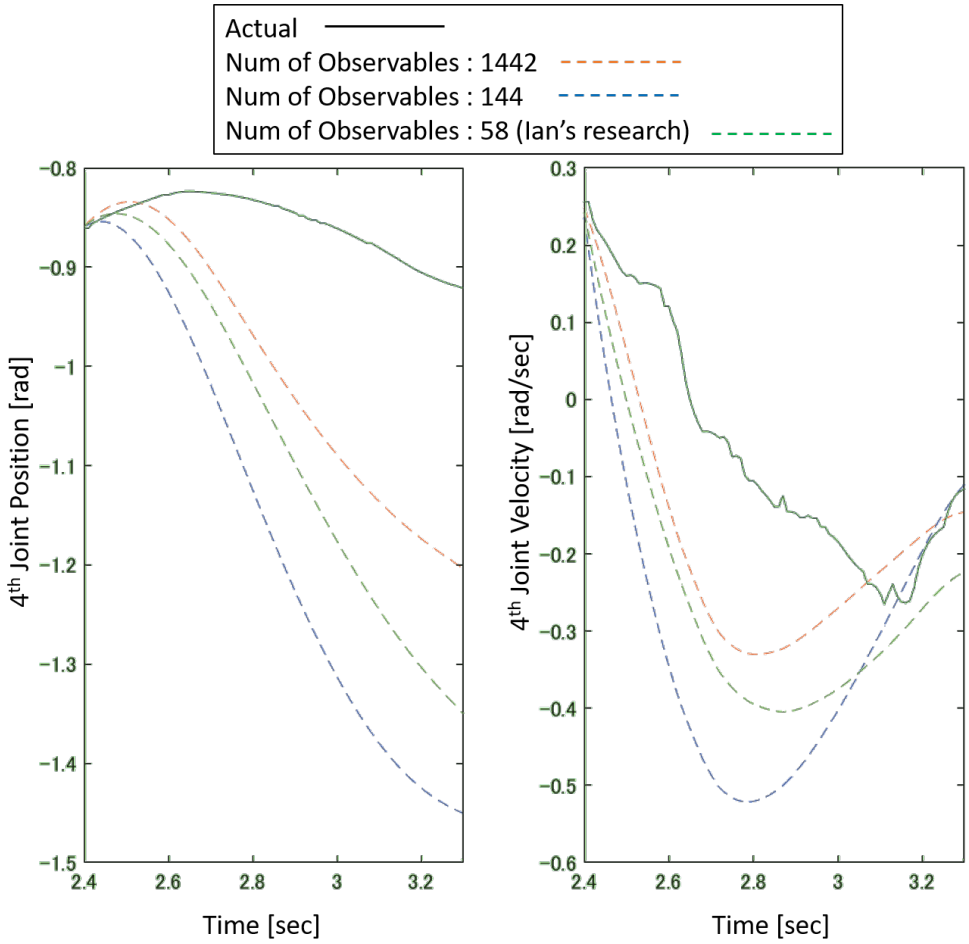

Figure 4.6: The relationship between the different observables and the accuracy of the linear predictor. This is shown for the fourth joint of a 7-DOF robot. In the proposed method, the accuracy of the linear predictor decreases as the number of observables is reduced. On the other hand, the linear predictor in the previous study achieves better accuracy with fewer observations. It is believed that there is a way to select more suitable observables.

Here, we discuss the advantages of choosing a large size observable over smaller size such as the previous study. The advantage is that the KP identified for a larger size of observables contains more information about the system. For example, the KP we have identified here, modified into a continuous time operator $\mathfrak{L}$ using (4.13), composed of elements that indicate the degree to which the input affects the angular acceleration. This is shown in Fig.4.7. This figure shows the evolution of the value of the element indicating the degree of influence of the input on the angular acceleration in the identified continuous time operator $\mathfrak{L}$ as the mass of the end-effector changes from 0.5 kg to 2.5 kg which shape is 10 [cm] diameter sphere. The approximate line is the least-squares approximate line. The results show that the information of the mass of the end-effector is reflected in the operator's value.

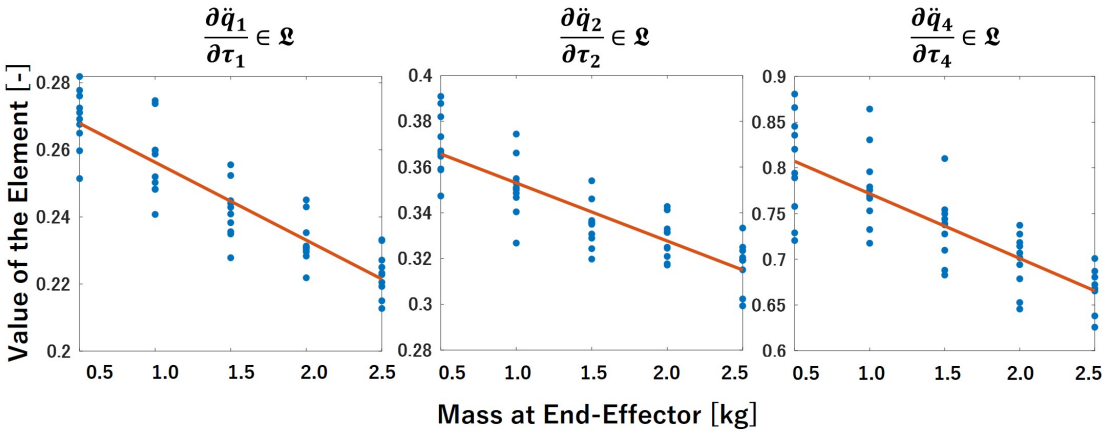

Figure 4.7: Change in the value of elements of the continuous-time operator $\mathfrak{L}$ in response to a change in the end-effector's mass (about 1st, 2nd, and 4th joint)

In addition to the intuitive elements as shown in Fig.4.7, the information of the end-effector mass is also reflected in the non-intuitive observables as shown in Fig.4.8. This suggests that information-rich KP can be analyzed using PCA or auto-encoders to estimate the mass and even the shape of the object at the tip of the hand. Moreover, if the observables are set appropriately, it could be possible to estimate the dynamical information of attached objects other than the end-effector such as anti-collision retrofit parts.

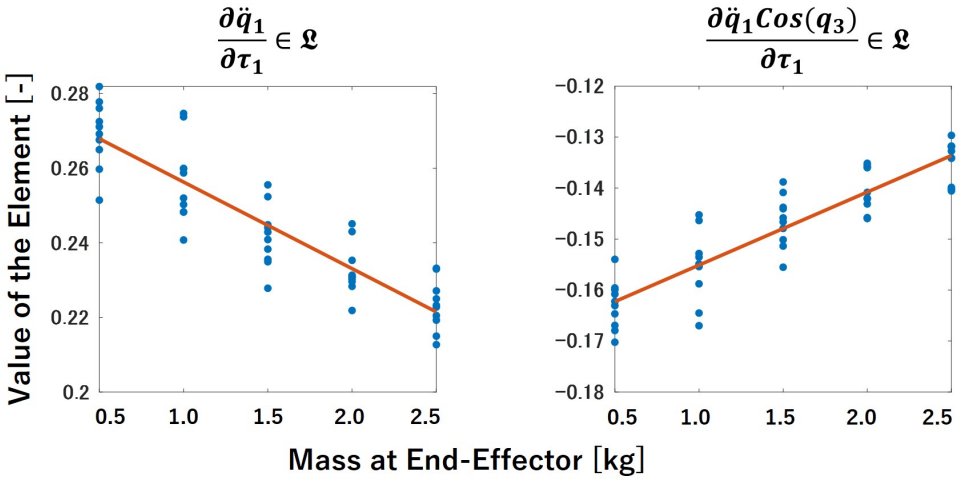

Figure 4.8: Change in the value of elements of the continuous-time operator $\mathfrak{L}$ in response to a change in the end-effector's mass (about 1st joint)

4.8 Conclusion

This chapter presents an example of the linearization of a robot manipulator's dynamics using Koopman operators (KP), and presents practical issues to be considered when applying the principle of KP theory to robotic systems, and introduced the advantages of setting up large-size observables. Using the KP, this work presents the successful design of a Generalized Predictive Controller (GPC) that is able to control robot position and velocity stably under a wide range of posture/velocity/acceleration for a 3-Dof robot arm. Control results show that our GPC performance outperforms a traditional PD controller in terms of adaptability to the diverse target trajectories.

This study contributes to one of the few instances in which the KP has been introduced into the serial link robot arm system. Its basic idea is to lift a non-linear function to a higher dimensional space by extending its explanatory variables, and these new variables are often referred to as "observables". In this higher dimension space, nonlinear functions can be described in a linear form. Here, how the observables should be augmented depends on the dynamical system's property. For the robotic manipulator, still, it remains unclear what the best observables set to describe its dynamics is.

In this study, the optimal observables choice was acquired by expanding the robot dynamics equation and using its monorails. The KP was then identified by the least-squares minimization, and an approximate linear state-space model of the 3 DoF simplified Panda robot was identified. Using this, GPC was designed, and it was compared with PD controller. Fig.4.3 shows that our observables choice was properly describing the non-linear behavior globally and made the linear state space model as accurate as previous research. Using these linear predictors, GPC was designed and its performance in the task space and joint space was shown in Fig.4.4 and Fig.4.5. The results depict GPCs have more adaptability to the different motions velocities than PD control.

Finally, the advantage of choosing a large size observable to identify KP was explained using Fig.4.7 and Fig.4.8. The advantage is that the KP identified for a larger size of observables contains more information about the system. The results show that the information of the mass of the end-effector is reflected in the operator's value. This suggests that information-rich KP can be analyzed using machine learning techniques such as PCA or auto-encoders to estimate the mass and even the shape of the object at the tip of the hand. Moreover, if the observables are set appropriately, it could be possible to estimate the dynamical information of attached objects other than the end-effector. There may be anti-collision retrofit parts, and people may personalize their home robot with decorations. More detailed identification will be our future research direction.

4.9 Publications

The following list contains the published and submitted papers regarding this doctoral work.

Conference

- S. Hagane, L. Jamone, G. Venture, Linearizing Robotic Manipulator's Dynamics Using Koopman Operator and Applying Generalized Predictive Control, 2023 IEEE/SICE International Symposium on System Integration (SII2023), WeP2M2.6 Atlanta USA, January 2023

Chapter 5

Robotic Manipulator Expressing Emotions as Sub-Task Using Kinematic Redundancy

SUMMARY

In the last study, we proposed a motion control system that uses the kinematic redundancy of the robot arm to perform emotional expressive motion when the robot arm performs a main task at its end-effector. As mentioned earlier, in order for robots to play an active role in society, they must be able to perform both communicative and productive tasks. Here, we used null-space control to exploit kinematic redundancy and applied the concept of manipulability ellipsoid to maximize motion within the null space. The proposed method was implemented on an actual upper body humanoid robot "Nextage-Open" and recorded the interactions with a person. Using the recordings, the robot's emotions were evaluated online by more than 200 people via Google-Form. The results showed that current rules surrounding industrial machines operating near people severely limit the robot's speed and prevent it from expressing the movement-by-movement changes necessary to express emotion. However, we found that people who viewed unclear emotional behavior used the content of the interaction, the robot's gaze, and their own prejudices about the robot, in addition to the emotional behavior, to interpret the robot's emotions.

5.1 Introduction

As social robots become more integrated into daily life, an important question arises: how can robots become more acceptable for people? In the 2017 Pew Research Center survey "Automation in Everyday Life", 4135 people in the United States were asked, "Are you interested in using robots in your daily life, including care-giving?" Out of the respondents 64% had negative perceptions on robots, stating that they "can't read robots' mood" and that robots are "simply scary" [6]. One of the ways being explored to gain user acceptance is to incorporate emotional movements. In order to overcome the negative reputation of robots, researchers have been implementing emotions into robots which are expressed towards the human user [8, 9, 10, 11]. This can be done by displaying affective content on the screen and/or through expressive sound and movements. One of the main purposes of robots expressing mood/emotion is that it gives social value to interactions between people and robots. In other words, when people empathize and socialize with robots, the robots are no longer recognized as simply a "tool" but as something social, which can help to appease feelings of aversion and dislike for robots.

Movement is often difficult to handle as a tool for emotional expression. Especially for robotic arms, movements have a significant impact on the main task, so there are few attempts to even make them express emotional motions. Popular tools for robots to share emotions are optical devices, such as displays that show animated facial expressions, and audio devices. However, it is questionable whether the robot's internal information expressed in voice and display alone is

sufficient to facilitate human sociability. In fact, in Brscic's work, the robot with voice device communicating its emotion and wish is abused by children in a shopping mall [126].

To state human basic reaction, the social mind is attributed to empathy, and empathy is triggered by the strength of the observed emotion [127]. According to a book "Introduction to Social Psychology : a European Perspective" by M. Hewstone, an emotion can be seen as "an episode of interrelated, synchronized changes in five components in response to an event of major significance to the organism" [128]. These five components are: the cognitive processing, the subjective feeling, the action tendencies, the physiological changes, and the motor expression. And C. Clavel pointed out that the synchronization of expressions of emotion across different components is expected to have major importance to human understanding of emotion perception. [129]. Therefore, It is important to make the emotions that people perceive strong not only through easily neglected expressions such as light and sound, but also by simultaneously performing the realistic element of movement.

Researchers have been trying to convey certain emotions through robot motions (see section 2.5). For instance, Knight attempted to use a small robot that only has a few degree of freedom [8], and Beck et al. used a humanoid that utilize human gesture information to generate expressive robot motions [9]. In these studies, the general tactics for differentiating emotions are adjusting pre-existing gestures/postures and/or the velocity of the motions. Those attempts have successfully conveyed arousal and mood through robot motion.

However, most research on emotional robots treat emotion conveyance as the main task, so the robots cannot do any other productive tasks such as manipulating objects. People's expectations of robots are becoming clearer, and the image of a convenient existence to do work is generally an important element in the image of robots. The reality is that household robots such as Vector and Kuri, which are only for sharing emotions, are forced to withdraw from the market even though these companies were viewed as promising [16]. This trend implies that, in order for a robot to be accepted, it should have both a productive function and an element that appeals to human sensibilities.

In this research, we develop a motion system that performs tasks while expressing emotions using the kinematic redundancy of robot arms. A system for performing emotional actions using redundant degrees of freedom while performing industrial tasks has not yet been demonstrated in the social robot field. Although Claret et al. have shown an example of a humanoid (Pepper) that utilize the redundant degrees of freedom to express emotion, its main task was a waving gesture and emotion conveyance, while productive task were not implemented [10]. In contrast, the target of our study is a robot arm that excels in practical use makes it more aware of actual operations in real life situation.

People's feelings toward robots are still mostly negative as mentioned above. Yamamoto et al. discussed human behavior toward computer programs, i.e., pseudo-interpersonal behavior [12], and found that the "improvement of system functions alone does not cause pseudo-interpersonal behavior, but some other element is necessary." This "element" is the social value of human-robot interaction, and our research is an effort towards creating that social value. We aim to change the perception of robots as a mere tool through the emotional expression of robots which can draw out human empathy in a natural way. This should significantly impact the way robots will be used in the service industry in the future. For example, it may prevent robots being used in public from being harassed by passersby. Particularly in the service industry[13].

Fig.5.3 provides an outline of our research. Chapter 5.2 describes the kinematics model for redundant robot manipulators. Chapter 5.3 introduces the proposed emotion conveyance method. Chapter 5.4 describes the implementation of the method on a real robot, questionnaire contents, and experiment. Chapters 5.5 and 5.6 present the experimental results and the discussion, respectively. Chapter 5.7 concludes this paper.

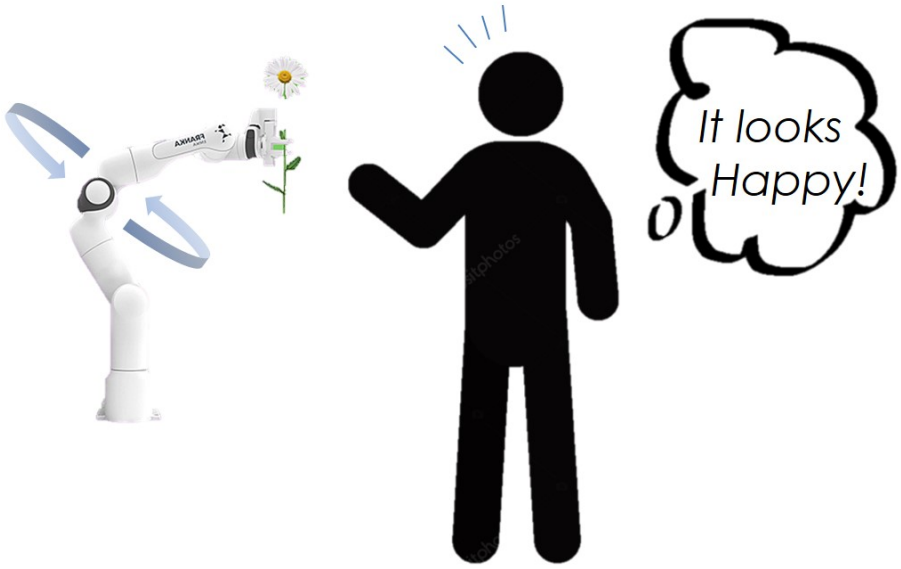

Figure 5.1: The main idea is the coexistence of productive tasks and emotional expression behavior. The robot's mood is communicated to the observer through emotional movement within a range of redundant degrees of freedom so as not to affect the main (productive) task at hand.

5.2 Robot Manipulator with Kinematic Redundancy

The definition of a redundant manipulator and its kinematics are described here. A redundant manipulator is defined as a manipulator that has more degree of freedom (DoF) than a task requires. The Jacobian is related to the DoF of the end-effecotr. In another words, it indicates the direction in which the end-effector can move at that joint coordination $\boldsymbol{q}$. The maximum rank of this Jacobian M is equal to the DoF of the task in Cartesian space. Here let's assume that the manipulator has N DoF and the rank of Jacobian r, there would be three cases as showed below.

1. In the case of $(N = M)$.
   The manipulator has exact number of DoF that required to complete the main task at its end-effector.

2. In the case of $(N > M)$.
   The manipulator is redundant with degree of $(N - M)$. The manipulator has ability to move along its redundant space without moving its end-effector and that kinetic redundant space is often refferd as *Null Space* in robotic motion control.

3. In the case of $(r < M)$.
   The manipulator is at the singular pose with degree of $(M\text{-}r)$ and its end-effector can not move to the certain direction (degenerate direction). If the Jacobian is square matrix, this is the case when the determinant of the matrix is 0. On the other case, such as the case of the redundant manipulator, it is the case when the determinant of the product of Jacobian and its transpose is 0.

5.2.1 Kinematics of Robotic Manipulators

In this study, the robot's emotional motion was generated via *Null Space Control* [82] and it is closely related to kinematics such as Jacobian matrix. So let us note the basics of manipulator's kinematics in this section. First, in the inverse kinematics of the robot arm, the relationship between the position and orientation $\boldsymbol{X}$ in three-dimensional space and the joint angle vector $\boldsymbol{q}$ can be expressed as $\boldsymbol{X} = f(\boldsymbol{q})$ using a certain nonlinear function f. Then, the differential relation in

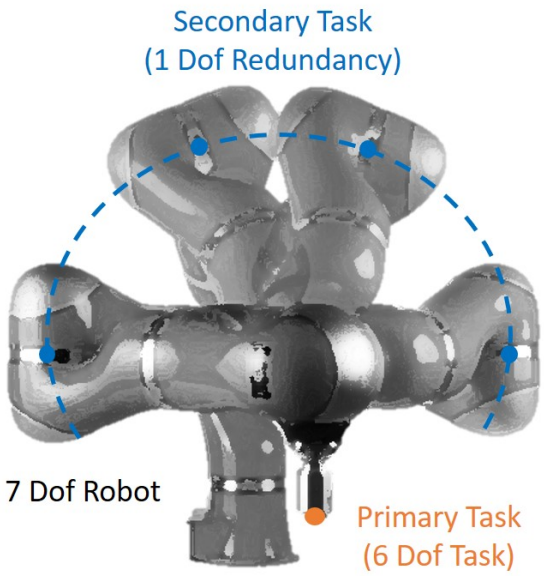

Figure 5.2: A redundant robotic manipulator can move parts of the body without affecting the position and/or posture of its end-effector

that equation can be expressed by a certain linear expression with Jacobian $\mathbf{J}(\boldsymbol{q}) := \delta f / \delta \boldsymbol{q}$ as the coefficient matrix.

$$\dot{\boldsymbol{X}} = \mathbf{J}\ \dot{\boldsymbol{q}} \tag{5.1}$$

There are two approaches to inverse kinematics with redundancy: one is to solve equation $\boldsymbol{X} = \boldsymbol{f(q)}$ using numerical methods such the Paul method [102], and the other is to solve equation 5.1. However, in the case of singular postures, since Jacobians do not have inverses, Moore-Penrose pseudo-inverses are often used.

$$\dot{\boldsymbol{q}} = \mathbf{J}^{+}\ \dot{\boldsymbol{X}} \tag{5.2}$$

However, in singular postures, even pseudo-inverse matrices can deteriorate the accuracy of the computation, and may generate discontinuous values when incorporated into a closed-loop control system. Therefore, in this study, the SR-Inverse (singularity robust inverse) was used [130].

$$\mathbf{J}^{\#} := \mathbf{J}^{\mathbf{T}}(\mathbf{J}\mathbf{J}^{\mathbf{T}} + k\mathbf{I})^{-\mathbf{1}} \tag{5.3}$$

where $\mathbf{I}$ is the identity matrix and k is a very small positive arbitrary constant. By using this SR inverse matrix, it avoids dealing with exact values in the process of calculating the inverse matrix in singular postures, and thus prevents discontinuity in the time series calculation results.

5.2.2 Null Space Control

Null space control is a method of controlling robot motion in the space of joint angular velocity or joint angular accelerations in degrees of freedom that are not used in the main task [82]. There are many studies that take advantage of the kinematic redundancy of robots as mentioned in Chapter 2.

The null space projection matrix $\mathbf{N}$ can be calculated from Jacobian.

$$\mathbf{N} = \mathbf{I} - \mathbf{J}^{\#}\mathbf{J} \tag{5.4}$$

In this research, since a 7-DOF manipulator is used, at least one redundant DoF is basically ensured as shown in Fig.5.2, but depending on the type of task, there may be more than two redundant DoFs. For example, the task of moving a ball along a fixed trajectory is represented by 3-DoFs ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$) for position only if the rotation of the ball ($\theta_x, \theta_y, \theta_z$) are arbitrary, which means that the 7-DoF manipulator has 4th degree of redundancy. In order to effectively utilize this redundancy, we devise a new method for the Jacobian $\mathbf{J_0}$ that is used to calculate the null space projection matrix $\mathbf{N}$.

$$\begin{aligned}
\mathbf{J_0} &= (\mathbf{J^T I_0})^{\mathbf{T}} \\
\mathbf{I_0} &:= \begin{bmatrix} i_x & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & i_{\theta_z} \end{bmatrix} \\
i_\alpha &= \begin{cases} 1\ (\alpha \in \mathit{main\ task}) \\ 0\ (\alpha \notin \mathit{main\ task}) \end{cases} \\
\alpha &\in \left[x,\ y,\ z,\ \theta_x,\ \theta_y,\ \theta_z\right]
\end{aligned} \tag{5.5}$$

Using this $\mathbf{N}$, any additional motion $\dot{\boldsymbol{q}}_{add}$ can be added to the main task $\dot{\boldsymbol{q}}_{task}$ by projecting it in the null space. And the new motion $\dot{\boldsymbol{q}}_{ref}$ takes a form that not affect the main task at the end-effector.

$$\begin{aligned}
\dot{\boldsymbol{q}}_{ref} &= \dot{\boldsymbol{q}}_{task} + \mathbf{N}\,\dot{\boldsymbol{q}}_{add} \\
\Rightarrow \mathbf{J}^{\#}\dot{\boldsymbol{X}}_{task} &+ (\mathbf{I} - \mathbf{J}_{\mathbf{0}}^{\#}\mathbf{J_0})\dot{\boldsymbol{q}}_{emo}
\end{aligned} \tag{5.6}$$

The final reference trajectory which contain main task and emotional motion $\dot{\boldsymbol{q}}_{emo}$ is given in second line of Eq (5.6). $\dot{\boldsymbol{X}}_{task}$ is the main task of the end-effector. In this research, it is important to define emotional component $\dot{\boldsymbol{q}}_{emo}$ effectively. Because it will be trimmed into the null space and it might loss some important motion feature such as spatial extension of the motions that would convey a dominance level of the mood. To cover this issue, the manipulability ellipsoid is adapted in null space control as it described later section 5.3.4.

5.3 Emotional Conveyance

5.3.1 Proposed Method

Our method is for generating a joint space motion for robotic manipulators. The generated joint space motion contains both, the main task at end-effector and the emotional motion in its null space. In order to express the intended emotion correctly, the corresponding motion features *jerkiness*, *velocity* and *spatial extent* are adjusted. The motion features were selected on the basis of state of the arts studies in the cognitive engineering field. Finally, the adjusted motion features are effectively executed in the null space by using the concept of manipulability ellipsoid.

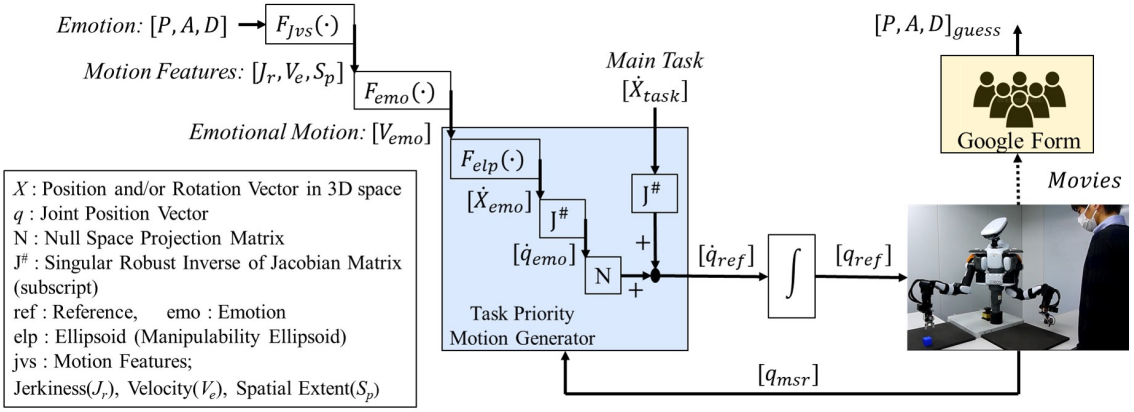

Figure 5.3: Overview of the proposed method of emotional motion conveyance using kinematic redundancy. Emotional information is given to the robot as PAD parameters, which are converted into a set of motion feature parameters J_r, V_e, S_p with reference to previous studies in cognitive engineering. Then, depending on the values of the motion features, the emotional motion in the null space is generated using its manipulability ellipsoid. This emotional motion was performed by the robot through null space control and recorded on video for an online survey.

Like Claret et al.'s work [10], our proposed method convert emotion (PAD) into motion features. However, in contrast to previous work, we concentrate on robotic manipulators that execute productive tasks at the end-effector. While the previous related work focused on humanoids, and set gestures (e.g., waving) as the main task, our method enables robots to be productive and expressive at same time. In addition, our method of converting emotion to motion features differs from that of previous work which converted emotion (PAD) into motion features by a simple linear formula. We use findings from previous cognitive engineering studies [94, 4, 131], focusing on the velocity of the motion to convey emotions. Furthermore, instead of the motion feature *Gaze* that is used in Claret et al.'s work, we introduce *Spatial Extent* to express the dominance of the emotion since manipulators don't have eyes. The degree of spatial extent was represented using geometric entropy [132].

5.3.2 From Emotion: PAD to the Motion Features: $J_r V_e S_p$

The PAD model is by Mehrabian [3] and used to express emotions in numbers. In this model, a certain emotion is described by 3 elements: pleasure P , arousal A and dominance D. Each elements are nearly independent and take values from -1 to 1 contentiously. Thus, different emotions can be expressed as points in the 3D space (PAD space) as shown in Fig.5.4.

Several studies have attempted to correlate human gestures with PAD parameters [133] [94]. However, as many studies have shown, it is not the meaning of the gesture categories that is important for humans to recognize and distinguish emotions in nonverbal communication, but the elements of motion contained in the movements [134] [135] [136] [137] [138]. There are researches on mapping the relationship between movement elements and PAD parameters to make robots perform actions that match emotions [10] [11]. In the study by Claret et al. [10] using the humanoid robot Pepper, Jerkiness and Activity were used as adjustment parameters. Following the work of Lim et al. [11], they mapped the correspondence between these movement features and PAD parameters, generated affective movements, and verified the validity of emotion transfer. In the study by Donald et al. [94], 25 movement elements are described as transmitting emotional information, and in particular, four of them are capable of expressing most emotions. These four movement components are *Activity*, *Excursion*, *Extent*, and *Jerkiness*.the *Activity* is correspond to the energy of movement. The *Excursion* indicates how kinetic energy is distributed throughout the movement, while the *Extent* refers to how large the gestures of the hands and arms are. *Jerkiness* is the derivative of acceleration.

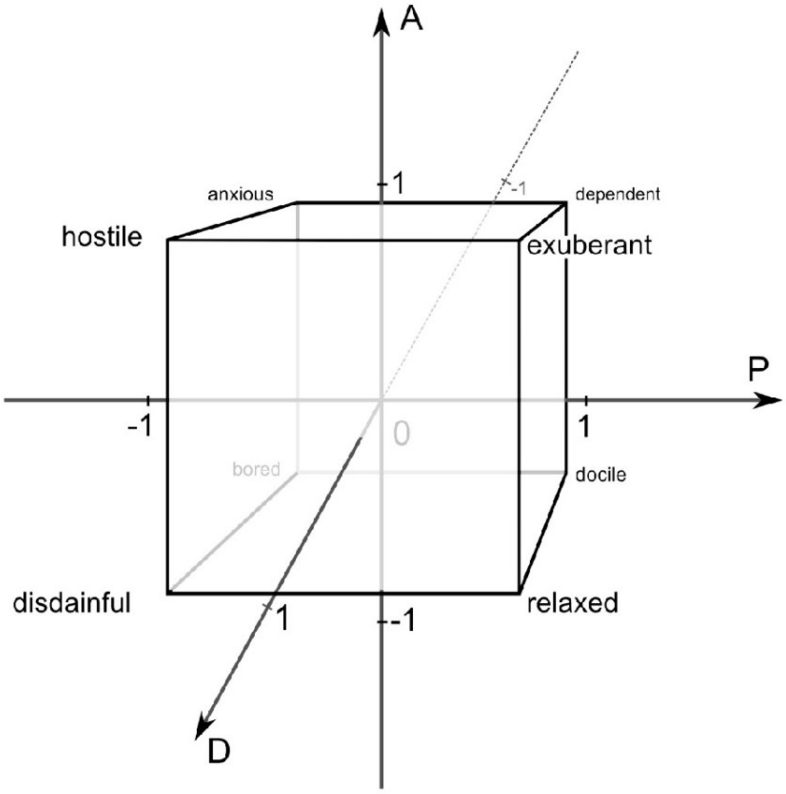

Figure 5.4: PAD space model by Mehrabian [3]: A single emotion can be broken down into three components: arousal, pleasure, and dominance. Each components takes a value between -1 and 1.

a) Kinetic Energy

It is stated that the energy of movement (Activity) is positively correlated with the degree of excitation. Therefore, it can be considered that the movement component corresponding to the Arousal in the PAD parameters is the kinetic energy of the movement. In addition, in a study by Wallbott et al. [4], the kinetic energy of movement (E_k) was found to play an important role in identifying the 14 emotions.

$$E_k = \frac{1}{2}\, \dot{\boldsymbol{q}}^T\, \mathbf{I_M}\, \dot{\boldsymbol{q}} \tag{5.7}$$

For instance, when the kinetic energy was high, the emotion expressed by the movement was perceived as anger, and when it was low, the emotion was perceived as joy or fear (see Fig.5.5). In the Eq.(5.7), $\mathbf{I_M}$ denotes inertial matrix of the manipulator.

b) Jerkiness/Smoothness

It was also stated that Jerkiness is negatively correlated with Pleasure P in case of highly excited emotions [94] [131]. Therefore, it is considered that the lower the value of Jerkiness (smooth movement), the more positive emotion can be expressed. Although the aforementioned kinetic energy can be perceived as both positive and negative emotion depending on its magnitude, it's expected that the addition of Jerkiness information will allow human to distinguish positiveness of the emotion, such as between joy and fear.

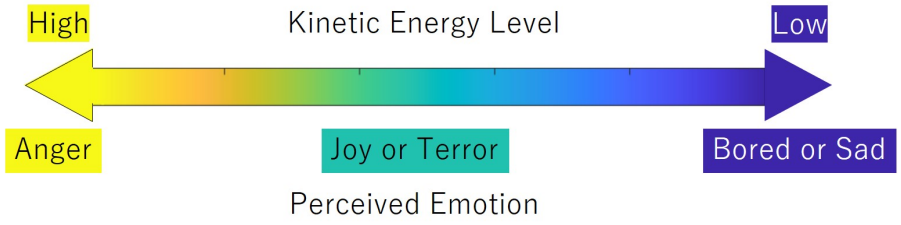

Figure 5.5: Finding of the research by Wallbott et al. [4]. When the kinetic energy was high, the emotion expressed by the movement was perceived as anger, and when it was low, the emotion was perceived as joy or fear

c) Spatial Extent

In the previous research [10], it is mentioned that the direction of eye (gaze) is significantly related to the transmission of dominance D in the PAD parameters. However, since the robot arm in general does not have parts corresponding to the face and eyes, eye movements cannot be introduced into the motion. On the other hand, Wallbot [4] regarded spatial extension of movements as indicator for distinguishing between active and passive emotions. Therefore, we treat the range of spatial use as corresponding to the D parameter in the PAD parameters.

In order to represent spatial extension of the joints' trajectories, the *Geometric Entropy* is used. The measure of geometric entropy provides information on how much the trajectory is spread/dispersed over the available space [132]. Glowinski, et al. [139] showed in his study about affective gestures that the Geometric Entropy Index can be used for comparison of hands trajectorys' spatial extension. Also, it is stated that the Geometric Entropy informs how the available space is explored even in very restrained spaces, including the extreme condition of close trajectory. The joints' trajectories exhibited in the null space frequently fell into such cases. The geometric entropy (H_{ge}) is computed by taking the natural logarithm of two times the length of the pattern traveled by the robot's links (LP) divided by the perimeter of the convex hull around that path (c) as shown in Eq.(5.8). H_{ge} is computed on the frontal plane relative to the participant's point of view. The Geometric Entropy is computed for all links and summed up to represent one emotional motion as whole.

$$H_{ge} = log\frac{2 \times LP}{c} \tag{5.8}$$

d) PAD to $J_rV_eS_p$

Based on the information from the references mentioned above, this study will be communicating a rough image of emotion (anger, joy, fear, boredom) by regulating the energy of movement (velocity), and of making people perceive positivity/negativity and dominace level by using the degree of jerkiness and the level of spatial extent.

Here is the mapping between the points in the PAD space $[PAD]^T \in [-1, 1]^3$ to the points in the $J_rV_eS_p$ space within the normalized domain $[J_rV_eS_p]^T \in [0, 1]^3$ (Eq.(5.9)). J_r, V_e and S_p are motion feature parameters that represents jerkiness, velocity and spatial extent respectively. The motion feature parameter configures amount of corresponding motion feature in the emotional motion. Mappings for jerkiness and spatial extent were defined linearly with reference to Claret's study [10]. On the other hand, the mapping for velocity were defined to reflect the findings presented in Wallbott's study [4] as showin in Fig.5.6. The actual equation of f_v is shown in the appendix.

$$F_{jvs}: \begin{bmatrix} P \\ A \\ D \end{bmatrix} \rightarrow \begin{bmatrix} J_r \\ V_e \\ S_p \end{bmatrix} = \begin{bmatrix} \frac{P-1}{2} \\ f_{v([PAD])} \\ \frac{D+1}{2} \end{bmatrix} \tag{5.9}$$

where

$$f_v = \begin{cases} 1 & PAD_{Hostile} \\ & = [-1, 1, 1] \\ 0.5 & PAD_{Exuberant,Anxious} \\ & = [1, 1, 1], [-1, 1, -1] \\ 0 & PAD_{Bored} \\ & = [-1, -1, -1] \\ [0 \sim 1] & PAD_{other} \end{cases} \tag{5.10}$$

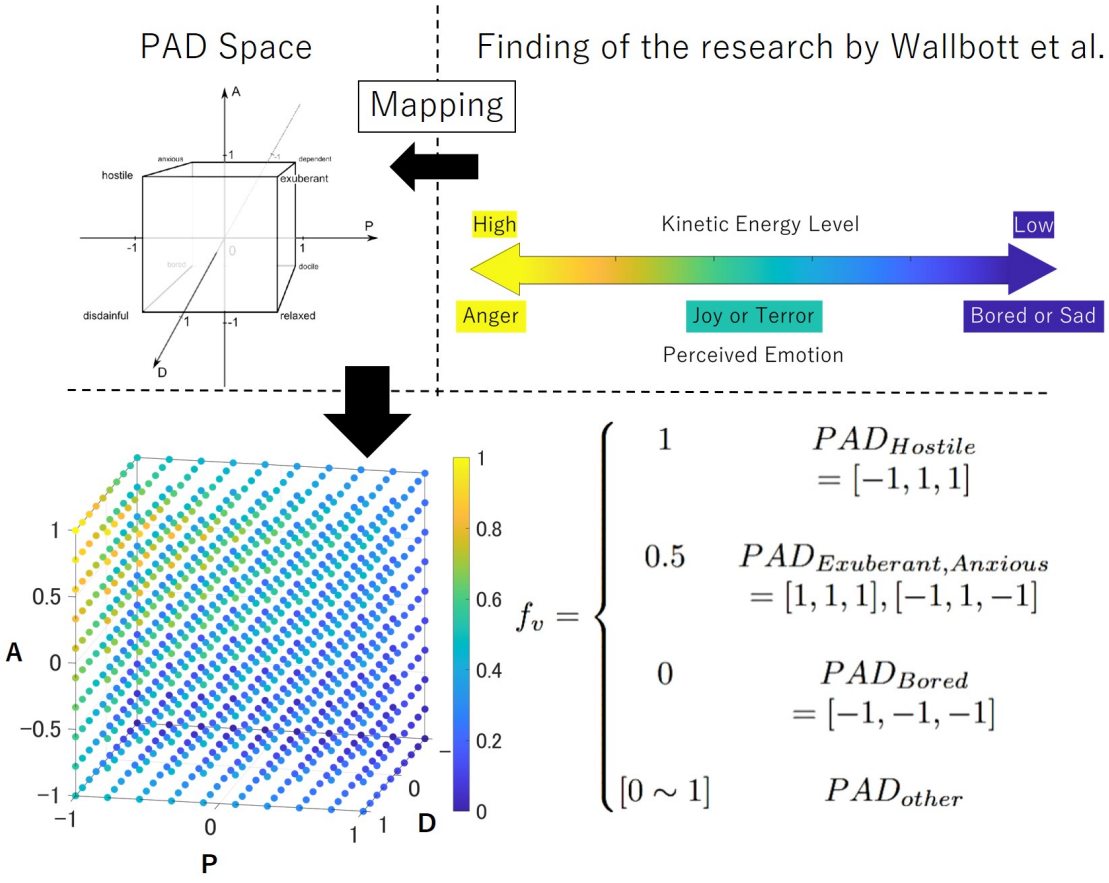

Figure 5.6: Expressed the finding of the research by Wallbott [4] by mapping f_v.

5.3.3 From Motion Features: $J_r V_e S_p$ to Emotional Motion in Null Space

The effective way to define adjustable emotional motion is presented by Claret et al. [10]. In their work, the motion was defined in joint velocity space. In contrast, the emotional motion is defined

in form of the Cartesian space speed (not velocity) $\mathbf{V_{emo}}$ of each i_{th} joint in this study, since this work is focusing on the motion in 3D space.

$$\begin{aligned} {}^{i}\mathrm{V}_{emo} &= F_{emo}(J_r, V_e, S_p) \\ &= A_i \; sin(\omega_i \; t + \phi_i) \end{aligned} \tag{5.11}$$

Where

$$\begin{aligned} A_i &= f_A([J_r, V_e, S_p]) \\ \omega_i &= f_\omega([J_r, V_e, S_p]) \\ \phi_i &= f_\phi([J_r, V_e, S_p]) \end{aligned} \tag{5.12}$$

By adjusting the amplitude A, frequency ω and phase disturbance ϕ, certain motion features such as velocity, jerkiness and spatial extension in Cartesian space can be modified. The A_i is amplitude of the sinusoidal wave where regulate maximum speed of the emotional motion. This coefficient directory related to the kinetic energy of the motion. On the other hand, this coefficient also has a significant impact on the spatial extent as well. This is because, in general, in continuous motion, the greater the velocity in three-dimensional space, the greater the space used by the motion. Therefore, the coefficient that defines the maximum value of this velocity A was designed to be proportional to both V_e and S_p parameters as shown below.

$$A_i = V_e \; S_p \; lim_i \tag{5.13}$$

Where lim_i is the maximum acceptable velocity of i_{th} joint. This limits the motion of the joint in redundant space to the range of $[0, lim_i]$.

Next, the phase disturbance of the sinusoidal wave ϕ affects the jerkiness when it set as below.

$$\phi_i = B_i sin(\omega_b t) + C_i cos(\omega_c t) \tag{5.14}$$

where

$$\begin{aligned} \left.\begin{matrix} B_i \\ C_i \end{matrix}\right\} &= J_r \; \phi_{imax} \\ \omega_b = k_b \; \omega_i, \;\; \omega_c &= k_c \; \omega_i \end{aligned} \tag{5.15}$$

ϕ_{imax} is the maximum design value of the phase shift for i_{th} joint. This limits the phase shift to a maximum of $2\phi_{imax}$. ω_b and ω_c is the frequency of the phase disturbance respectively. These are determined relative to ω_i in Eq.(5.11) using positive value constants k_b, k_c.

Finally, the frequency of the periodic velocity shift in 3D space ω_i has effect on the spatial extent of the motion. The frequency ω_i was defined to be inversely correlated to S_p. In this way, as the S_p parameter increases, the velocity change in three-dimensional space is expected to be slower, and the spatial utilization of the motion larger.

$$\omega_i = -S_p \; k_{Sp} + k_\omega \tag{5.16}$$

Where k_{Sp} and k_ω are constant for parameter S_p and offset (max value) of the frequency respectively. The constant k_{Sp} is set to be the minimum value of the ω_i is more than 0.

From the above equations, a motion $\mathbf{V_{emo}}$ involving features of jerkiness, velocity, spatial extent, specified by the $[J_r, V_e, S_p]^T$ parameter is defined. However, since this is defined in the form of a speed in 3D space, the direction in which this speed is to be achieved must be specified appropriately. For example, if the axis of this speed can always be set in a direction in which the robot can move without affecting the main task, then the designed motion feature could be realized in the null space. On the other hand, if the direction of $\mathbf{V_{emo}}$ is set in a direction in which it is structurally impossible to move, most of the motion feature will be lost when it is projected to the null space. Therefore, this study addresses this issue using manipulability ellipsoid. The manipulability ellipsoid is projected into the null space, and its first principal axis is set as the direction of $\mathbf{V_{emo}}$. This means that the manipulator always moves for emotional expression in the direction in which it can move in the null space. Therefore, the amount of motion feature lost by the null-space projection matrix can be minimized. $\mathbf{V_{emo}}$ was converted to velocity of 3D space emotional motion $\mathbf{X_{emo}}$ as shown in the following section.

5.3.4 Manipulability Ellipsoid in the Null Space

Manipulability ellipsoids are effective tools to perform task space analysis of robotic manipulators. Generally, it is used for the evaluation of the capability at the end-effector in terms of velocities, accelerations and forces. There have been many studies about manipulability ellipsoid. For instance, Yoshikawa introduced the manipulability ellipsoid with consideration of dynamics of the manipulators, which called dynamic manipulability ellipsoid [140]. Also, He present the manipulability ellipsoid for manipulators with redundancy [141]. Chiacchio et al. have focused on singular pose of the manipulator and improved the manipulability ellipsoid for manipulators with redundancy in terms of task-space accelerations [142]. However, those works are about end-effector, and only recently have manipulability ellipsoid in null space in redundant manipulators been studied by Kim et al., in 2021 [143].

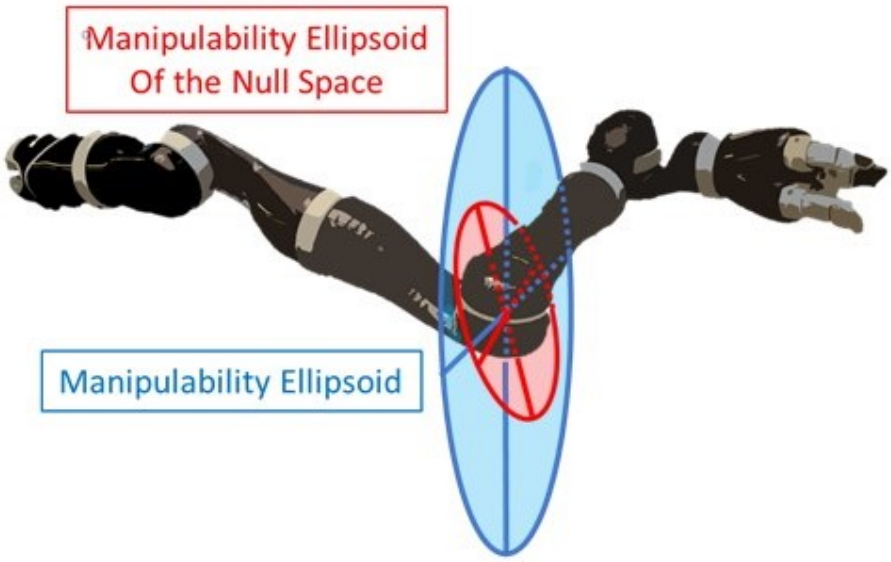

Figure 5.7: Manipulability Ellipsoid for 4th joint. The blue/bigger ellipsoid is for the normal unbounded case. The red/smaller ellipsoid represents the one in the degrees of freedom constrained by the end-effector.

In Kim et al.'s study, it is stated that the primary axis of the manipulability ellipsoid ${}^i\dot{\boldsymbol{X}}_{elp}$ for the i_{th} joint that is constrained by the end-effectors motion can be calculated using null space projection matrixs $\mathbf{N}$ and jacobian $\mathbf{J}$.

$$ {}^{i}\dot{\boldsymbol{X}}_{elp} = \sigma_1 \boldsymbol{U}_1 \tag{5.17} $$

where

$$ \begin{aligned} {}^{i}\mathbf{J}_N &= \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathbf{T}} \\ {}^{i}\mathbf{J}_N &:= \mathbf{J}_i\mathbf{N}_i \end{aligned} \tag{5.18} $$

The null projected Jacobian ${}^{i}\mathbf{J}_N$ contains the relationship between the joint and the task motion in the null space. The null projected Jacobian of the i_{th} link is product of the elements of the $1 - i_{th}$ row and column of the null space projection matrix $\mathbf{N}_i$ and Jacobian of the i_{th} link $\mathbf{J}_i$. The relationship can be obtained from the singular values and vectors by the singular value decomposition (SVD) as Eq.(5.18) shows. Where $\mathbf{U} \in \mathbb{R}^{n\times n}$ and $\mathbf{V} \in \mathbb{R}^{m\times m}$ are orthogonal unitary matrices involving singular vectors of ${}^{i}\mathbf{J}_N$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times m}$ includes singular values, σ_i in the diagonal elements. Thus, the primary axis of the null projected manipulability ellipsoid could be obtained with the first singular value σ_1 and the singular vectors of first row of $\mathbf{U}$ as shown in Eq.(5.17).

Finally, the emotional motion speed for each joints $\mathbf{V_{emo}}$ from Eq.(5.11) was transformed to the velocity along the vector $\dot{\boldsymbol{X}}_{elp}$, and set as emotional motion $\dot{\boldsymbol{X}}_{emo}$ using Eq.(5.20). Then it is converted to the joint space velocity motion $\dot{\boldsymbol{q}}_{emo}$ as below.

$$ \begin{aligned} \dot{\boldsymbol{q}}_{emo}(t) &= \textstyle\sum_{i=1}^{n_j} {}^{i}\dot{\boldsymbol{q}}_{emo}(t) \\ {}^{i}\dot{\boldsymbol{q}}_{emo}(t) &= \mathbf{J}_i^{\#}\, {}^{i}\dot{\boldsymbol{X}}_{emo}(t) \end{aligned} \tag{5.19} $$

where,

$$ {}^{i}\dot{\boldsymbol{X}}_{emo}(t) = {}^{i}V_{emo}(t)\, {}^{i}\dot{\boldsymbol{X}}_{elp} \tag{5.20} $$

Here, n_j represents the total number of joints. Note here that we are adding up the joint displacements to move each joint in the null space. This is because the aim is to allow each joint to move to its maximum extent in its null space. In a redundant robot arm with a serial structure, the joints located in the middle have the largest range of movement in the null space due to its structure. Therefore, when the joint displacements for each joint to move in the null space are added together, the displacement to move the joint located in the middle naturally becomes dominant. Although this effect will be small for a fewer DoF robot arm case, this method can be applied to linearly structured robots with any degrees of freedom, and therefore, the advantage will be significant when applied to a robot arm with larger DoF.

Finally, Eq.(5.6) is used to transfer this emotional motion to the null space. As noted above, the direction of this emotional motion is set as the principal axis of the null space manipulability ellipsoid of each joint, so that maximum expressivity can be expected.

5.4 Implementation

The proposed method is implemented to an upper-half body humanoid "Nextage-Open" shown in Fig.5.8, developed by Kawada Robotics. Nextage-Open is one of the industrial robot with two of 6 DoF manipulators. However, the manipulator can have 7 DoF when the chest joint is included (1st joint in Fig.5.8). The main task at the end-effector is set as moving an object in a horizontal semicircle line. Only the position of the object was specified so that the robot has 4th degree of redundancy (3 rotational + 1 structural). The joint position control was performed with high accuracy by the internal controller. The implementation is done through Python3, and the robot

was operated via Robot Operating System (ROS: Melodic) with Ubuntu environment (Ubuntu 18.04).

The human shaped robot was chosen to conduct survey, so that the participants can naturally perceive emotion. The point of this work is that, it can be applied to any form of single tree structure robot, and can move emotionally without disturbing the main task.

In this implementation, 8 kinds of emotional motions were generated; Hostile, Exuberant, Disdainful, Relaxed, Dependent, Anxious, Bored, and Intermediate. First 7 emotions are located at the vertices of the cube formed in the PAD space in Fig.5.4. On the other hand, the intermediate emotion is the PAD parameter of [0,0,0].

In order to avoid self collision, motor position limit and velocity limit was strictly set for each joints following the regulations based on the ISO/TS 15066 and ISO 10218-1 [144]. Also, the emotional motion was simulated in Gazebo simulator and saved beforehand, then played at the right moment along the scenario even though this method can generate emotional null space motion online along the main task execution. This is because the robot was owned by Kawada Robotics and our team want maximum safety for people and the robot. Also, the other arm (left) that is not conducting main task, was used to express emotion in null space as well, while it keeps safe distance from the other arm. The neck and face was simply tracking the object during the interaction, except the part of the initiation of the interaction and the moment of handing object, in which the robot look at the human.

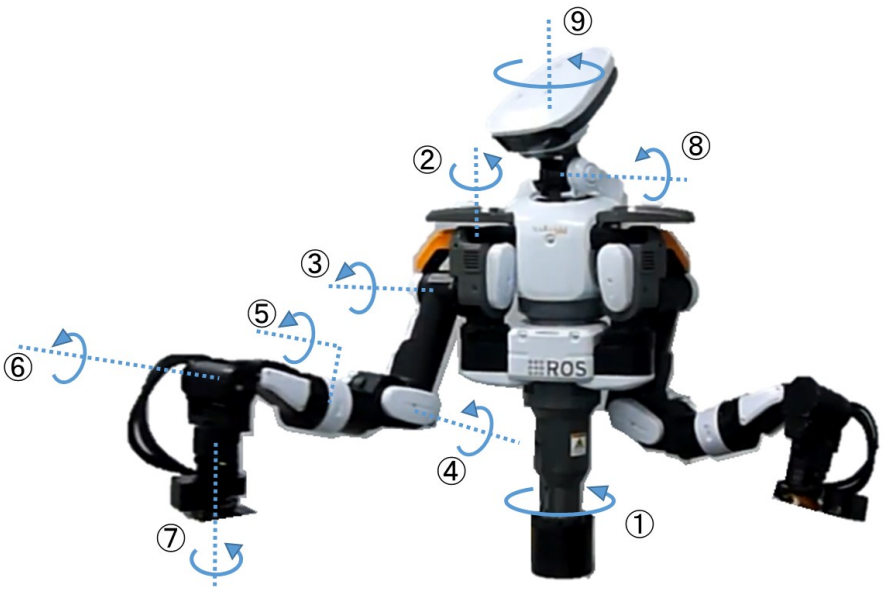

Figure 5.8: Nextage-Open from Kawada Robotics. Its arm has 7 Dof if the chest joint is included. The mass is 36 [kg], Hight is 0.93 [m].

5.4.1 Questionnaire

The questionnaire is compose of three parts with following order; Negative Attitude Towards Robot Scale (NARS), PAD evaluation using SAM scale, and Big 5 personality test. The aim for this questionnaire is to find out;

1. If the robot was able to convey intended emotion to human using null space

2. Is the personality and negative attitude towards robot affects interpretation of the emotional motion

The questionnaire was conducted via internet using Google-Form. The participants are recruited by a recruiting company (Whateverpartners Co., Ltd) via their platform through the internet. The participants received 110 Japanese yen after the questionnaire. The informed consent was acquired at the beginning of the Google-Form by providing the information of the content and background of the research and possible risk for participation. Also, participants were given the chance to freely express their comments on their impressions at the end of the questionnaire. The actual questionnaire is available through the author's Github repository (URL is in the appendix.

a) SAM Scale

The Self-Assessment Manikin (SAM) by Lang [5] was used to evaluate people's interpretation of the robot's emotion. The SAM scale is a visual questionnaire composed of three scales. Each scale is corresponding to a dimension of the PAD space. Lang conducted a comparative analysis of measurement results from both the SAM and Mehrabian and Russell's PAD scales. The correlation coefficients between the two scales were 0.94 for pleasure, 0.94 for arousal, and 0.66 for dominance [95]. The SAM scale can eliminates the effects related to verbal measures because of its visual nature. Also, the participants can write their responses fast and intuitively. The guidelines for a questionnaire with the SAM scale can be found in [145]. The actual SAM scale that was used in this work is shown in Fig.5.9. After observing the robot's movements, the participants asked to evaluate the PAD element level in the movements on a 5-point scale, referring to the SAM scale diagram. Also, after watching each video and estimating the PAD, the respondents were asked to rate a scale of 1 to 5 how confident they were in answering that PAD (1: not confident at all, 5: very much confident). In this research, 8 kinds of emotional motion was generated. The order of the questions are randomized for each participants.

b) Big Five Personality Test

Big Five personality test [146] is a taxonomy of personality traits. The five dimensions are defined as; Openness to experience, Conscientiousness, Extroversion, Agreeableness, and Neuroticism. Although not all researchers agree on the Big Five, it is the most powerful descriptive model of personality and is also well-established basic framework in psychology field [146]. The Big Five has several method to measure; International Personal Item Pool (IPIP) [147], The Ten-Item Personality Inventory (TIPI) [148], Five Item Personality Inventory (FIPI) [148] and etc.

The most frequently used measures of the Big Five compose of items that are self-descriptive sentences [149]. In this work, the simplified version of the Big Five personality traits questionnaire was used in which consists of 15 Japanese questions. This simplified version was based on TIPI-J by Oshio et al. [150]. This test was used in Yue's research [151] to diagnose the personality of humans interacting with robots. Each 3 questions among 15 questions are corresponding to each dimensions of the Big Five taxonomy. In Yue's work, it is showed that, the person who has more openness to experience, tend to think robots to be physically reliable (tough), independent and emotional (Agency). Thus, the personality would indeed affect the perception of the robot. It will be helpful to understand the trend of this human's tendency for designing robot's behavior for in the future.

c) Negative Attitude Toward Robots Scale (NARS)

The Negative Attitude Towards Robot Scale (NARS) was developed to measure human attitudes toward communication robots in daily-life [152]. This scale consists of fourteen questionnaire items in Japanese. These 14 items are classified into three sub-scales, **S1**: "Negative Attitude toward

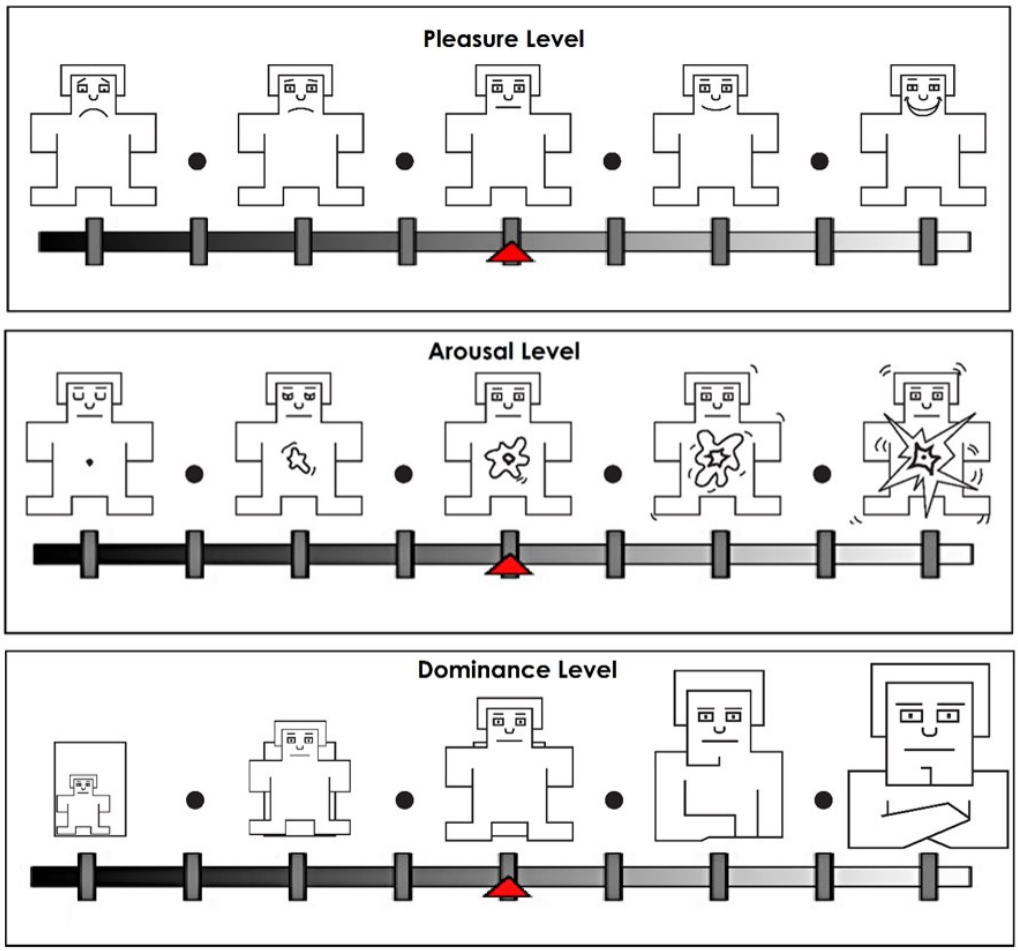

Figure 5.9: The Self-Assessment Manikin (SAM) by Lang [5]. This scale can eliminates the effects related to verbal measures because of its visual nature and help participants to write their responses fast and intuitively.

Situations of Interaction with Robots" (6 items), **S2**: "Negative Attitude toward Social Influence of Robots" (5 items), and **S3**: "Negative Attitude toward Emotions in Interaction with Robots" (3 items). Each question is answered on a five-point scale (1: I strongly dis-agree, 2: I disagree, 3: Undecided, 4: I agree, 5: I strongly agree), and the score of an person at each sub-scale is calculated by summing the scores of all the items in the sub-scale. Thus, the minimum score and maximum score are 6 and 30 in **S1**, 5 and 25 in **S2**, and 3 and 15 in **S3**, respectively.

5.4.2 Experiment Description

The interaction between a person (actor) and the robot is recorded on a camera. During the interaction, the robot performs an emotion transfer behavior designed by the researcher. There are three different scenarios for the video: a positive scenario (PS), a negative scenario (NS), and a positive scenario from frontal view without the human (PSNH) as shown in Fig.5.10. The flow of the interactions are described below.

<Initializing an Interaction >

1. A robot is rearranging objects (Blue Cubes)
2. A person stands in front of the robot
3. The robot notices the person and turns to face at him

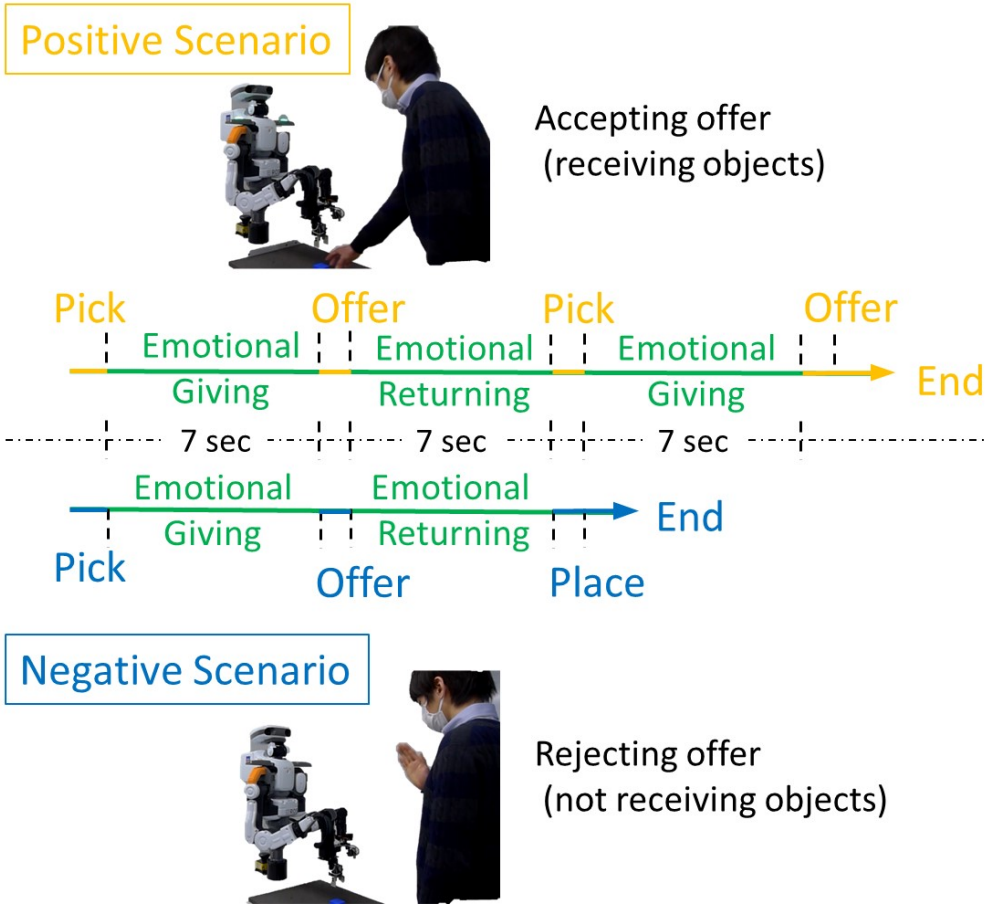

Figure 5.10: There are three different kinds for the video. First is a positive scenario (PS) where a human makes an affirmative gesture (nodding) and accepts the object offered by the robot. Second is a negative scenario (NS) where The human makes a negative gesture and does not accept the object offered by the robot. Finally, a positive scenario from frontal view without the human (PSNH) was prepared.

4. The robot offers the object to him (emotion transfer behavior #1)

<Pattern 1 (Positive Scenario) >

1. The person accepts the object with an affirmative action (nodding)
2. Robot goes to pick up the second object (emotion transfer behavior #2)
3. The robot offers the second object to the person (emotion transfer behavior #3)
4. The person accepts the object with an affirmative action

<Pattern 2 (Negative Scenario) >

1. The person makes a negative action and does not accept the object
2. Robot returns the object to its original place (emotion Transfer Behavior #2)

The main task was specified as time-series 3D space position vector only and not with rotation vector so that the robot has more redundant degree of freedom to express motion in the null space. The duration was 7 second and joint angler velocity and acceleration at start and end point was set to zero. Each emotional behavior is pre-generated and the same emotional expression behavior

is used in each scenario. After the carrying task with emotional motion, the robot was switched to position based control and placed the object using predefined position command without the null space motion which duration is 3 second (see Fig.5.10).

5.5 Result

This section presents the results of the evaluation of the technical aspects of the proposed method and the results of the evaluation related to human cognition. The result of technical aspects shows if the proposed method generates motions that actually contain the desired motion features and if it is able adjust it magnitude. The results of human recognition part presents four results. First, the quality of the emotion conveyance. The second is the correspondence between the motion features (jerkiness, velocity, spatial extent) in the movement and the perceived PAD values. The third result is a comparison between those who estimated the robot's emotions through positive scenarios and those who estimated them through negative scenarios. Finally, the results of the comparison of the PAD values perceived by those with high NARS and those with low scores are presented.

5.5.1 Generated Motions and Their Motion Features

The 3-dimensional trajectory of emotion "Hostile" is shown in Fig.5.11. As the figure shows, $4_{th}, 5_{th}$, and 6_{th} joints are moving dynamically while the end-defector keeps predefined trajectory.

The proposed method is capable of adjusting the degree of Jerkiness, kinetic energy, and spatial extent in the null space, respectively. In order to confirm this, the correspondence between the motion feature parameters; J_r, V_e, S_p and the amount of the desired motion features index in the actually generated movement is presented in Fig.5.12. The dotted lines are lines approximated by the least-squares method. In all these cases, there was a positive correlation between the control parameters of the motion feature; J_r, V_e, S_p and the amount of the target movement elements in the movements produced by them. In each figure, the two motion feature parameters other than the desired parameter are varied between $[0, 1]$. For example, in the right figure of Fig.5.12, which is plot about J_r parameter, the plots shows the parameters V_r and S_p varied from 0, 0.5, and 1, respectively.

Although, originally simulated motions ,which are plotted in blue color, show variety of motion feature amount, the actually implemented motions, which are plotted in orange color, show limited variance in the motion features. Especially, in case of jerkiness, it does not shows wide rage of jerkiness even though J_r parameter varies. The discussion about this will follow in the subsequent Chapter 5.6.

5.5.2 Emotion Conveyance Results

Secondly, results of the human perception of the robot's emotions are presented. The Table 5.1 shows the ratings of the robot's emotions by the participants who viewed the aforementioned emotional expression videos in Section 5.4.2. The age distribution of the participants was; 20's: 43 (male: 14, female: 29), 30's: 106 (male:43, female: 63) and 40's: 118 (male: 64, female: 54), total is 267. They are all have Japanese cultural background.

There are three different scenarios for the video: a positive scenario (PS), a negative scenario (NS), and a positive scenario from frontal view without the human (PSNH). Of the 267 respondents, 70 viewed and answered to PS, 87 viewed and answered to NS, and 110 answered to PSNH.

The values of the PAD parameters representing each emotion are shown besides the emotion. In the "MEAN" column, those highlighted in bold are those for which the majority of participants guessed the PAD value in the correct direction. "CONF MEAN" is the mean of the confidence values; after watching each video and estimating the PAD, the respondents were asked to rate a scale of 1 to 5 how confident they were in answering that PAD (1: not confident at all, 5: very much confident). In the row of "Bored" and "Docile", " x" represents any value between 0 and

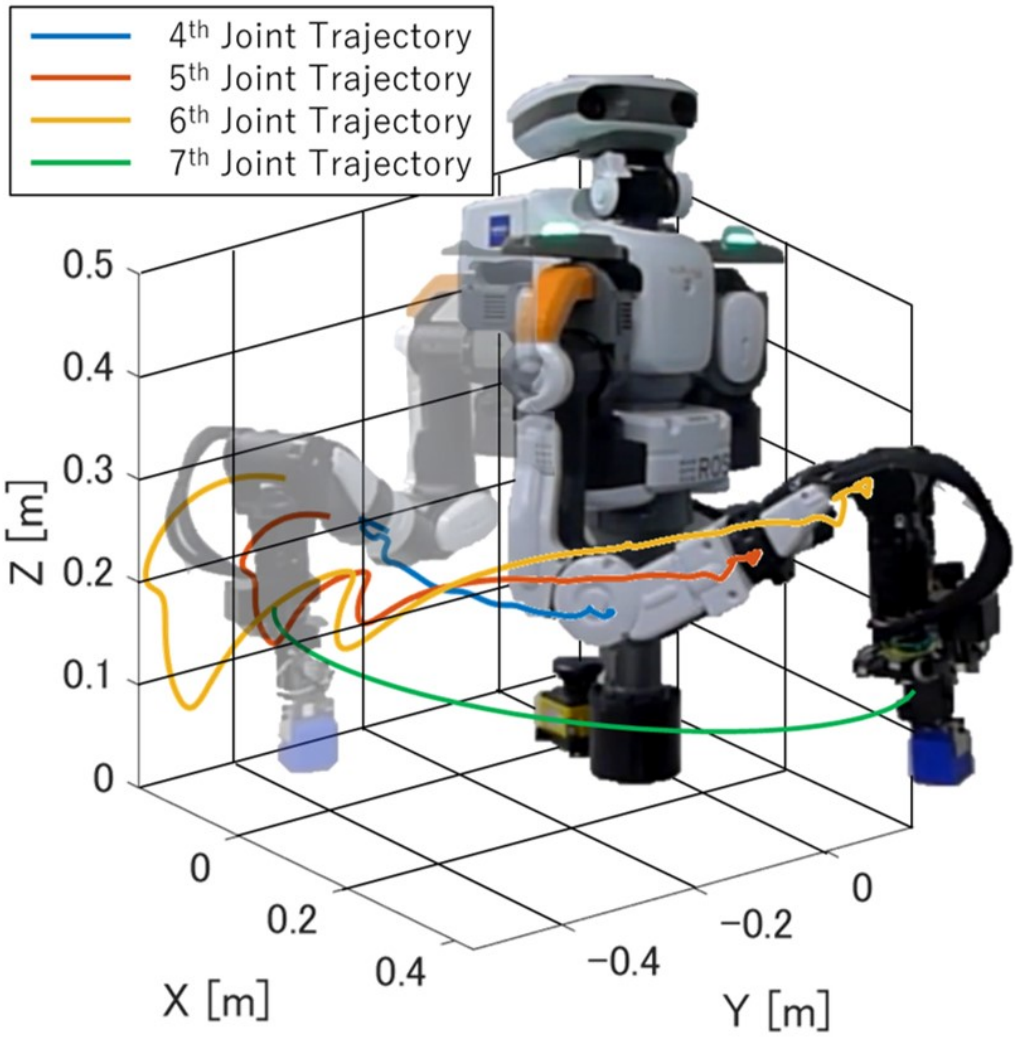

Figure 5.11: The 3-dimensional trajectory of emotion "Hostile". As the figure shows, 4th, 5th, and 6th joints are moving dynamically while the end-defector keeps predefined semicircle trajectory.

1. The reason why they share same row is that the motion feature parameters V_e and S_p of the motion assigned to these two emotions are both 0, which leads to zero amplitude of the emotional motion in Eq.(5.11). This means that for these emotions, the null space motion will be no motion at all.

The following is a summary of the comments that were voluntarily answered at the end of the survey. A total of 113 comments were received. Of these, 38 mentioned that it was difficult to read robot's emotions. Conversely, 17 expressed that the robot was emotionally expressive. Of the participants who commented that it was difficult to read emotions, 8 people commented that they could not read emotions due to the lack of facial expression. 41 people left positive comments about the experiment, which were hopeful about the future of robots and humans. On the other hand, 7 people commented on their fear of robots and their desire for thoughtful development of intelligent robot.

Here is the relationship between the magnitude of the motion features in each emotional movement and the perceived PAD values 5.13. Each plot represents each emotional motion, and the dotted lines are lines approximated by the least-squares method. While kinetic energy and perceived arousal level have positive correlation as other researches suggest, other two motion feature

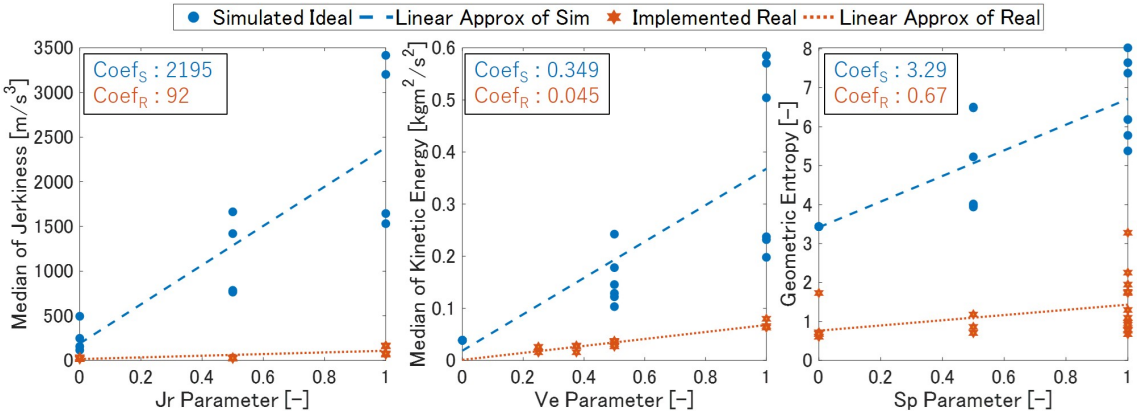

Figure 5.12: Motion features of each generated motions. The proposed method is capable of adjusting the degree of Jerkiness (left), kinetic energy (center), and spatial extent(right) in the null space, respectively. Those plotted in circles are the motions generated in the simulation environment (blue), and those plotted with star marks are the ones actually implemented to the real robot (orange).

didn't vary much and correlation was not that obvious from those results.

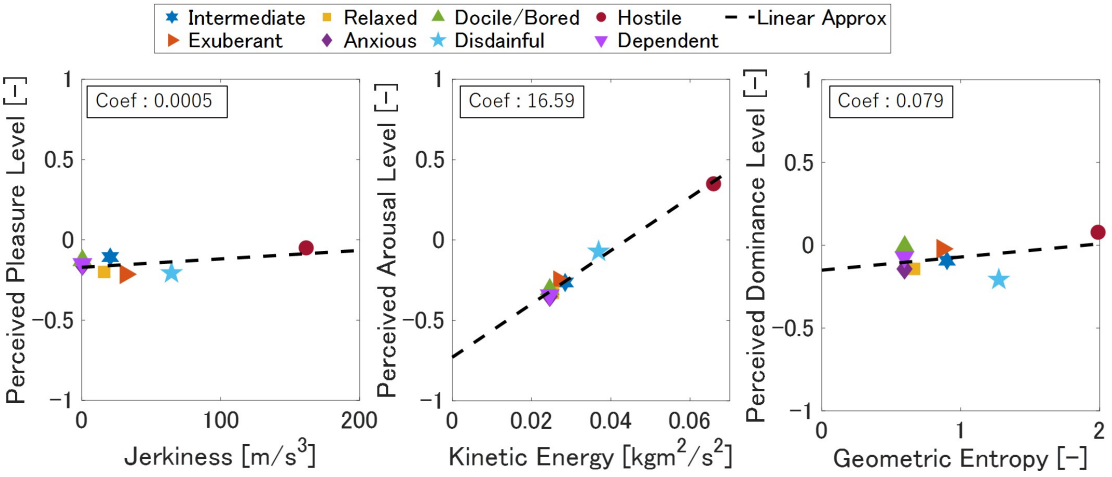

Figure 5.13: Motion features and perceived emotions.Each plot represents a different emotions, and the dotted line is the approximate line by the least squares method. Each figure is for Jerkiness (left), Kinetic Energy (center), and Geometric Entropy (right). "Hostile" was the only one with a distinctive motion features, while the other emotions were similar

"Hostile" have distinct value in terms of the motion feature as it intended. On the other hand, the movements other than Hostile were almost identical in terms of movement characteristics. This must be the reason why the emotion conveyance was not proper as seen in the Table 5.1.

Table 5.2 compares the responses of those who estimated PAD by observing a positive scenario with those who responded by viewing a negative scenario. p-values are obtained using the "Mann-Whitney U Test". Significant differences ($p < 0.05$) are highlighted in bold.

Table 5.2: Comparison between the group who estimated PAD by observing a positive scenario, and who with the negative scenario

	Positive - Negative		
Emotion	p_P	p_A	p_D
Hostile	0.18	0.18	**0.04**
Exuberant	0.30	0.31	**0.01**
Dependent	**0.00**	**0.05**	**0.01**
Anxious	**0.00**	0.28	**0.01**
Disdainful	0.37	0.34	0.17
Relaxed	**0.05**	0.36	0.24
Bored/Docile	**0.00**	0.30	**0.00**
Intermediate	**0.00**	**0.00**	**0.00**

This evaluation was done by right-tailed evaluation. To be clarify, Let F(u) and G(u) be the cumulative distribution functions of the distributions underlying x; an answer from PS and y; an answer from NS, respectively. In this case, the distribution underlying x is stochastically grater than the distribution underlying y, i.e. $F(u) > G(u)$ for all u. Thus, positive scenarios tended to be perceived with higher Pleasure and Dominance than negative ones.

Finally, the Table 5.3 compares the answers of robot emotion (PAD) estimation tendencies of those with high and low NARS scores. Significant differences ($p < 0.1$) are highlighted in bold.

Table 5.3: Comparison between the group with high and low NARS scores

	NARS High vs Low		
Emotion	p_P	p_A	p_D
Bore/Docile	**0.01**	**0.00**	**0.07**
Anxious	0.55	0.19	0.87
Dependent	**0.05**	**0.00**	**0.06**
Disdainful	0.12	0.35	0.30
Relaxed	**0.01**	0.12	**0.01**
Intermediate	**0.01**	**0.01**	**0.04**
Exuberant	**0.04**	**0.01**	0.38
Hostile	**0.06**	0.23	0.18

This evaluation was done by left-tailed evaluation. To be clarify, Let F(u) and G(u) be the cumulative distribution functions of the distributions underlying x; an answer from high NARS people and y; an answer from low NARS people, respectively. In this case, the distribution underlying x is stochastically less than the distribution underlying y, i.e. $F(u) < G(u)$ for all u. Thus, People who has less negative attitude towards robots tended to be perceived robot emotion with higher Pleasure, Arousal and Dominance.

It was difficult to draw clear-cut conclusions about the high and low Big5 personality factors and the perception of robot's emotion. This is because the observed correlations were limited because robot's emotional expressions were not so different to each other as mentioned in Fig.5.13.

5.6 Discussion

Prior research has shown that motion features such as kinetic energy, jerkiness, and spatial extent affect human's perception of emotions. Our approach seeks to change these elements through modulating the emotional parameters (PAD). The perceived emotion and the intensity of each movement element are in reference to previous research as described in Section 5.3.2.

The movements executed in the simulation environment were highly variable, whereas those were implemented in the actual robot were relatively unvarying. The gap between the implementation and the simulation was the due to safety concerns.

The Nextage-Open is capable of conducting industrial operations with high control accuracy. Therefore, it is equipped with heavy/powerful motors. The actor's safety was our primary concern, so in the experiment, we limited the range of movements and joint velocity following the regulations based on the ISO/TS 15066 and ISO 10218-1 [144]. In addition, the joint angular jerkiness was limited to prevent any damage to the robot system.

According to ISO/TS 15066, contact between a robot and a human is classified into two types, transient contact and quasi-static contact. Transient contact is defined as physical contact between the robot and human that lasts less than 0.5 seconds, i.e., a collision. Quasi-static contact is defined as contact that lasts longer than 0.5 seconds and is classified as a "pinching" accident.

ISO/TS 15066 also describes the maximum allowable pressure, which indicates how much force the human body can take before becoming injured. For example, in quasi-static contact, pressure of up to 150 N on the neck, 210 N for the back, and 140 N for the fingers are permissible. However, these numbers are simply indicators of when injury occurs, so they cannot be used as safety measures for motion generation. To avoid extreme cases, ISO/TS 15066 provides safety design measures such as speed limits to forces and torques. Table **??**, which is from the same ISO document, provides a more realistic guide in ensuring the safety of cooperative robots.

The requirements specified in ISO/TS 15066 and ISO 10218-1 state that the moving speed of an industrial robot that can be used as a cooperative robot must not exceed 250 mm/s. When applied to the robot used in our study (torso-right arm: 12 kg), the maximum kinetic energy that can be exerted is roughly 0.375 $[kgm^2/s^2]$. Therefore, an industrial robot is required to express emotional behavior in the range of 0 - 0.375 $[kgm^2/s^2]$. Considering that the maximum value of kinetic energy in the implemented motion was 0.72 $[kgm^2/s^2]$, it is clear that our method implements almost the maximum value of motion that can be adapted safely. The presented results show that emotional movements in the range of 0 - 0.375 $[kgm^2/s^2]$ are difficult for people to discriminate. In other words, current industrial robots must exaggerate their movements to a dangerous in order to express distinct emotions in a comprehensible manner. Thus, collision avoidance techniques and soft robots are needed to deal with this problem. In order for robots to move freely in human society, it is necessary to take human safety into consideration and abide by relevant laws. In light of this, soft robots and robots made of light materials will have an advantage in entering human society.

Another reason that participants were unable to distinguish between emotional motions may be that the conversion from PAD parameters to $J_r V_e S_p$ parameters was not appropriate. In particular, there is room for improvement in the conversion of the V_e parameter, which is responsible for the amplitude of the motion in the null space. As shown in Fig.5.13, only "Hostile" had a noticeable change in the movement component. This is due to the fact that when converting PAD parameters to V_e through f_v in Eq.(5.9), the V_e parameters were set to take a range of 0 to 0.5 except the case of "Hostile". The V_e parameters were not scattered between 0 and 1 but clustered between 0 to 0.5, resulting in similar movements of the elements of the behavior except the case of "Hostile".

As shown in Table 5.1, for all three scenarios, the majority correctly estimated the "Arousal" level of "Bored," "Hostile," and "Relaxed." However, most of items in the table have a standard deviation of more than 0.4, which suggest that participants had difficulty in discerning emotions.

Between the responses when viewing the robot-only video (PSNH) and when viewing the HRI video (PS, NS), participants were relatively more confident in their estimation of emotion when viewing the HRI. This suggests that the context of the interaction greatly affected the recognition of emotion in the HRI, especially if it was difficult to find differences in emotional behavior.

As shown in Table 5.2, the positive scenarios tended to be perceived with higher Pleasure and Dominance than the negative ones. The participants who observed the NS provided the following comments referring to the context of the interaction:

"I felt a little sorry for the robots that always got turned down."

"It was easy to get a negative impression because I felt sad that the robot's offer was not accepted."

"The way the robot looked at people when they approached, and the way it seemed to be disappointed after being rejected were very human-like."

Besides the context of the interactions, prejudice toward robots also altered the emotional readings. Table 5.3 shows that those with high NARS were more likely to interpret the robot emotion PAD as low.

These results suggest that emotion judgments are made based on the emotion expression behavior, the content of the interaction (story), and the observer's impression of the robot. In this case, because it was difficult to read emotions from the robot's movements, the observers either rationalized the emotions from the scenario or perceived low negative emotions by unconsciously projecting their own prejudices toward the robot.

There were eight comments that referred to the robot's facial expressions and facial movements, which were not used as emotional expression movements at this time. Those comments indicated that facial expressions and gaze (facial orientation) play an important role in conveying emotional information for humanoid robots.

These elements are complementary, so that if one is missing, the others compensate for it, and then people try make sense of the emotion.

5.7 Conclusion

Sharing emotion between human and robot is one of the most important elements of human-robot interaction (HRI). Through sharing emotion, people may find HRI socially variable and robots would be able to work at full capacity without aversion/disturbance from people.

We introduced a method of conveying emotions through a robot arm while it simultaneously executes other tasks. Our method utilizes a null space control scheme to exploit the kinematic redundancy of a robot manipulator. The null space control is for adding sub-tasks (additional motion) to the robot without affecting the task at the end-effector. We have also presented an effective motion generating method that combines manipulability ellipsoid and null space control. This method enable the robot to continue to move within the redundant degrees of freedom. Prior research has shown that motion features such as kinetic energy, jerkiness, and spatial extent affect humans' perception of emotion. Thus, our approach seeks to adjust these features through modulating the emotional parameters (PAD).

The Nextage-Open was used to implement the proposed method, and the HRI was recorded on video. Eight different emotions (Bored, Anxious, Dependent, Disdainful, Relaxed, Hostile, Exuberant, and Intermediate) were expressed for each interaction. Then a questionnaire containing the Self-Assessment Manikin (SAM) scale, Negative Attitude Towards Robot Scale (NARS), and Big Five Personality Test was conducted via the internet using Google Forms.

Responses were collected from 267 people, and the results suggested that even when industrial machines perform emotional behaviors within the safety standards set by ISO/TS 15066, it is difficult to provide enough variety of motion for each emotion to be distinguished. However, the analysis of the generated motions in simulation showed that the proposed method can adjust the important motion features through modulating the motion feature parameters $J_r V_e S_p$.

Furthermore, the results suggest that emotion judgments are made based on the influence of the emotion expressing behavior, the context of the interaction (scenario), the observer's impression of the robot, and gaze/facial expression, and these factors compensate for each other. In our study, because the robot's emotional motion was difficult to distinguish, the observers either rationalized the emotions from the scenario or perceived low PAD emotions (negative, less active and passive) by unconsciously projecting their own prejudices toward the robot.

Finally, if a robot unilaterally expresses emotions while ignoring the person's emotional state, there will be no empathy between the two. A system that observes the person's reactions and express emotions accordingly (human-in-loop) is also necessary and will be our next research direction. Moreover, because some robotic arms do not have practical joint position limits, (e.g., the Kinova Jaco-arm), another interesting direction for this research would be to explore the possibility of expressing emotions through non-human like movements.

5.8 Publications

The following list contains the published and submitted papers regarding this doctoral work.

Journals

- S. Hagane, G. Venture, Robotic Manipulator's Expressive Movements Control Using Kinematic Redundancy, *Machines*, 10(12), 1118, October 2022

Table 5.1: Results of the emotion conveyance questionnaire

(a) Positive Scenario (PS)		MEAN			STD			CONF MEAN
Emotion	[P, A, D]	μ_P	μ_A	μ_D	δ_P	δ_A	δ_D	μ_C
Intermediate	[0, 0, 0]	**-0.11**	-0.27	-0.09	0.40	0.39	0.51	3.07
Exuberant	[1, 1, 1]	-0.21	-0.25	-0.02	0.47	0.41	0.49	3.24
Relaxed	[1, -1, 1]	-0.20	**-0.33**	-0.14	0.43	0.42	0.43	3.11
Anxious	[-1, 1, -1]	-0.17	-0.36	**-0.16**	0.44	0.42	0.54	3.03
Bore/Docile	[x, -1, -1]	-0.14	**-0.31**	-0.01	0.50	0.43	0.53	2.99
Disdainful	[-1, -1, 1]	**-0.22**	-0.08	-0.22	0.53	0.49	0.56	3.27
Hostile	[-1, 1, 1]	-0.06	**0.35**	0.06	0.67	0.53	0.55	3.84
Dependent	[1, 1, -1]	-0.16	-0.37	-0.09	0.41	0.44	0.50	3.07

(b) Negative Scenario (NS)		MEAN			STD			CONF MEAN
Emotion	[P, A, D]	μ_P	μ_A	μ_D	δ_P	δ_A	δ_D	μ_C
Intermediate	[0, 0, 0]	-0.41	-0.45	-0.33	0.43	0.41	0.46	2.92
Exuberant	[1, 1, 1]	-0.24	-0.28	-0.21	0.42	0.42	0.49	3.30
Relaxed	[1, -1, 1]	-0.30	**-0.36**	-0.20	0.39	0.44	0.55	3.08
Anxious	[-1, 1, -1]	**-0.46**	-0.41	**-0.39**	0.35	0.45	0.43	2.92
Bore/Docile	[x, -1, -1]	-0.40	**-0.35**	**-0.29**	0.39	0.39	0.41	3.02
Disdainful	[-1, -1, 1]	-0.20	-0.03	-0.14	0.47	0.47	0.51	3.26
Hostile	[-1, 1, 1]	-0.17	**0.29**	-0.09	0.48	0.49	0.50	3.28
Dependent	[1, 1, -1]	-0.39	-0.48	**-0.26**	0.42	0.38	0.44	2.94

(c) PS Without Human (PSNH)		MEAN			STD			CONF MEAN
Emotion	[P, A, D]	μ_P	μ_A	μ_D	δ_P	δ_A	δ_D	μ_C
Intermediate	[0, 0, 0]	-0.09	-0.35	-0.11	0.39	0.42	0.51	2.75
Exuberant	[1, 1, 1]	-0.09	-0.28	-0.05	0.45	0.43	0.49	2.95
Relaxed	[1, -1, 1]	-0.15	**-0.39**	-0.11	0.45	0.46	0.51	2.80
Anxious	[-1, 1, -1]	-0.13	-0.33	-0.09	0.38	0.43	0.50	2.73
Bore/Docile	[x, -1, -1]	-0.08	**-0.28**	0.04	0.41	0.50	0.53	2.77
Disdainful	[-1, -1, 1]	-0.15	-0.04	-0.21	0.56	0.55	0.56	3.37
Hostile	[-1, 1, 1]	-0.08	**0.30**	-0.10	0.66	0.61	0.56	3.75
Dependent	[1, 1, -1]	-0.07	-0.30	-0.01	0.42	0.49	0.50	2.68

"CONF MEAN" is the mean of the confidence values: how confident they were in answering that PAD (1: not confident at all, 5: very much confident). In the "MEAN" column, those highlighted in bold are those for which the majority of participants guessed the PAD value in the correct direction.

(a) 70 answers are acquired in total for PS. (b) 87 answers are acquired in total for NS. (c) 110 answers are acquired in total for PSNH.

Chapter 6

Conclusion

SUMMARY resents the conclusions of the thesis, outlining the main contributions and directions for further research regarding limitations of presented works. Also, published work during the doctoral course is presented.

6.1 Summary of the Contributions

The main general contributions of this thesis are the following:

- The development of the AGPC system that adapt to the new reference model based on the fast dynamic and geometric identification with optimal trajectories and capable of precise motion and force control.
- presents the successful design of a Generalized Predictive Control/Controller that is able to control robot's joint position and velocity stably under a wide range of posture/velocity/acceleration for a 3-DoF robot arm using linearization method by Koopman Operator.
- Developed an emotion expression method that exploits manipulability ellipsoid for expressing emotional motion in the null space effectively while executing a main task at the end-effector.

Regarding the Adaptive Generalized Predictive Control/Controller and Cartesian force control for robot arm using dynamics and geometric identification, which corresponds to the Chapter 3 of this thesis, the particular contributions are:

- Presented the successful application of GPC to the robot manipulator, in which result showed that GPC can be stabilized even with local and simple lionization when the control parameter is adapted according to the stability index.

In this study, integration of the fast dynamic and geometric identification of robotic manipulator and Model Predictive Control/Controller method namely GPC was presented. The system is named AGPC and it compose of the tuning system of AGPC's control parameters for the optimal predictor, with respect to the dynamic and geometric modeling of the robot. The proposed methods were applied to the KUKA lightweight robot. During the roller painting task, the external forces applied to the end-effector could be estimated and controlled without a force sensor mounted at the end-effector, due to the accurate model identification. Moreover, the controllers were developed using the identification performed with the exciting motion of 10 s, which was generated by an optimization. The proposed adaptive predictive controllers were compared with other controllers such as the PD control with gravitational compensation, feedforward with PD control, and GPC. The AGPC updated itself during the motion and successfully controlled the robot with good stability by exploiting the fine identified SIPs. The GPC and AGPC demonstrated excellent control performances such as a 90% reduction in the position errors when compared with

the results of the previous experiment. Moreover, a novel tuning design with robust control will be developed in the future to enhance the AGPC. Furthermore, the future work will extend this method to a significantly larger number variables, rollers, and tools.

Regarding the linearizing robotic manipulator's dynamics using koopman operator and applying generalized predictive control, which corresponds to the Chapter 4 of this thesis, the particular contributions are:

- Presents working a candidate for new state vector (observables) for 3 DoF serial link manipulators. The optimal observables choice was acquired by expanding the robot dynamics equation and using its monorails. The KP was then identified by the least-squares minimization, and an approximate linear state-space model of the 3 DoF simplified Panda robot was identified. Using this, GPC was designed, and it was compared with PD controller. Fig.4.3 shows that our observables choice was properly describing the non-linear behavior globally and made the linear state space model as accurate as previous research. Using these linear predictors, GPC was designed and its performance in the task space and joint space was shown in Fig.4.4 and Fig.4.5. The results depict GPCs have more adaptability to the different motions velocities than PD control.

- Presents a method which exploit Singular Value Decomposition (SVD) of the KP to obtain proper observables for identification of better KP.

- presents advantage of choosing a large size observable to identify KP which that the KP identified for a larger size of observables contains more information about the system. The results show that the information of the mass of the end-effector is reflected in the operator's value. This suggests that information-rich KP can be analyzed using machine learning techniques such as PCA or auto-encoders to estimate the mass and even the shape of the object at the tip of the hand.

Chapter 4 contributes to one of the few instances in which the KP has been introduced into the serial link robot arm system. This study presents an example of the linearization of a robot manipulator's dynamics using KP, and presents practical issues to be considered when applying the principle of KP theory to robotic systems which is the question of how to embed the new observables to get sufficient linearized model. Also, it introduced the advantages of setting up large-size observables. Using the KP, this work presents the successful design of a Generalized Predictive Controller (GPC) that is able to control robot position and velocity stably under a wide range of posture/velocity/acceleration for a 3-Dof robot arm. Control results show that our GPC performance outperforms a traditional PD controller in terms of adaptability to the diverse target trajectories.

Regarding the robotic manipulator expressing emotions as sub-task using kinematic redundancy, which corresponds to the Chapter 5 of this thesis, the particular contributions are:

- Developed the motion generation system for robotic manipulator that adjust motion feature in the redundant degree of freedom (a.k.a null space). The adjusted motion features are effectively executed in the null space by using the concept of manipulability ellipsoid. The result showed that the proposed method is capable of adjusting the degree of Jerkiness (left), kinetic energy (center), and spatial extent(right) in the null space.

- Presents actual examples of emotional expression within the speed limit as stated in the guidelines for industrial robots performing cooperative tasks (ISO/TS 15066 [144]). The result showed that current industrial robots must exaggerate their movements to a dangerous in order to express distinct emotions in a comprehensible manner.

- Presents the results suggest that emotion judgments are made based on the influence of the emotion expressing behavior, the context of the interaction (scenario), the observer's impression of the robot, and gaze/facial expression, and these factors compensate for each other in case of emotion is unclear from movements.

The chpter 5 introduces an emotion expression method that can coexist with any main task at end-effector. This method exploits manipulability ellipsoid for expressing emotional motion in the null space effectively. By adjusting the velocity of its null space motion, jerkiness, and spatial extent, this method enables any manipulators to show its mood while executing the main task at its end-effector. The Nextage-Open was used to implement the proposed method, and the HRI was recorded on video. Eight different emotions (Bored, Anxious, Dependent, Disdainful, Relaxed, Hostile, Exuberant, and Intermediate) were expressed for each interaction. Then a questionnaire containing the Self-Assessment Manikin (SAM) scale, Negative Attitude Towards Robot Scale (NARS), and Big Five Personality Test was conducted via the internet using Google Forms. Responses were collected from 267 people, and the results suggested that even when industrial machines perform emotional behaviors within the safety standards set by ISO/TS 15066, it is difficult to provide enough variety of motion for each emotion to be distinguished. However, the analysis of the generated motions in simulation showed that the proposed method can adjust the important motion features through modulating the motion feature parameters $J_r V_e S_p$. Furthermore, the results suggest that emotion judgments are made based on the influence of the emotion expressing behavior, the context of the interaction (scenario), the observer's impression of the robot, and gaze/facial expression, and these factors compensate for each other. In our study, because the robot's emotional motion was difficult to distinguish, the observers either rationalized the emotions from the scenario or perceived low PAD emotions (negative, less active and passive) by unconsciously projecting their own prejudices toward the robot.

6.2 Limitations and Future Work

During the realization of this doctoral work, several limitations of the proposed method became clear and interesting directions to follow in future research have emerged.

About AGPC, the computation intensity is the biggest limitation. The control sampling frequency will be higher when the computation process is parallelized. In the AGPC system, the computations of inertial matrix and optimization of the GPC control parameter are biggest source of the computation time consumption. The simple ways to parallelize those process are using multiple processor and optimizing scripts. Moreover, applicable system for this AGPC method is limited to the legit body structure and require joint torque and acceleration information for DPI. However, in the future, the structure of robot will be more diverse and acquiring joint sensory information will be not simple. For example, soft robot that doesn't have joints and bio-inspired joint with complex actuator is not easy for this method to adapt.

Here, data driven dynamical system analysis such as KP is interesting for DPI application. Data driven approach doesn't require symbolic modeling about robot structure but use only history of motion data (e.g. time-series control input and corresponding output of controlled system) and generate model that represents a relationship between those data. Although, the acquired model often doesn't show system's dynamics intuitively and it seems difficult to extract variable numbers such as SIP, machine learning approach will be a great tool to obtain dynamics data from the model. Even though, by skipping detailed symbolic structural modeling, there is a possibility of insufficient modeling result, recent rise of computation power will cover this problem and it is open question to us how to obtain accurate model and SIP from motion data only.

In this thesis, robotic manipulator's dynamics was identified as state space model using KP. And it was utilized to design GPC. However, the performance of GPC was not as good as Computed Torque Control/Controller (CTC) with accurate dynamical and kinematic model due to the model error. This model error is from the choice of observables that was set when KP was identified and quality of data that is used to identify KP. If the motion data was biased to a certain posture and acceleration was not enough, the non-linearity of robot's dynamics will be not modeled properly. In another word, how to generate motion trajectory that satisfied enough acceleration and variation of posture is important question. Moreover, the cost function for evaluating the quality of the trajectory for KP is interesting research direction.

About the optimisation of choice of observables, it is also open question that ought to be answered. It should be possible that one can design specialized observables and identify KP to detect dynamical parameter change of robot. In this thesis, SVD was used to KP to optimize the observables. On the other hand, author believe that it is also interesting to search for elements of the Koopman operator matrix that change in response to changes in the end-effector mass, in order to determine the optimal observables. Furthermore, if the observables are optimal and the motion trajectory is good quality, it might possible to detect not only the mass information of the end-effector but also the shape of the end-effector (tool).

ultimately, MPC that occasionally update its reference model automatically to deal with dynamic parameter change and realize fast and accurate motion is possible. it is author's hope to control soft/light robot with high precision using MPC so that finally robot can work close to the people safely and advance understanding how to deal with robot and better use of robot. The reason why soft/light robot is author's interest is that the heavy legit robots are still difficult to be operated close to human. As mentioned in chapter 5, the speed limit of the end-effector of cooperative robot is quite slow as it regulated in ISO. However, emotional motions require more speed range to be perceived as different emotions.

Besides the social constrain of the robot, emotional motion generator still need to be adjust its mapping between emotion (PAD) to motion feature (JVS). Especially, the mapping of PAD to velocity parameter is interesting topic since the velocity of motion affect significantly how the motion is perceive by people. Also, it will be interesting to introduce this method to non-human shape robot. In the presented application experiment of the system in chapter 5, the emotional motion was constrained within human like posture due to motor position limit. However, there are many robot with non-human shaped and some robot doesn't have joint position limit so it can realize non-human-like motion. It is interesting to see how people will perceive these motions.

Finally, if a robot unilaterally expresses emotions while ignoring the person's emotional state, there will be no empathy between the two. A system that observes the person's reactions and express emotions accordingly (human-in-loop) is also necessary and will be our next research direction.

6.3 Publications

The following list contains the published and submitted papers generated during the course of this doctoral work.

Journals

- S. Hagane, L.K. Rincon-Ardila, T. Katsumata, V. Bonnet, P. Fraisse, G. Venture, Adaptive Generalized Predictive Controller and Cartesian Force Control for Robot Arm Using Dynamics and Geometric Identification, *Journal of Robotics and Mechatronics*, volume 30(6), pp.927-942, December 2018, DOI: 10.20965/jrm.2018.p0927, *Associated to Chapter 3

- S. Hagane, G. Venture, Robotic Manipulator's Expressive Movements Control Using Kinematic Redundancy, *Machines*, 10(12), 1118, October 2022, DOI: 10.3390/machines10121118, *Associated to Chapter 5

- S. Capy, P. Osorio, S. Hagane, C. Aznar, D. Garcin, E. Coronado, D. Deuff, I. Ocnarescu, I. Milleville, and G. Venture, Yōkobo: A Robot to Strengthen Links Amongst Users with Non-Verbal Behaviours, *Machines*, volume 10(8), number 708, August 2022, DOI: 10.3390/machines10080708, *Associated to doctoral dissertation of Siméon Capy

- S. Capy, L.K. Rincon-Ardila, E. Coronado, S. Hagane, S. Yamaguchi, V. Leve, Y. Kawasumi, Y. Kudou, G. Venture, Expanding the Frontiers of Industrial Robots Beyond Factories: Design and Validation in the Wild of a Multimodal, Affective and Cognitive Robotic System, *Machines*, 10(12), 1179, October 2022, DOI: 10.3390/machines10121179, *Associated to doctoral dissertation of Siméon Capy

Conference

- T. Katsumata, S. Hagane, B. Navaro, P. Fraisse, G. Venture, Adaptive cartesian force control using payload dynamics and geometric identification, The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec2017), Fukushima Japan, Session 1P1-H11, November 2017, ISSN: 2424-3124, DOI: 10.1299/jsmermd.2017.1P1-H11, *Associated to Chapter 3

- S. Hagane, L. Rincon, T. Katsumata, G. Venture, Adaptive Generalized Predictive Control for 7 DoF Robot Arm, The Proceedings of 23th Robotics Symposia, Shizuoka Japan, Session 5C1, pp 324-329, March 2018, *Associated to Chapter 3

- S. Hagane, L. Jamone, G. Venture, Linearizing Robotic Manipulator's Dynamics Using Koopman Operator and Applying Generalized Predictive Control, 2023 IEEE/SICE International Symposium on System Integration (SII2023),WeP2M2.6, Atlanta USA, January 2023, *Associated to Chapter 4

- D. Deuff, I. Milleville-Pennel, I. Ocnarescu, D. Garcin, C. Aznar, S. Capy, S. Hagane, P. Osorio, E. Coronado, L.K. Rincon-Ardila, and G. Venture, Together Alone, Yōkobo, a Sensible Presence Robject for the Home of Newly Retired Couples, Association for Computing Machinery, 2022 Designing Interactive Systems Conference (DIS'22), Virtual Event and Australia, pp1773–1787, June 2022, DOI 10.1145/3532106.3533485 *Associated to doctoral dissertation of Dominique Deuff

- S. Capy, E. Coronado, P. Osorio, S. Hagane, D. Deuff and G. Venture, Integration of a Presence Robot in a Smart Home, 2023 3rd International Conference on Computer, Control and Robotics (ICCCR2023), Shanghai China, March 2023, Accepted, *Associated to doctoral dissertation of Siméon Capy

Bibliography

[1] T. Katsumata, B. Navarro, V. Bonnet, P. Fraisse, A. Crosnier, and G. Venture. Optimal exciting motion for fast robot identification. application to contact painting tasks with estimated external forces. *Robotics and Autonomous Systems*, 113:149–159, 2019.

[2] Clarke D. W. Generalized predictive control, part i, the basic algorithm. *Automatica*, 23(2):137, 1987.

[3] A. Mehrabian. Pleasure-arousal-dominance : A general framework for describing and measuring individual differences in temperament. *J. of Current Psychology*, 14(4):261–292, 1996.

[4] G.H. Wallbott and R.K. Scherer. Cues and channels in emotion recognition. *Journal of Personality and Social Psychology*, 51:690–699, 1986.

[5] M.M. Bradley and P.J. Lang. Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(1):49–59, 1994.

[6] A. Smith and M. Anderson. Automation in everyday life.

[7] Provide Japan. ”感情重視?効率重視?"ロボットではなく人間がやった方がいいと思う仕事は半数以上の方が”医療や介護のスタッフ"と回答.背景にあるのは”安心感"や”信頼感"といった感情にあることが判明.

[8] H. Knight. *Expressive Motion for Low Degree-of-Freedom Robots*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, August 2016.

[9] A. Beck, L. Cañamero, A. Hiolle, L. Damiano, P. Cosi, F. Tesser, and G. Sommavilla. Interpretation of emotional body language displayed by a humanoid robot: A case study with children. *International Journal of Social Robotics*, 5, 08 2013.

[10] J.A. Claret, G. Venture, and L. Basañez. Exploiting the robot kinematic redundancy for emotion conveyance to humans as a lower priority task. *International Journal of Social Robotics*, 9(2):277–292, 1 2017.

[11] A. Lim, T. Ogata, and G.H. Okuno. Converting emotional voice to motion for robot telepresence. In *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pages 472–479, 2011.

[12] Y. Yamamoto. Pseudo-interpersonal behavior - conditions for induction. *Cognitive Studies*, 1(2):2–95–99, 1994.

[13] M. Juang. Next-gen robots: The latest victims of workplace abuse.

[14] Hiroshi Ashida. Visual psychophysics (8): Visual motion perception and motion pictures. *Kyokai Joho Imeji Zasshi/Journal of the Institute of Image Information and Television Engineers*, 58(8):1151–1156, 2004.

[15] G. Johsnsson. Visual perception of biological motion and a model for its analysis. *Perception & Psyehophysics*, 14(2):201–211, 1973.

[16] M. Simon. R.i.p., anki: Yet another home robotics company powers down.

[17] Shohei Hagane, Liz Katherine Rincon Ardila, Takuma Katsumata, Vincent Bonnet, Philippe Fraisse, , and Gentiane Venture. Adaptive generalized predictive controller and cartesian force control for robot arm using dynamics and geometric identification. *Journal of Robotics and Mechatronics*, 30(6):927–942, 2018.

[18] J.M. Maciejowski, P.J. Goulart, and E.C. Kerrigan. *Constrained Control Using Model Predictive Control*, pages 273–291. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[19] G. Arfken. *Lagrange Multipliers*. Number 945-950 in 17.6 in Mathematical Methods for Physicists, 3rd ed. Academic Press, 1985.

[20] K. Holkar, K Wagh, and L. Waghmare. An overview of model predictive control. *International Journal of Control and Automation*, 3, 01 2011.

[21] K. Taji. モデル予測制御問題に対するいくつかの試み. In *19th RAMP Symposium*.

[22] J Legarda. *LINEARIZATION TECHNIQUES*, pages 75–95. Springer US, Boston, MA, 2006.

[23] J. J. E. Slotine and W. Li. *Applied nonlinear control*. Prentice Hall, 1991.

[24] Julian Nubert, Johannes Kohler, Vincent Berenz, Frank Allgower, and Sebastian Trimpe. Safe and fast tracking on a robot manipulator: Robust MPC and neural network control. *IEEE Robotics and Automation Letters*, 5(2):3050–3057, 4 2020.

[25] Davide Bazzi, Costanza Messeri, Andrea Maria Zanchettin, and Paolo Rocco. Identification of robot forward dynamics via neural network. pages 13–21, 10 2020.

[26] U. O. Akpan and M. R. Kujath. Stochastic Vibration of a Mobile Manipulator. *Journal of Applied Mechanics*, 64(3):670–675, 09 1997.

[27] K. J. Latawiec. Extended horizon adaptive model algorithmic control. *IFAC Proceedings Volumes*, 30(11):305–310, 1997. IFAC Symposium on System Identification (SYSID'97), Kitakyushu, Fukuoka, Japan, 8-11 July 1997.

[28] R.M.C. De Keyser and A.R. Van Cauwenberghe. Extended prediction self-adaptive control. *IFAC Proceedings Volumes*, 18(5):1255–1260, 1985. 7th IFAC/IFORS Symposium on Identification and System Parameter Estimation, York, UK, 3-7 July.

[29] A. L. Elshafei, G. A. Dumont, and A. Elnaggar. Adaptive gpc based on laguerre-filters modelling. *Automatica*, 30(12):1913–1920, 1994.

[30] Avadesh Meduri, Huaijiang Zhu, Armand Jordana, and Ludovic Righetti. Mpc with sensor-based online cost adaptation, 2022.

[31] T. Henmi. Control parameters tuning method of nonlinear model predictive controller based on quantitatively analyzing. *Journal of Robotics and Mechatronics*, 28(5):695–701, 10 2016.

[32] S.J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

[33] E. F. Camacho and C. (Carlos) Bordons. *Model predictive control*. Advanced textbooks in control and signal processing. Springer, 2nd ed edition, 2004.

[34] D.W. Clarke. Application of generalized predictive control to industrial processes. *IEEE Control Systems Magazine*, 8(2):49–55, 1988.

[35] S. Ajwad, J. Iqbal, M. Ullah, and A. Mehmood. A systematic review of current and emergent manipulator control approaches. *Frontiers of Mechanical Engineering*, 10:1–13, 04 2015.

[36] M. Makarov, M. Grossard, P. Rodríguez-Ayerbe, and D. Dumur. Generalized predictive control of an anthropomorphic robot arm for trajectory tracking. In *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 948–953, 2011.

[37] A. Pedro Aguiar, João P. Hespanha, and Petar V. Kokotović. Performance limitations in reference tracking and path following for nonlinear systems. *Automatica*, 44(3):598–610, 2008.

[38] Christopher G. Atkeson, Chae H. An, and John M. Hollerbach. Estimation of inertial parameters of manipulator loads and links. *The International Journal of Robotics Research*, 5(3):101–119, 1986.

[39] Jun Wu, Jinsong Wang, and Zheng You. An overview of dynamic parameter identification of robots. *Robotics and Computer-Integrated Manufacturing*, 26(5):414–419, 2010.

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

[41] Y. Fujimoto and A. Kawamura. Robust biped walking with active interaction control between robot and environment. In *Proceedings of 4th IEEE International Workshop on Advanced Motion Control - AMC '96 - MIE*, volume 1, pages 247–252 vol.1, 1996.

[42] M. Gautier and W. Khalil. On the identification of the inertial parameters of robots. In *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 2264–2269 vol.3, 1988.

[43] W. Khalil and E. Dombre. *Modeling, Identification and Control of Robots*. Kogan Page Science paper edition Modeling, identification & control of robots. Elsevier Science, 2004.

[44] M. Gautier. Numerical calculation of the base inertial parameters of robots. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 1020–1025 vol.2, 1990.

[45] M. Gautier and G. Venture. Identification of standard dynamic parameters of robots with positive definite inertia matrix. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5815–5820, 2013.

[46] B. Armstrong, O. Khatib, and J. Burdick. The explicit dynamic model and inertial parameters of the puma 560 arm. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 510–518, 1986.

[47] W. Khalil and E. Dombre. Chapter 9 - dynamic modeling of serial robots. In W. Khalil and E. Dombre, editors, *Modeling, Identification and Control of Robots*, pages 191–233. Butterworth-Heinemann, Oxford, 2002.

[48] W. Khalil, M. Gautier, and P. Lemoine. Identification of the payload inertial parameters of industrial manipulators. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4943–4948, 2007.

[49] J. Swevers, C. Ganseman, D.B. Tukel, J. de Schutter, and H. Van Brussel. Optimal robot excitation and identification. *IEEE Transactions on Robotics and Automation*, 13(5):730–740, 1997.

[50] G. Calafiore, M. Indri, and B. Bona. Robot dynamic calibration: Optimal excitation trajectories and experimental parameter estimation. *Journal of Robotic Systems*, 18(2):55–68, 2001.

[51] W. Rackl, R. Lampariello, and G. Hirzinger. Robot excitation trajectories for dynamic parameter estimation using optimized b-splines. In *2012 IEEE International Conference on Robotics and Automation*, pages 2042–2047, 2012.

[52] J. Jin and N. Gans. Parameter identification for industrial robots with a fast and robust trajectory design approach. *Robotics and Computer-Integrated Manufacturing*, 31:21–29, 2015.

[53] R Serban and JS Freeman. Identification and identifiability of unknown parameters in multibody dynamic systems. *Multibody System Dynamics*, 5(4):335–350, 2001.

[54] Etienne Burdet and Alain Codourey. Evaluation of parametric and nonparametric nonlinear adaptive controllers. *Robotica*, 16(1):59–73, 1998.

[55] KS Narendra and K Parthasarathy. Identification and control of dynamical systems using neural network. *IEEE Trans Neural Networks, l999 (l)*, 1992.

[56] Jun Wu, Jinsong Wang, and Zheng You. An overview of dynamic parameter identification of robots. *Robotics and computer-integrated manufacturing*, 26(5):414–419, 2010.

[57] B. O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proc. of the National Academy of Sciences*, 17(5):315–318, 1931.

[58] B. O. Koopman and J. v. Neumann. Dynamical systems of continuous spectra. *Proc. of the National Academy of Sciences*, 18(3):255–263, 1932.

[59] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[60] D Bruder, B Gillespie, C. D. Remy, and R Vasudevan. Modeling and control of soft robots using the koopman operator and model predictive control. *ArXiv*, 2019.

[61] I. Abraham and T. D. Murphey. Active learning of dynamics for data-driven control using koopman operators. *IEEE Trans. on Robotics*, 35:1071–1083, 2019.

[62] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[63] M. Korda and I. Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, jul 2018.

[64] Y Igarashi, M Yamakita, J Ng, and H. H Asada. Mpc performances for nonlinear systems using several linearization models. In *American Control Conf. (ACC)*, pages 2426–2431, 2020.

[65] S. Gregory and S. Zhuoyuan. Koopman operator theory for nonlinear dynamic modeling using dynamic mode decomposition, 2021.

[66] T. Iwata and Y. Kawahara. Controlling nonlinear dynamical systems with linear quadratic regulator-based policy networks in koopman space. In *60th IEEE Conf. on Decision and Control, CDC 2021*, Proc. of the IEEE Conf. on Decision and Control, pages 5086–5091, United States, 2021. IEEE.

[67] V Cibulka, M Korda, T Haniš, and M Hromčík. Model predictive control of a vehicle using koopman operator, 2021.

[68] S Peitz, S. E. Otto, and C. W. Rowley. Data-driven model predictive control using interpolated koopman generators. *SIAM J. on Applied Dynamical Systems*, 19(3):2162–2193, 2020.

[69] Y. Susuki, I. Mezic, F. Raak, and T. Hikihara. Applied koopman operator theory for power systems technology. *Nonlinear Theory and Its Applications, IEICE*, 7(4):430–459, 2016.

[70] L Jamone, B Damas, and J Santos-Victor. Incremental learning of context-dependent dynamic internal models for robot control. In *IEEE Int. Symp. on Intelligent Control*, pages 1336–1341, 2014.

[71] M. Khoramshahi, A. Laurens, T. Triquet, and A. Billard. From human physical interaction to online motion adaptation using parameterized dynamical systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1361–1366, 2018.

[72] D.J. Agravante, A. Cherubini, A. Bussy, P. Gergondet, and A. Kheddar. Collaborative human-humanoid carrying using vision and haptic sensing. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 607–612, 2014.

[73] M. Suphi Erden and T. Tomiyama. Human-intent detection and physically interactive control of a robot without force sensors. *IEEE Transactions on Robotics*, 26(2):370–382, 2010.

[74] W. Astrid, B. Regina, T. Manfred, and Y. Eiichi. Addressing user experience and societal impact in a user study with a humanoid robot. Adaptive and Emergent Behaviour and Complex Systems - Proceedings of the 23rd Convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour, AISB 2009, pages 150–157, December 2009.

[75] S. Ivaldi, S. Anzalone, W. Rousseau, O. Sigaud, and M. Chetouani. Robot initiative in a team learning task increases the rhythm of interaction but not the perceived engagement. *Frontiers in neurorobotics*, 8:5, 02 2014.

[76] H. Kamide, K. Kawabe, S. Shigemi, and T. Arai. Anshin as a concept of subjective well-being between humans and robots in japan. *Advanced Robotics*, 29(24):1624–1636, December 2015.

[77] J. Schmidtler, K. Bengler, F. Dimeas, and A. Campeau-Lecours. A questionnaire for the evaluation of physical assistive devices (quead): Testing usability and acceptance in physical human-robot interaction. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 876–881, 2017.

[78] D. Kulić and E. Croft. Physiological and subjective responses to articulated robot motion. *Robotica*, 25(1):13–27, 2007.

[79] M. Saerbeck and C. Bartneck. Perception of affect elicited by robot motion. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 53–60, 2010.

[80] J. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:1161–1178, 12 1980.

[81] K. Nakagawa, K. Shinozawa, H. Ishiguro, T. Akimoto, and N. Hagita. Motion modification method to control affective nuances for robots. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5003–5008, 2009.

[82] P. Corke. *Dynamics and Control*, pages 251–281. Springer International Publishing, Cham, 2017.

[83] A. Liegeois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(12):868–871, 1977.

[84] C.A. Klein and C.H. Huang. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(2):245–250, 1983.

[85] R.V. Dubey, J.A. Euler, and S.M. Babcock. An efficient gradient projection optimization scheme for a seven-degree-of-freedom redundant robot with spherical wrist. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 28–36 vol.1, 1988.

[86] B. Nemec and L. Zlajpah. Null space velocity control with dynamically consistent pseudo-inverse. *Robotica*, 18, 09 2000.

[87] A. Maciejewski and C. Klein. Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *The International Journal of Robotics Research*, 4, 09 1985.

[88] J. Hollerbach and Ki Suh. Redundancy resolution of manipulators through torque optimization. *IEEE Journal on Robotics and Automation*, 3(4):308–316, 1987.

[89] P. Baerlocher and R. Boulic. Task-priority formulations for the kinematic control of highly redundant articulated structures. In *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, volume 1, pages 323–329 vol.1, 1998.

[90] P. Baerlocher and R. Boulic. An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 20:402–417, 01 2004.

[91] Z.X. Peng and N. Adachi. Compliant motion control of kinematically redundant manipulators. *IEEE Transactions on Robotics and Automation*, 9(6):831–836, 1993.

[92] R. von Laban and L. Ullmann. *Modern Educational Dance*. Macdonald and Evans, 1975.

[93] Tanja Bänziger and Klaus R. Scherer. Using actor portrayals to systematically study multimodal emotion expression: The gemep corpus. In Ana C. R. Paiva, Rui Prada, and Rosalind W. Picard, editors, *Affective Computing and Intelligent Interaction*, pages 476–487, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[94] D. Glowinski, N. Dael, A. Camurri, G. Volpe, M. Mortillaro, and K. Scherer. Toward a minimal representation of affective gestures. *IEEE Transactions on Affective Computing*, 2(2):106–118, 2011.

[95] P.J. Lang. Behavioral treatment and bio-behavioral assessment: Computer applications. In J. B. Sidowski, J. H. Johnson, and T. A. Williams, editors, *Technology in mental health care delivery systems*, pages 119 – 137. Norwood, NJ: Ablex, 1980.

[96] I. Iglesias, M.A. Sebastián, and J.E. Ares. Overview of the state of robotic machining: Current situation and future potential. *Procedia Engineering*, 132:911–917, 2015. MESIC Manufacturing Engineering Society International Conference 2015.

[97] T Brogardh. Robot control overview: An industrial perspective. *Modeling, Identification and Control*, 30, 07 2009.

[98] A. B. Moniz and B.J. Krings. Robots working with humans or humans working with robots? searching for social dimensions in new human-robot interaction in industry. *Societies*, 6(3), 2016.

[99] A. Albu-Schäeffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger. The dlr lightweight robot – design and control concepts for robots in human environments. *INDUSTRIAL ROBOT-AN INTERNATIONAL JOURNAL*, 34:376–385, 08 2007.

[100] M. H. Raibert and J. J. Craig. Hybrid Position/Force Control of Manipulators. *Journal of Dynamic Systems, Measurement and Control*, 103(2):126–133, 06 1981.

[101] A. Wahrburg, S. Zeiss, B. Matthias, and H. Ding. Contact force estimation for robotic assembly using motor torques. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1252–1257, 2014.

[102] W. Khalil and D. Creusot. Symoro : A system for the symbolic modelling of robots. *Robotica*, 15(2):153–161, 1997.

[103] D. Tolani, A. Goswami, and N. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, 62:353–88, 10 2000.

[104] Masayuki Shimizu, Hiromu Kakuya, Woo-Keun Yoon, Kosei Kitagaki, and Kazuhiro Kosuge. Analytical inverse kinematic computation for 7-dof redundant manipulators with joint limits and its application to redundancy resolution. *IEEE Transactions on Robotics*, 24(5):1131–1142, 2008.

[105] Richard P. Paul. *Robot manipulators : mathematics, programming, and control : the computer control of robot manipulators*. The MIT Press series in artificial intelligence. MIT Press, 1981.

[106] G. Venture, P.-J. Ripert, W. Khalil, M. Gautier, and P. Bodson. Modeling and identification of passenger car dynamics using robotics formalism. *IEEE Transactions on Intelligent Transportation Systems*, 7(3):349–359, 2006.

[107] M. Gautier and S. Briot. New method for global identification of the joint drive gains of robots using a known payload mass. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3728–3733, 2011.

[108] T. Katsumata. Adaptive cartesian force control using payload dynamics and geometric identification. Master's thesis, Tokyo University of Agriculture and Technology, 2017.

[109] J. De Schutter and H. Van Brussel. Compliant robot motion ii. a control approach based on external control loops. *The International Journal of Robotics Research*, 7(4):18–33, 1988.

[110] A. D. Luca, B. Siciliano, and L. Zollo. Pd control with on-line gravity compensation for robots with elastic joints: Theory and experiments. *Automatica*, 41(10):1809–1819, 2005.

[111] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control—part i. the basic algorithm. *Automatica*, 23(2):137–148, 1987.

[112] S. Masuda, T Yamamoto, and M. Ooshima. Model predictive control : Iii : Generalized predictive control (gpc) and the relevant topics. *Systems, control and information*, 46(9):578–584, 2002.

[113] John Rossiter. *Model-Based Predictive Control: A Practical Approach*. 01 2003.

[114] I.D. Landau. The r-s-t digital controller design and applications. *IFAC Proceedings Volumes*, 30(6):23–33, 1997. IFAC Conference on Control of Industrial Systems "Control for the Future of the Youth", Belfort, France, 20-22 May.

[115] C. Decker, A.U. Ehrlinger, P. Boucher, and D. Dumur. Application of constrained receding horizon predictive control to a benchmark problem. *European Journal of Control*, 1(2):157–165, 1995.

[116] L. K. R ARDILA. *Estudo do comportamento dinâmico de máquinaferramenta CNC com ênfase na implementação de sistemas de controlen*. PhD thesis, State University of Campinas, 2013.

[117] K. Ogata. 制御対象の種々のモデルに対する一般化予測制御の研究. PhD thesis, Nagoya University, 1993.

[118] I Mezić. On applications of the spectral theory of the koopman operator in dynamical systems and control theory. In *IEEE Conf. on Decision and Control*, pages 7034–7041, 2015.

[119] M. Budiš ić, R. Mohr, and I. Mezić. Applied koopmanism. *Chaos: An Interdisciplinary J. of Nonlinear Science*, 22(4):047510, dec 2012.

[120] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLOS ONE*, 11(2):1–19, 02 2016.

[121] J. L. Proctor, S. L. Brunton, and J. N Kutz. Generalizing koopman theory to allow for inputs and control. *SIAM J. on Applied Dynamical Systems*, 17(1):909–930, 2018.

[122] Matthew O. W., Maziar S. H., Scott T.M. D., Ioannis G. K., and Clarence W. R. Extending data-driven koopman analysis to actuated systems. *IFAC-PapersOnLine*, 49(18):704–709, 2016. IFAC Symp. on Nonlinear Control Systems.

[123] A Mauroy and Jorge Goncalves. *Parameter Estimation and Identification of Nonlinear Systems with the Koopman Operator*, pages 335–357. Springer Int. Publishing, Cham, 2020.

[124] S. MASUDA. A design method for generalized predictive control systems based on a state-space approach. *J. of the Society of Instrument and Control Engineers*, 39(5):326–330, 2000.

[125] Wenjian Yu, Yu Gu, and Yaohang Li. Efficient randomized algorithms for the fixed-precision low-rank matrix approximation. *SIAM J. on Matrix Analysis and Applications*, 39(3):1339–1359, 2018.

[126] Dražen Brščić, Hiroyuki Kidokoro, Yoshitaka Suehiro, and Takayuki Kanda. Escaping from children's abuse of social robots. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI '15, page 59–66, New York, NY, USA, 2015. Association for Computing Machinery.

[127] Jean Decety and Thierry Chaminade. Neural correlates of feeling sympathy. *Neuropsychologia*, 41(2):127–138, 2003. The cognitive neuroscience of social behavior.

[128] M. Hewstone. *Introduction to Social Psychology: A European Perspective*. B. Blackwell, 1988.

[129] Céline Clavel, Justine Plessier, Jean-Claude Martin, Laurent Ach, and Benoit Morel. Combining facial and postural expressions of emotions in a virtual character. In Zsófia Ruttkay, Michael Kipp, Anton Nijholt, and Hannes Högni Vilhjálmsson, editors, *Intelligent Virtual Agents*, pages 287–300, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[130] Y. Nakamura and H. Hanafusa. Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control. *Journal of Dynamic Systems, Measurement, and Control*, 108(3):163–171, 09 1986.

[131] M.J. Montepare, E. Koff, D. Zaitchik, and S.M. Albert. The use of body movements and gestures as cues to emotions in younger and older adults. *Journal of Nonverbal Behavior*, 23:133–152, 1999.

[132] P. Cordier, F.M. Mendés, J. Pailhous, and P. Bolon. Entropy as a global variable of the learning process. *Human Movement Science*, 13(6):745–763, 1994.

[133] D. Bernhardt and P. Robinson. Detecting affect from non-stylised body motions. In *Proceedings of the 2nd International Conference on Affective Computing and Intelligent Interaction*, ACII '07, page 59–70, Berlin, Heidelberg, 2007. Springer-Verlag.

[134] G. Castellano, M. Mortillaro, A. Camurri, G. Volpe, and K.R. Scherer. Automated analysis of body movement in emotionally expressive piano performances. *Music Perception*, 26(2):103–119, January 2008.

[135] P. Siple. *Understanding language through sign language research / edited by Patricia Siple.* Perspectives in neurolinguistics and psycholinguistics. Academic Press, New York, 1978.

[136] J. Van den Stock, R. Righart, and B. Gelder. Body expressions influence recognition of emotions in the face and voice. *Emotion (Washington, D.C.)*, 7:487–94, 09 2007.

[137] R.L. Birdwhistell. *Kinesics and Context: Essays on Body Motion Communication*. Conduct and Communication. University of Pennsylvania Press, Incorporated, 2010.

[138] E. Douglas-Cowie, N. Campbell, R. Cowie, and P. Roach. Emotional speech: Towards a new generation of databases. *Speech Communication*, 40(1):33–60, 2003.

[139] J.R.(Rob) Pijpers, R.D.R. Oudejans, F. Holsheimer, and C.F. Bakker. Anxiety–performance relationships in climbing: a process-oriented approach. *Psychology of Sport and Exercise*, 4(3):283–304, 2003.

[140] T. Yoshikawa. Dynamic manipulability of robot manipulators. *Transactions of the Society of Instrument and Control Engineers*, 21(9):970–975, 1985.

[141] T. Yoshikawa. Manipulability of robotic mechanisms. *The international journal of Robotics Research*, 4(2):3–9, 1985.

[142] P. Chiaacchio and M. Concilio. The dynamic manipulability ellipsoid for redundant manipulators. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 1, pages 95–100 vol.1, 1998.

[143] S Kim, S Yun, and D Shin. Numerical quantification of controllability in the null space for redundant manipulators. *Applied Sciences*, 11(13), 2021.

[144] R.J. Kirschner, N. Mansfeld, S. Abdolshah, and S. Haddadin. Iso/ts 15066: How different interpretations affect risk assessment, 2022.

[145] M.M. Bradley and P.J. Lang. *International Affective Picture System*, pages 1–4. Springer International Publishing, Cham, 2017.

[146] G. Hurtz and J. Donovan. Personality and job performance: The big five revisited. *The Journal of applied psychology*, 85:869–79, 01 2001.

[147] J. Hendriks, W. Hofstee, and B. Raad. *The Five-Factor Personality Inventory: assessing the Big Five by means of brief and concrete statements [ch. 4]*, pages 79–108. Hogrefe and Huber, 08 2002.

[148] D.S. Gosling, P.J. Rentfrow, and B.W. Swann. A very brief measure of the big-five personality domains. *Journal of Research in Personality*, 37(6):504–528, 2003.

[149] De Fruyt, F. and Mccrae, R.R. and Szirmak, A. and Nagy, J. The Five-Factor Personality Inventory as a measure of the five-factor model - Belgian, American, and Hungarian comparisons with the NEO-PI-R. *ASSESSMENT*, 11(3):207–215, 2004.

[150] A. Oshio, S. Abe, and C. Pino. Japanese edition, ten item personality inventory (tipi-j) "sakuseino kokoromi". *The Japanese Journal of Personality*, 21(1):40–52, 2012.

[151] Y. Hu, M. Benallegue, G. Venture, and E. Yoshida. Interact with me: An exploratory study on interaction factors for active physical human-robot interaction. *IEEE Robotics and Automation Letters*, 5(4):6764–6771, 2020.

[152] T. Nomura, T. Kanda, T. Suzuki, and K. Kato. Psychology in human-robot communication: An attempt through investigation of negative attitudes and anxiety toward robots. In *RO-MAN 2004. 13th IEEE international workshop on robot and human interactive communication (IEEE catalog No. 04TH8759)*, pages 35–40. IEEE, 2004.