Synchronized Motion Control of Dual Robot Manipulator Systems

by

Cem Cetin

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2020

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Dual manipulator systems are capable of accomplishing a variety of tasks, such as manipulating large and heavy objects and assembling parts, that might be impossible for single manipulators. These collaborative tasks require dual manipulators to be driven by a controller that can coordinate and synchronize the motions of the individual manipulators. Furthermore, many domestic and industrial applications involve the manipulation of objects with unknown properties. For instance, a cooking robot would have to deal with many ingredients of different sizes and weights. In this thesis, an adaptive synchronous controller for dual manipulator systems, that is capable of motion synchronization while manipulating objects of unknown properties, is presented. This controller only utilizes position information, hence does not require extra instrumentation, such as force sensors, often not found in industrial manipulators. The performance of the controller is validated through simulations which showed that the inclusion of motion synchronization errors in controller design is critical in accomplishing collaborative tasks. The real-life implementation details are also discussed.

Acknowledgements

I would like to express my gratitude to my supervisors, Dr. Baris Fidan and Dr. William Melek, for their continuous support and guidance. I would like to thank my readers, Dr. Stewart McLachlin and Dr. Soo Jeon, for their feedback.

Thank you to my colleague and friend Huseyin Demircioglu for making my time at Waterloo all the better. I am eternally grateful to Misha Nili. This thesis would not be possible without her support and encouragements. Finally, I would like to thank my family for supporting me through this time of my life.

Dedication

This thesis is dedicated to my family.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Domain and Motivation

Robots are becoming more and more prevalent both in domestic and industrial settings such as manufacturing, medical robotics, and humanoid robotics. Articulated manipulators are required to perform many complex tasks, such as assembly and welding. Some of these tasks may involve heavy or big objects which are hard to manipulate with a single arm. Moreover, most of the industrial manipulators are not mobile due to their high speed of operation, limiting their workspace. Dual or multiple arm systems can help resolve the aforementioned issues at the cost of increased computational and system complexity.

One of the reasons for the increased complexity stems from the fact that for any meaningful collaborative task involving more than one manipulator, synchronization of motion of manipulators must be taken into account. This means that the control methodologies for trajectory tracking for single manipulator systems do not straightforwardly scale over to multi-manipulator ones. As such, new control strategies need to be developed to better utilize the advantages of multi-manipulator systems.

In this thesis, an adaptive synchronous position control design for dual manipulator systems is presented and validated. Most of the existing industrial manipulators lack the extra instrumentation required for non-position based synchronization control methods such as force sensors. Hence, the focus on position control design allows for practical implementation on a much wider range of manipulators whose functionalities can be extended for use in collaborative tasks.

1.2 The Control Problem and The Approach

In this thesis, we propose and validate a control scheme capable of coordinating the motion of two manipulators with uncertain dynamics through their desired trajectories, while simultaneously satisfying certain relative position and orientation constraints between their end effectors. Such a controller is necessary in following common situations in industrial settings:

1. Two manipulators moving an object with an unknown mass together.
2. Two manipulators moving separate objects with unknown masses that need to be kept apart at a certain distance and / or orientation during their movement.

When the object to be manipulated is of unknown mass, it adds uncertainty to the dynamics model of the manipulator. Once the object is grasped, it can be seen as a part of the last link of the manipulator. Hence, the last link will have unknown centre of mass and inertia. This uncertainty can be dealt with using an adaptive and/or robust controllers to maintain end effector positioning accuracy.

Position tracking errors when following a trajectory are unavoidable since no controller is perfect. In both aforementioned scenarios, the individual motion controller of each manipulator needs to take into account the other's tracking error. In the first scenario, as the error difference between the manipulators increases, the reaction force on one of the end effectors might increase greatly or one of the manipulators might lose grasp of the object. In the second situation with two separate objects to move, the constraints on the relative position and orientation of the objects directly relates to the position errors of the manipulators. Therefore, synchronization must be incorporated to the overall controller design to achieve a successful trajectory tracking in collaborative tasks.

The derivation of this controller design is introduced gradually, by starting from the simplest case of joint-space trajectory tracking of a single manipulator to more involved task-space trajectory tracking, and ending with simultaneous synchronization and trajectory tracking of a dual manipulator system.

1.3 Thesis Contributions

Some works in the literature developed robust and adaptive methods for coordinated motion planning of dual arm industrial manipulators. However, these methods lack validation

of their performance in realistic scenarios, and practical implementation details. They are often hard to recreate due to either assumed background knowledge or lack of application discussions. The main goal of this thesis is to present and analyze such an adaptive control method using minimal sensors feedback and validate its performance through simulations while discussing real-life practical considerations.

1.4 Thesis Outline

The thesis is structured in the following way. In Chapter 2, relevant background is presented together with a literature review. In Chapter 3, the design and analysis of the individual adaptive tracking control scheme for each of the manipulators, in the joint and the task space, are presented. The design and analysis of the synchronous adaptive controller, which builds on the previous controllers and provides synchronization of motion between the dual arm manipulators with individual trajectory tracking, is provided in Chapter 4. The implementation considerations along with a realistic case study are presented in Chapter 5 to highlight the application and effectiveness of the control scheme developed throughout this thesis. The conclusions are presented in Chapter 6.

Chapter 2

Background and Literature Review

2.1 Background

2.1.1 Dynamics

There are two main methods to derive the dynamics equations of multibody systems including robotic manipulators: Euler-Lagrange and Newton-Euler. Both methods produce the same equations. However, the Euler-Lagrange formulation is more concise and straightforward since the internal forces need not be considered. Dynamics of manipulators are explained in all fundamental robotics textbooks such as [2], [3], [4], [5]. The formulation presented in [2] is repeated here for completeness. The Lagrangian is defined as:

$$\mathcal{L} = \mathcal{K} - \mathcal{P} \tag{2.1}$$

where $\mathcal{K}$ is the kinetic energy and $\mathcal{P}$ is the potential energy of the system. For a generalized coordinate x and generalized force f, the following relationship holds:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} - \frac{\partial \mathcal{L}}{\partial x} = f \tag{2.2}$$

For a manipulator, the generalized coordinates are the Denavit–Hartenberg (DH) parameters q_i and the generalized forces are the torques in each joint τ_i where i is the joint number. Therefore equation (2.2) becomes:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i \qquad \text{where} \quad i = 1, .., n. \tag{2.3}$$

Kinetic energy of the system can be calculated by summing the kinetic energies of the links due to linear and angular velocities:

$$\mathcal{K} = \frac{1}{2} m v^T v + \frac{1}{2} \omega^T \mathcal{I} \omega \,, \tag{2.4}$$

where m is the total mass, v and ω are the linear and angular velocity vectors respectively and $\mathcal{I}$ is the inertia tensor. All these terms are expressed in the inertial frame. The inertia tensor in the body attached frame is a constant matrix whereas in the inertial frame, it changes with time. The inertia tensor in the inertial frame ($\mathcal{I}$) is related to body attached frame inertial tensor (I) with the following equation:

$$\mathcal{I} = R I R^T \,, \tag{2.5}$$

where R is the orientation transformation from the body attached frame and the inertial frame. The linear and angular velocities can be calculated using the Jacobians corresponding to each term.

$$v_i = J_{v_i}(q)\dot{q} \,, \qquad \omega_i = J_{\omega_i}(q)\dot{q} \,. \tag{2.6}$$

Hence the kinetic energy of the system can be expressed as

$$\begin{aligned} K &= \frac{1}{2}\dot{q}^T \left[\sum_{i=1}^{n} m_i J_{v_i}^T J_{v_i} + J_{\omega_i}^T R_i I R_i^T J_{w_i}^T \right] \dot{q} \\ &= \frac{1}{2}\dot{q}^T D(q) \dot{q} \,. \end{aligned} \tag{2.7}$$

D is called the inertia matrix and it is an $n \times n$, symmetric and positive definite matrix. Equation (2.7) can also be rewritten by expressing the matrix multiplication as a summation

$$K = \sum_{i,j}^{n} d_{ij}(q) \dot{q}_i \dot{q}_j \,,$$

where d_{ij} represents the i,j-th element of the D matrix. Potential energy of each link i is given by

$$P_i = m_i g^T r_{ci}\,, \tag{2.8}$$

where m_i is the mass of the link, g is the gravity force vector expressed in the inertial frame and r_{ci} is the coordinate of the centre of mass of link i. Therefore, the Lagrangian of an n-link robotic arm is

$$\begin{aligned} L &= K - P \\ &= \frac{1}{2}\sum_{i,j}^{n} d_{i,}(q)\dot{q}_i\dot{q}_j - P(q)\,. \end{aligned} \tag{2.9}$$

Since the kinetic and potential energy of the system is now defined, the partial derivatives in (2.3) can be calculated. First, the partial derivative with respect to the velocity of each joint i is given by

$$\begin{aligned} \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} &= \frac{d}{dt}\sum_{j}^{n} d_{ij}\dot{q}_j \\ &= \sum_{j}^{n} d_{ij}\ddot{q}_j + \sum_{j}^{n}\frac{d}{dt}d_{ij}\dot{q}_j \\ &= \sum_{j}^{n} d_{ij}\ddot{q}_j + \sum_{j,k}^{n}\frac{\partial d_{ij}}{\partial q_k}\dot{q}_k\dot{q}_j\,. \end{aligned} \tag{2.10}$$

The partial derivative with respect to the position of joint i is

$$\frac{\partial L}{\partial q_i} = \frac{1}{2}\sum_{j,k}^{n}\frac{\partial d_{j,k}}{\partial q_i}\dot{q}_j\dot{q}_k - \frac{\partial P}{\partial q_i}\,. \tag{2.11}$$

Finally, (2.3) can be written for each joint k as

$$\sum_{j}^{n} d_{kj}\ddot{q}_j + \sum_{i,j}^{n}\left[\frac{\partial d_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial d_{ij}}{\partial q_k}\right]\dot{q}_i\dot{q}_j + \frac{\partial P}{\partial q_k} = \tau_k\,. \tag{2.12}$$

By exploiting the symmetry of the inertia matrix, it can be shown that

$$\sum_{i,j}^{n}\left[\frac{\partial d_{kj}}{\partial q_i}\right]\dot{q}_i\dot{q}_j = \frac{1}{2}\sum_{i,j}^{n}\left[\frac{\partial d_{kj}}{\partial q_i}+\frac{\partial d_{ki}}{\partial q_j}\right]\dot{q}_i\dot{q}_j\,. \tag{2.13}$$

Therefore,

$$\begin{aligned}\sum_{i,j}\left[\frac{\partial d_{kj}}{\partial q_i}-\frac{1}{2}\frac{\partial d_{ij}}{\partial q_k}\right]\dot{q}_i\dot{q}_j &= \sum_{i,j}\frac{1}{2}\left[\frac{\partial d_{kj}}{\partial q_i}+\frac{\partial d_{ki}}{\partial q_j}-\frac{\partial d_{ij}}{\partial q_k}\right]\dot{q}_i\dot{q}_j \\ &= \sum_{i,j}^{n} c_{ijk}\dot{q}_i\dot{q}_j\,. \end{aligned}\tag{2.14}$$

The terms c_{ijk} are Christoffel symbols of the first kind. By defining the partial derivative of the potential energy with respect to position of generalized coordinates as

$$g_k = \frac{\partial P}{\partial q_k}\,, \tag{2.15}$$

the Euler-Lagrange equation becomes

$$\sum_{j=1}^{n} d_{kj}(q)\ddot{q}_j + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ijk}(q)\dot{q}_i\dot{q}_j + g_k(q) = \tau_k\,, \qquad k = 1, ..., n\,. \tag{2.16}$$

or, in the matrix form,

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau\,. \tag{2.17}$$

Three important properties of the dynamics equation (2.17) are used when designing controllers. These properties are given in the following remarks [2].

Remark 1. *In (2.17), $C(q,\dot{q})$ can be selected as such that the matrix $N(q,\dot{q}) = \dot{D}(q) - 2C(q,\dot{q})$ is skew symmetric (i.e. $n_{jk} = -n_{kj}$).*

Remark 2. *The inertia matrix ($D(q)$) is bounded and positive definite with the eigenvalues $0 < \lambda_1 \leq ... \leq \lambda_n < \infty$.*

Remark 3. *The dynamics equation (2.17) is linear in inertial parameters, i.e., there exists a function $Y(q,\dot{q},\ddot{q})$ such that*

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = Y(q,\dot{q},\ddot{q})\Theta\,, \tag{2.18}$$

where Θ consists of inertial parameters such as link masses, length, and moments of inertia.

2.1.2 Sliding Mode Control

Sliding mode control is a nonlinear control method in which an n dimensional system tracking or regulation problem can be reduced to a 1^{st} order regulation problem. First a *sliding surface* $s(t) = 0$ is defined and the control input is selected such that the system dynamics tends to this *sliding surface* and "slides" on it, i.e. converges to the equilibrium set defined by $s(t) = 0$ (as illustrated in Figure 2.1). The main controller task is performed on the reduced order system hence it is a type of model order reduction [6].

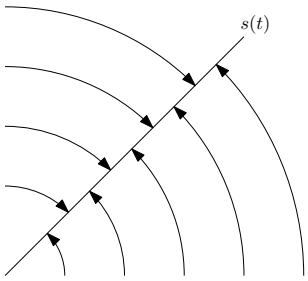

Figure 2.1: In sliding mode control, the states move to the sliding surface and slide on it towards the origin.

The main control objective for manipulators is trajectory tracking which can be expressed as convergence of x to x_d, where x and x_d are actual and desired states, respectively. The state x is the same dimension as the degrees of freedom for a non-redundant manipulator. Sliding mode control allows capturing the trajectory tracking sufficiently with the surface which is a scalar. The surface (s) can be defined as

$$s(x;t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x}\,, \tag{2.19}$$

where n is the dimension of the system (i.e. 6 for a 6 degrees of freedom non-redundant manipulator), $\tilde{x}$ is the tracking error and λ is a positive-definite matrix, a design parameter. Given the initial condition $x_d(0) = x(0)$, the surface s is representative of the true measure of tracking performance [7] and as such the bounds on s translates to bounds on tracking error, $\tilde{x}$.

Design of a controller such that

$$\frac{1}{2}\frac{d}{dt}s^2 \leq -\eta|s|\,, \tag{2.20}$$

where η is a positive constant makes sure that the states converge to the sliding surface $s(t) = 0$. The design can be extended to MIMO (multiple input, multiple output) manipulator system cases, defining s as a vector of sliding mode variables instead of a scalar variable and replacing (2.20) with

$$\frac{1}{2}\frac{d}{dt}s^T s \leq -\eta(s^T s)^{1/2}\,. \tag{2.21}$$

2.2 Literature Review

2.2.1 Single Arm Controllers

The dynamics of manipulators are nonlinear due to the coupling between links. Control designs considering this nonlinear dynamics without modeling simplification lead to sophisticated nonlinear control structures. Most of the more practical and effective approaches model the coupling and other nonlinear dynamic effects as disturbances and treat the system as multiple SISO (single input, single output) systems, one system for each joint. This approach, called independent joint control [2], allows use of various well developed linear control structures to be used such as proportional-integral-derivative (PID) and Linear Quadratic Regulation (LQR) controllers. However, to reduce the coupling effects (hence, the disturbances) the gear ratio between the motors and links need to be increased. This results in slower action response of the links, defeating the purpose of using robotic arms in high throughput environments.

Ineffectiveness of independent joint control prompted researchers to look into different methods that exploit some of the properties of robot dynamics. One of these methods called computed torque control, first explained in [8] and [9], takes advantage of the fact that the inertia matrix D is invertible. By choosing the input torque τ in the form

$$\tau = Da_q + C\dot{q} + g\,, \tag{2.22}$$

(2.17) becomes equivalent to the simple double integrator dynamics

$$\ddot{q} = a_q\,. \tag{2.23}$$

At this point, the input for the system is transformed from torque to acceleration and the equivalent closed-loop system is linear. Various linear control methods can be

used to obtain the desired performance for (2.23). However, exact knowledge of D and C matrices are required to implement the control law (2.22). This drawback can be addressed with the use of adaptive and/or robust control schemes. To make the computed torque control law (2.22) to be robust to uncertainties and time variations, another term is added. This additional term is designed according to the bounds described in Remark 2. Robust computed torque methods were introduced in [10] and [11]. These methods are similar to the sliding mode control, and have been applied to manipulators in [12]. In adaptive control, the control parameters are tuned on-line as opposed to non-adaptive controllers with constant parameters. The on-line tuning of the control parameters is done using an adaptive law, that is typically defined to minimize a model matching cost function via gradient descent or least-squares (LS). For defining such an adaptive law, first, the system model is written in parametric form where the vector of parameters that need to be estimated are separated from the measurable system variables, that are typically collected in a single matrix or vector, called the regressor. For robotic arms such parameterization exists based on Remark 3. In this case, Y is the regressor and Θ is the vector of parameters. The adaptive law is constructed such that the parameter estimates tend to their true values with time. A key example of adaptive computed torque control can be seen in [13]. Several other control strategies are developed for robotic arms other than the computed torque approach using different properties of robot dynamics such as passivity [14] where robust and adaptive control techniques are also employed.

2.2.2 Dual Arm Controllers

To take advantage of multiple manipulator systems, the movement of the manipulators must be coordinated. Coordination is achieved by constraining the kinematic and dynamic relationship between individual arms. Carrying the same load, an assembly task where one arm holds the piece while the other arm assembles a component onto it are some examples of coordinated motions. Most of the robot manipulator control literature, as also mentioned in Subsection 2.2.1, is focused on single manipulator systems; controller design for multiple arm systems still has unsolved challenges. The traditional approaches to multiple arm system control are master-slave control [15] and cooperative control [16]. The downside of the designs in [15], [16] is the need for force measurements which require extra hardware (i.e. force sensors). Also, the estimation of internal forces may not be reliable in real life problems [17]. Position synchronized control addresses this problem by using only the kinematic properties in controller design [18]. Next, an overview of the existing literature for position synchronized control will be given since this is the method employed in this thesis.

First controllers designed for position synchronized control were in the joint space of the robots as opposed to the task space since it simplifies the problem. [19] developed an adaptive synchronized controller for an assembly task using the cross coupling technique (feeding one manipulators kinematic information to another's controller). [20] introduced a velocity observer to estimate the velocity of each joint since this information is not available in real life systems. Uncertainties in kinematic and dynamic properties of robots were considered in some of the designs such as the aforementioned adaptive controller in [19], and the robust adaptive terminal sliding mode synchronized controller in [21].

Aforementioned examples of coordinated motions are done in the task space (i.e. end effector positions in cartesian coordinates) and as such controllers designed in task space are more suitable for multiple manipulator systems. Research on task space synchronization control focuses on different parts of the problem. Namely, uncertainties in kinematics and/or dynamics and how each robot share information (the properties of the network). [22] and [23] deal with kinematic and dynamic uncertainties by estimating the Jacobian. They differ in the topology of the network graph and [22] only considers dual arm systems whereas others do not have this limitation. However, in this thesis only dual arm systems are considered and as such the differences in topology or limitation on robot count do not matter. Another approach is to exploit the passivity property of the manipulator dynamics. [24] and [25] are examples of adaptive controller design for multiple manipulator systems based on passivity.

2.2.3 Applications of Dual Manipulator Systems

Many domestic and industrial applications of dual manipulator systems are studied in the literature. Due to the fact that deformable object manipulation is easier to accomplish with two manipulators, clothes manipulation is one of the focus areas of domestic applications. [26] is one of the earliest research on this subject and focuses on the planning of unfolding clothes that are grasped by two manipulators. Similarly, [27] utilizes two industrial manipulators equipped with inchworm grippers to perform edge tracing to spread a towel. [28] uses the Willow Garage PR2 robotic platform which is equipped with dual manipulators to demonstrate the effectiveness of their cloth grasp point detection algorithm. Cooking is another area of domestic applications that is being explored. [29] accomplishes several cooking tasks such as peeling vegetables and making a salad with a humanoid robot. [30] presents an experiment involving two PR2 robots working collaboratively to make pancakes.

When it comes to industrial applications, the main focus of research is on assembly tasks in manufacturing environments. [31] is an example of an early concept on how dual

manipulator systems can be used to assemble a gearbox. [32] demonstrates a common assembly problem of inserting a peg into a slot with dual manipulators. Another frequent assembly task of screwing nuts is successfully performed by a dual manipulator system in [33]. In addition to assembly tasks, flexible beam deforming has important industrial applications. A control algorithm for a dual manipulator system capable of bending and manipulating sheet metal is proven with experiments in [34]. Two Puma 560 manipulators are utilized in [35] to bend a flexible sheet.

Chapter 3

Adaptive Controller Design

In this chapter, an adaptive controller for a single manipulator is developed to realize a desired motion when an object with unknown mass is being manipulated. This design will later be combined with the synchronization and coordination control design to be presented in Chapter 4. First, the controller is designed for trajectories given in the joint space and then the task space, which introduces more complexity. The design process is accompanied by examples from a planar arm with two links. However, the choice of a manipulator with two degrees of freedom is only to restrict the size of matrices and the controller is applicable to spatial robot arms with higher degree of freedom as well. This controller design follows the design steps presented in [36], [7].

3.1 Joint Space Adaptive Controller

In this section a robust controller for joint space trajectories is developed by utilizing sliding control. The preliminaries of such controllers are given in Section 2.1.2. The first step is to define the sliding surface that will capture the states of the system. The states in the case of a manipulator are represented by q, the vector denoting the joint angles, since the controller operates in the joint space. The definition of a surface were given in Equation (2.19). For a manipulator the sliding surface can be defined as

$$s = \dot{\tilde{q}} + \Lambda\tilde{q} = \dot{q} - \dot{q}_r = 0\,. \tag{3.1}$$

Λ is a positive-definite matrix that can be seen as the analog of proportional control

in PID systems. $\dot{q}_r$ is called the reference velocity which can be thought as a shift in the desired velocities ($\dot{q}_d$) according to joint position errors ($\tilde{q}$).

$$\dot{q}_r = \dot{q}_d - \Lambda\tilde{q} \tag{3.2}$$

The states of the manipulator (q) following the reference state (q_r) means that they are on the surface defined in Equation (3.1). Then the dynamic equation of the system becomes

$$D(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r = Y_r(q,\dot{q},\dot{q}_r,\ddot{q}_r)\Theta\,. \tag{3.3}$$

Example 3-1. Deriving Y_r For a Planar Manipulator

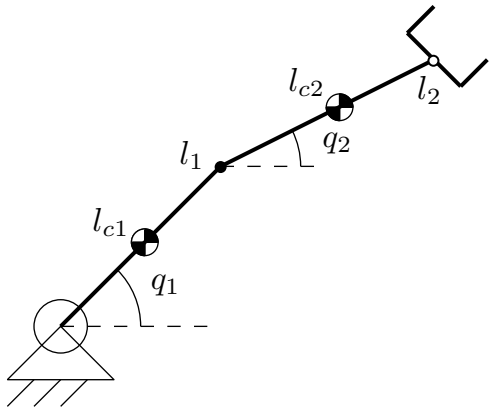

Figure 3.1: Planar manipulator with two revolute joints.

Consider the planar manipulator with two revolute joints, depicted in Figure 3.1, with link mass m_i, link moment of inertia I_i, joint angle q_i, link length l_i and centre of mass length l_{ci} for $i = 1, 2$. The forward kinematics of the planar arm is formulated using geometry as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} l_1\cos(q_1) + l_2\cos(q_2) \\ l_1\sin(q_1) + l_2\sin(q_2) \end{bmatrix}\,. \tag{3.4}$$

x represents the end effector position vector. The Jacobian ($J = \frac{\partial x}{\partial q}$), its derivative $\dot{J}$ and its inverse J^{-1} are required for the design of the proposed task space adaptive controller.

$$J = \begin{bmatrix} -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{bmatrix} \tag{3.5}$$

$$\dot{J} = \begin{bmatrix} -l_1 \cos q_1 \dot{q}_1 - l_{c2} \cos(q_1 + q_2)\,(\dot{q}_1 + \dot{q}_2) & -l_{c2} \cos(q_1 + q_2)\,(\dot{q}_1 + \dot{q}_2) \\ -l_1 \sin q_1 \dot{q}_1 - l_{c2} \sin(q_1 + q_2)\,(\dot{q}_1 + \dot{q}_2) & -l_{c2} \sin(q_1 + q_2)\,(\dot{q}_1 + \dot{q}_2) \end{bmatrix} \tag{3.6}$$

$$J^{-1} = \begin{bmatrix} \frac{\cos(q_1+q_2)}{l_1 \sin(q_1)} & \frac{\sin(q_1+q_2)}{l_1 \sin(q_2)} \\ -\frac{\cos(q_1)}{l_{c2} \sin(q_2)} - \frac{\cos(q_1+q_2)}{l_1 \sin(q_2)} & -\frac{\sin(q_1)}{l_{c2} \sin(q_2)} - \frac{\sin(q_1+q_2)}{l_1 \sin(q_2)} \end{bmatrix} \tag{3.7}$$

The determinant of the Jacobian (3.8) shows that singularities will occur when $q_2 = 2\pi k$ where $k = 0, 1, ..., n$.

$$|J| = l_1 l_{c2} \sin(q_2) \tag{3.8}$$

The dynamics of the planar arm is given in Equation (3.9). Each term is defined the same as the dynamics equation for manipulators given in Equation (2.17) in Chapter 2.

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

$$\begin{bmatrix} m_1 l_{c1}^2 + m_2 \left(l_1^2 + l_{c2}^2 + 2 l_1 l_{c2}^2 + 2 l_1 l_{c2} \cos q_2\right) + I_1 + I_2 & m_2 \left(l_{c2}^2 + l_1 l_{c2} \cos q_2\right) + I_2 \\ m_2 \left(l_{c2}^2 + l_1 l_{c2} \cos q_2\right) + I_2 & m_2 l_{c2}^2 + I_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \tag{3.9}$$

$$\begin{bmatrix} -m_2 l_1 l_{c2} \sin q_2 \dot{q}_2 & -m_2 l_1 l_{c2} \sin q_2\,(\dot{q}_1 + \dot{q}_2) \\ m_2 l_1 l_{c2} \sin q_2 \dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

As mentioned before, the purpose of an adaptive controller is to command the manipulator to a certain position in the absence of parameter knowledge. In order to do that, the parameter terms need to be separated from the rest. This is possible due to Remark 3 in Chapter 2. Let Θ and Y_r represent a collection of inertial parameters, and the regressor matrix containing rest of the terms making up the dynamics equation respectively. For the two link planar arm case, Θ can be found as

$$\Theta = \begin{bmatrix} I_1 + m_1 l_{c1}^2 + I_2 + m_2 l_{c2}^2 + m_2 l_1^2 \\ I_2 + m_2 l_{c2}^2 \\ m_2 l_1 l_{c2} \end{bmatrix} . \tag{3.10}$$

In order to find the regressor matrix, D and C matrices need to be expressed in terms of Θ,

$$D = \begin{bmatrix} \Theta_1 + 2\Theta_3 \cos(q_2) & \Theta_2 + \Theta_3 \cos(q_2) \\ \Theta_2 + \Theta_3 \cos(q_2) & \Theta_2 \end{bmatrix} , \tag{3.11}$$

$$C = \begin{bmatrix} -\Theta_3 \sin(q_2)\dot{q}_2 & -\Theta_3 \sin(q_2)\,(\dot{q}_1 + \dot{q}_2) \\ \Theta_3 \sin(q_2)\dot{q}_1 & 0 \end{bmatrix} . \tag{3.12}$$

Y_r can be calculated by plugging in D and C matrices from Equations (3.11) and (3.12) respectively into (3.3).

$$Y_r = \begin{bmatrix} \ddot{q}_{r1} & \ddot{q}_{r2} & 2\ddot{q}_{r1}\cos(q_2) + \ddot{q}_{r2}\cos(q_2) - \dot{q}_2\dot{q}_{r1}\sin(q_2) - (\dot{q}_1 + \dot{q}_2)\,\dot{q}_{r2}\sin(q_2) \\ 0 & \ddot{q}_{r1} + \ddot{q}_{r2} & \ddot{q}_{r1}\cos(q_2) + \dot{q}_1\dot{q}_{r1}\sin(q_2) \end{bmatrix} \tag{3.13}$$

In order to generate the torque τ to drive the system states to the sliding surface, a constructive Lyapunov analysis is carried out. Note that, the matrix $D(q)$ is always positive definite per mark 2, consider the positive definite (Lyapunov) function,

$$V_0(t) = \frac{1}{2}(s^T D s) . \tag{3.14}$$

The time derivative of V_0 is calculated, using (3.1), as

$$\begin{aligned} \dot{V}_0(t) &= s^T(D\ddot{q} - D\ddot{q}_r) + \frac{1}{2}s^T\dot{D}s , \\ &= s^T(D\ddot{q} - D\ddot{q}_r) + s^T C s . \end{aligned} \tag{3.15}$$

noting that $\dot{D} - 2C$ is skew symmetric per Remark 1. (2.17) is substituted to rewrite (3.15) in terms of the input torque τ, leading to

$$\dot{V}_0(t) = s^T(\tau - g - D\ddot{q}_r - C\dot{q}_r)\,. \tag{3.16}$$

Now the input can be selected in such a way to make sure $\dot{V}_0 < 0$,

$$\tau = \hat{\tau} - k \odot \mathrm{sgn}(s)\,, \tag{3.17}$$

where the sgn() operator applies element-wise to the vector s, and $\hat{\tau} = \hat{D}\ddot{q}_r + \hat{C}\dot{q}_r + g$ is the control input that would make $\dot{V}_0$ equal to zero if the system dynamics were known exactly. The superscript hat represents an estimated value of a parameter (e.g. $\hat{\tau}$ is the estimated value of the torque) whereas the superscript tilde represent the difference between the estimated (or desired) and actual value of a parameter (e.g. $\tilde{D} = \hat{D} - D$).

$$\dot{V}_0 = s^T(\hat{D}(q)\ddot{q}_r + \hat{C}(q,\dot{q})\dot{q}_r - \sum_{n=1}^{n} k_i|s_i|\,, \tag{3.18}$$

where $s = [s_1, \ldots, s_n]^T$, and $k = [k_1, \ldots, k_n]^T$ is a constant vector that must be selected according to the known bounds of the dynamics of the system $k_i \geq \left|\tilde{D}\ddot{q}_r + \tilde{C}(q,\dot{q})\dot{q}_r + \tilde{G}(q)\right|_i + \eta_i$ for certain positive design constants η_i, for $i = 1, \ldots, n$. Choosing k in such a fashion results in the following inequality which represents the sliding condition:

$$\dot{V}_0 \leq -\sum_{n=1}^{n} \eta_i|s_i|\,. \tag{3.19}$$

The sliding condition (3.19) ensures that the trajectories reach the sliding surface $s = 0$ and remain there. To incorporate parameter estimation into Lyapunov analysis, the Lyapunov function V_0 is replaced, to capsulate the parameter estimation error $\tilde{\Theta} = \hat{\Theta} - \Theta$, with the Lyapunov-like function:

$$V(t) = \frac{1}{2}[s^T D s + \tilde{\Theta}^T \Gamma^{-1} \tilde{\Theta}]\,. \tag{3.20}$$

Using (3.16), the time derivative of (3.20) is calculated as

$$\dot{V}(t) = s^T(\tau - g - D\ddot{q}_r - C\dot{q}_r) + \dot{\hat{\Theta}}^T \Gamma^{-1} \tilde{\Theta}\,. \tag{3.21}$$

Based on (3.21), the control law (3.17) is modified as

$$\tau = Y_r \hat{\Theta} - K_d s\,, \tag{3.22}$$

the term $Y_r\hat{\Theta}$ replaces the non-adaptive term $\hat{\tau}$ in (3.17), written in the form that allows the separation of parameters from system dynamics, and $K_d s$ term is a simple derivative term , with K_d being a positive definite gain matrix, which replaces $k \odot \text{sgn}(s)$. Therefore $\dot{V}$ becomes

$$\dot{V}(t) = s^T Y_r \tilde{\Theta} - s^T K_d s + \dot{\hat{\Theta}}^T \Gamma^{-1} \tilde{\Theta}\,. \tag{3.23}$$

Selecting the adaptive parameter estimation law as

$$\dot{\hat{\Theta}} = -\Gamma Y_r^T s\,, \tag{3.24}$$

results in $\dot{V}(t) = -s^T K_d s \leq 0$. The sliding condition is satisfied again, ensuring the convergence of the states to the surface s and hence showing that $\tilde{q}$ and $\dot{\tilde{q}}$ are tending to zero as t approaches infinity.

The block diagram for the overall system can be seen in Figure 3.2.

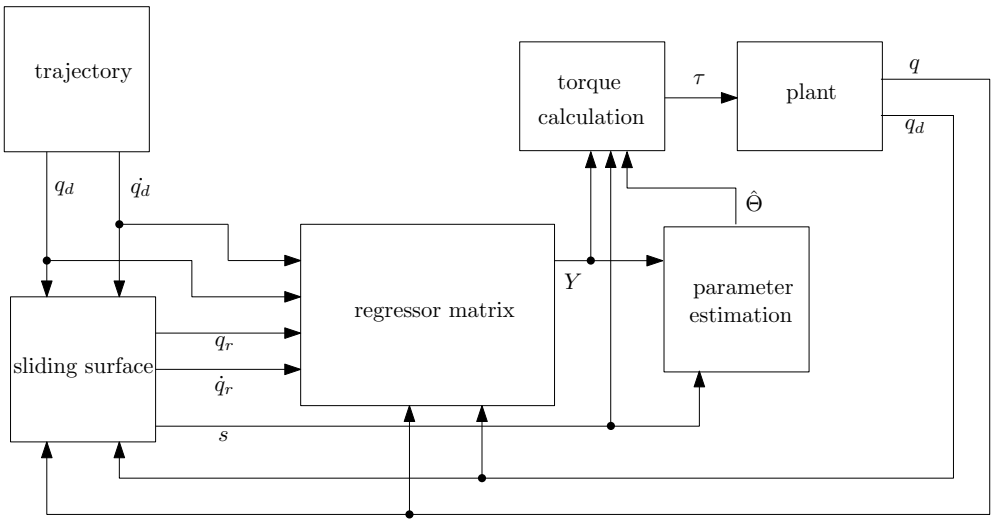

Figure 3.2: Block diagram for the adaptive joint space controller.

3.2 Numerical Simulation of Joint Space Adaptive Controller

To validate the controller developed in Section 3.1, we consider the trajectories

$$
\begin{aligned}
q_{d1} &= 30^\circ \sin(\pi t)\,, \\
q_{d2} &= 45^\circ \sin(\pi t)\,,
\end{aligned} \tag{3.25}
$$

in a simulation of the planar arm shown in Figure 3.1 with properties given in Table 3.1.

Table 3.1: Parameter values of a planar manipulator with two revolute joints.

Parameter	**Description**	**Value**	**Unit**
l_1	Length of link 1	1	m
l_{c1}	Length of centre of mass of link 1	0.5	m
m_1	Mass of link 1	1	kg
I_1	Moment of inertia of link 1	.12	$\frac{kg}{m^2}$
l_2	Length of link 2	1.2	m
l_{c2}	Length of centre of mass of link 2	0.6	m
m_2	Mass of link 2	2	kg
I_2	Moment of inertia of link 2	.25	$\frac{kg}{m^2}$

The design parameters of the simulation are selected as:

$$
\begin{aligned}
\Lambda &= \begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix}\,, \\
\Gamma &= \begin{bmatrix} 0.03 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}\,, \\
K_d &= \begin{bmatrix} 800 & 0 \\ 0 & 800 \end{bmatrix}\,.
\end{aligned} \tag{3.26}
$$

It is assumed that there is no prior knowledge on the system parameters, i.e. the initial estimate is set to $\hat{\Theta}(0) = 0$. The actual joint angle q_i and the desired angle q_{di} for $i = 1, 2$ are plotted in Figures 3.3 and 3.4 respectively. Even without any knowledge of the system parameters, the controller effectively tracks the desired trajectory. The lack of assumed knowledge on the link masses and inertias also implies that this controller is able to perform the same if an unknown object was attached to the end effector.

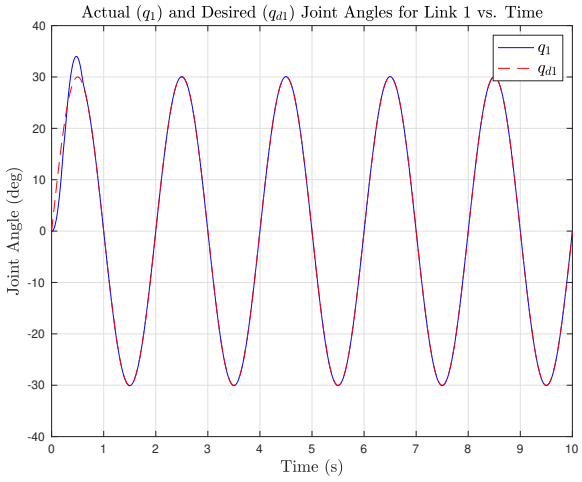

Figure 3.3: The actual and desired joint angles of link 1 of the adaptive joint space controller.

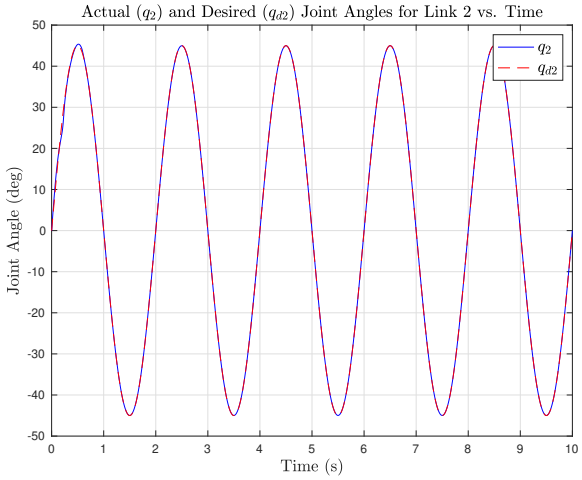

Figure 3.4: The actual and desired joint angles of link 2 of the adaptive joint space controller.

The root mean square error (RMSE) of the joint positions, $\tilde{q}_1$ and $\tilde{q}_2$, are tabulated in Table 3.2. Excluding the first few seconds of the motion decreases the RMSE since the estimation of unknown system parameters have not converged to satisfactory values by then. Even with the beginning of the motion included, the RMSE is less than a degree,

showcasing the effectiveness of the control design.

Table 3.2: The root mean square errors of joint position of the adaptive joint space controller simulation.

Parameter	**RMSE**	**RMSE After 1s**	**RMSE After 2s**
$\tilde{q}_1$	0.9907°	0.0357°	0.0333°
$\tilde{q}_2$	0.2566°	0.0185°	0.0176°

The parameter estimation is shown in Figure 3.5. The actual parameter vector of the system is $\Theta = \begin{bmatrix} 4.98 & 0.97 & 1.2 \end{bmatrix}^T$ and as such it is clear that the parameter estimates are not converging to the actual parameter values. However, this is expected since the controller is designed to follow the trajectory, not to estimate the parameters accurately as seen in the Lyapunov analysis.

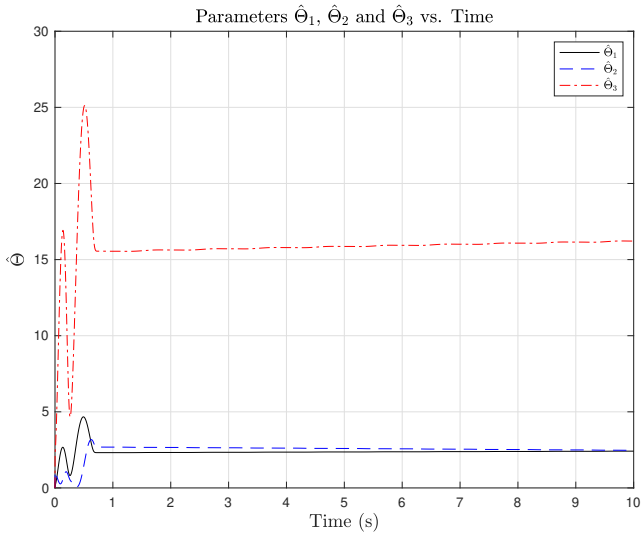

Figure 3.5: Parameter estimation of the adaptive joint space controller.

3.3 Task Space Adaptive Controller

Following the kinematic relation $\dot{x} = J\dot{q}$ [4], where x denotes the end effector position vector, the surface defined in Equation (3.1) can be written in the task space by expressing reference velocity and accelerations in terms of end effector positions, as

$$\dot{q}_r = J^{-1}\left[\dot{x}_d - \Lambda\tilde{x}\right] , \tag{3.27}$$

$$\ddot{q}_r = J^{-1}\left[\ddot{x}_d - \Lambda\dot{\tilde{x}} - \dot{J}\dot{q}_r\right] , \tag{3.28}$$

where $\tilde{x} = x - x_d$ for the reference position x_d therefore, the sliding surface becomes

$$s = J^{-1}\left[J\dot{q} - \dot{x}_d + \Lambda\tilde{x}\right] . \tag{3.29}$$

The controller design is essentially identical to the joint space controller except the way the surface is calculated. As such, the Lyapunov analysis that was carried out in Section 3.1 holds true for the task space controller. The desired joint angles (q_d) are not needed as seen from Equation (3.28). Hence, inverse kinematics calculations are not necessary. The only extra step is to calculate the real end effector positions using forward kinematics once the joint angles are read through the encoders.

There are two major implications of this definition of the surface in the task space.

1. The Jacobian and forward kinematics of the system must be known.
2. Since the inverse of Jacobian is utilized, the determinant of the Jacobian cannot be zero (ie. singularities must be avoided).

As seen in Equation (3.5), and Equation (3.4) the Jacobian and the forward kinematics only require the knowledge of link lengths and joint angles. It is reasonable to assume that the link length of a robot will be known to the controller designer. However, if the arm is picking up an object of an unknown length, the length of the last link becomes unknown as well. Because of this, it is assumed in this controller design that the object being picked up is not of a significant length to affect the controller stability.

The desired trajectory must be designed to avoid singularities. In the case of a planar arm, the determinant of the Jacobian given by Equation (3.8) necessitates that the joint angle of link 2 cannot be zero during the trajectory execution.

3.4 Numerical Simulation of Task Space Adaptive Controller

The effectiveness of this controller is demonstrated with a simulation for the same planar arm depicted in Figure 3.1 with parameters given in Table 3.1. For the purpose of the simulation, the desired trajectory is designed to be a third order polynomial, so that it generates smooth velocities and accelerations.

$$x_d(t) = At^3 + Bt^2 + Ct \tag{3.30}$$

where A, B and C are 2×1 coefficient matrices that needs to be calculated according to initial ($x(0)$) and final ($x(t_f)$) desired positions and the time it takes to complete the trajectory (t_f). When the initial and final velocity and accelerations are zero, the coefficients can be calculated as

$$\begin{bmatrix} A_i \\ B_i \end{bmatrix} = \begin{bmatrix} t_f^3 & t_f^2 \\ 3t_f^2 & 2t_f \end{bmatrix}^{-1} \begin{bmatrix} x_{di}(t_f) - x_{di}(0) \\ 0 \end{bmatrix} \tag{3.31}$$

$$C_i = x_{di}(0)\,. \tag{3.32}$$

where $i = 1, 2$ for the planar arm. For $x_d(0) = \begin{bmatrix} 0.594 & 0.078 \end{bmatrix}^T$, $x_d(t_f) = \begin{bmatrix} 0.246 & 0.472 \end{bmatrix}^T$ and $t_f = 10$, the trajectory has the properties shown in Figure 3.6. Both joint and effector positions and velocities are smooth. The singularities avoided since q_2 is never zero. The design parameters are chosen the same as the joint space controller parameters given in Equation (3.26).

The simulation is conducted assuming no knowledge of the system parameters, i.e. initial $\hat{\Theta} = 0$. The error ($\tilde{x}$) between desired (x_d) and the actual (x) end effector positions are plotted in Figure 3.7 and 3.8. Even without any knowledge of system parameters, the controller tracked the desired trajectory satisfactorily.

The root mean square error (RMSE) of the end effector position in x ($\tilde{x}_1$) and in y ($\tilde{x}_2$) are tabulated in Table 3.3. As it was the case in the joint space adaptive controller, the RMSE decreases when the first few seconds of the motion is excluded due to uncertainty of the system parameters. Regardless, the RMSE is less than a millimeter during the whole motion.

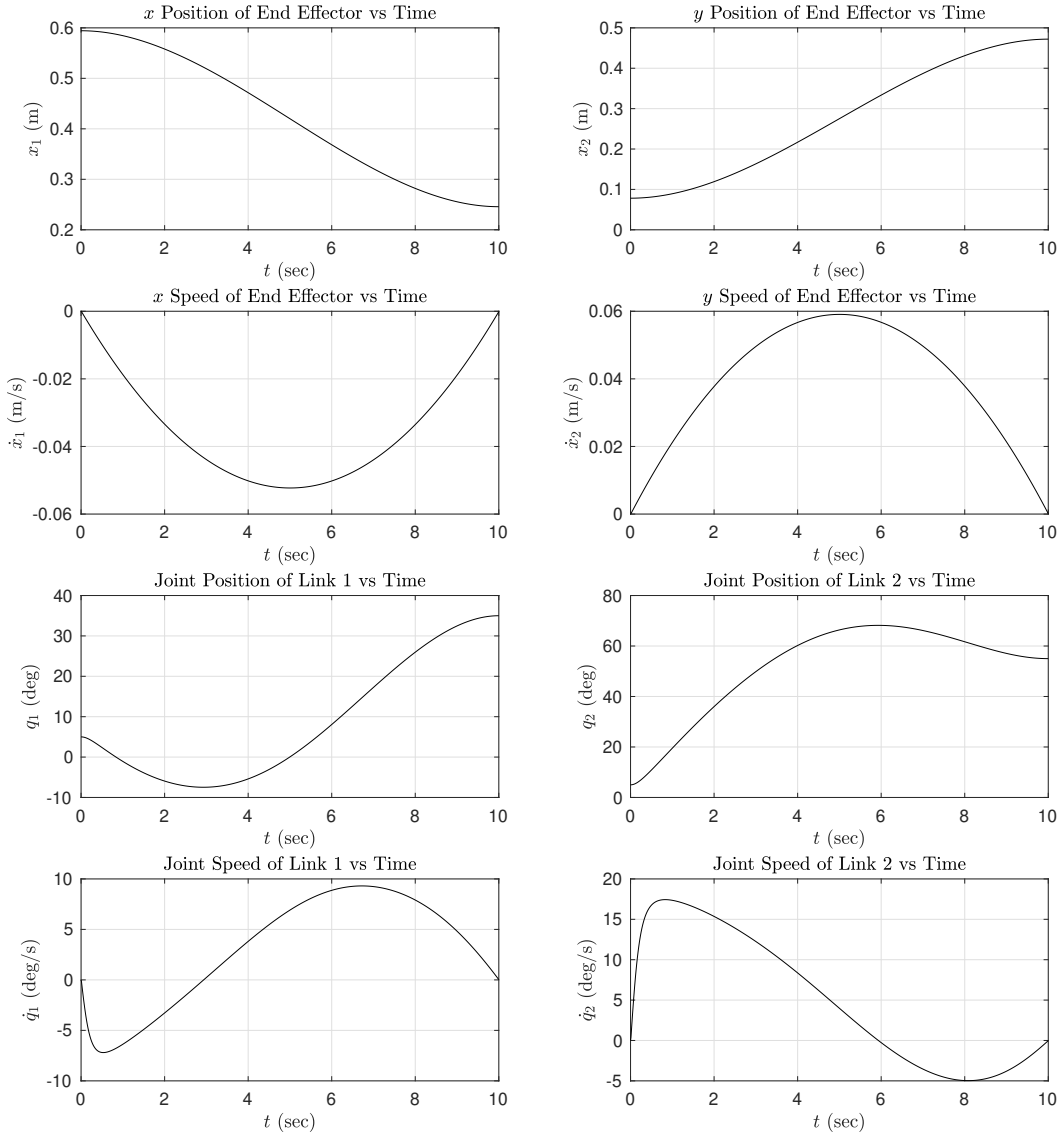

Figure 3.6: The desired trajectory for the task space controller simulation.

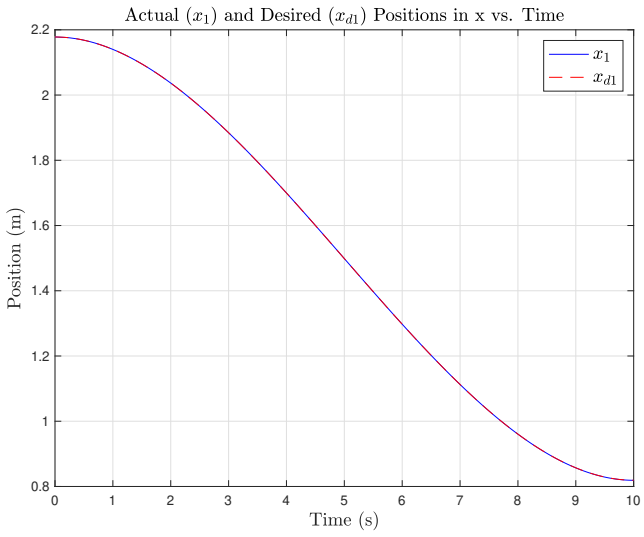

Figure 3.7: Position error in x of the end effector of the adaptive task space controller.

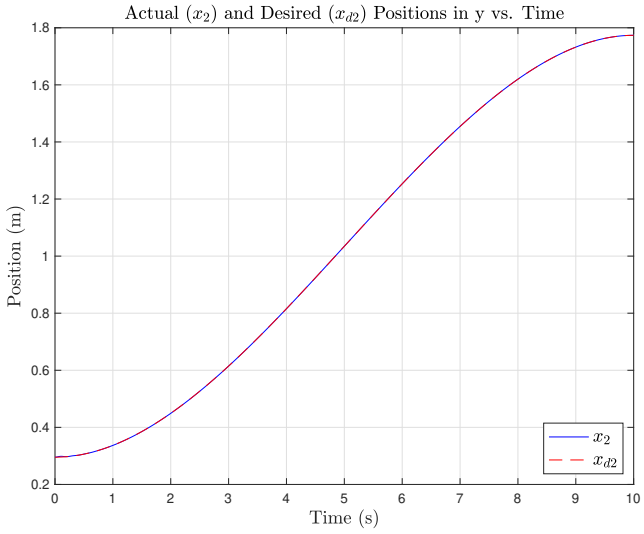

Figure 3.8: Position error in y of the end effector of the adaptive task space controller.

The parameter estimation is shown in Figure 3.9. The real parameters of the system are $\Theta = \begin{bmatrix} 4.98 & 0.97 & 1.2 \end{bmatrix}^T$ and the estimated values are not converging to the real values as was the case for the joint space adaptive controller.

Table 3.3: The root mean square of the end effector position errors of the adaptive task space controller.

Parameter	**RMSE**	**RMSE After 1s**	**RMSE After 2s**
$\tilde{x}_1$	8.231×10^{-5} m	7.572×10^{-5} m	7.531×10^{-5} m
$\tilde{x}_2$	2.519×10^{-4} m	8.994×10^{-5} m	7.988×10^{-5} m

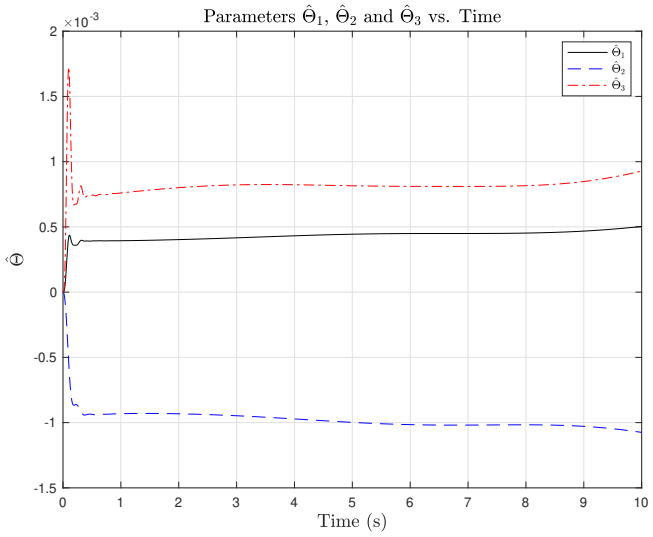

Figure 3.9: Parameter estimation of the adaptive task space controller.

The torques applied to the joints are shown in Figure 3.10. The maximum torque applied is less than $100\,[Nm]$ which is a reasonable amount.

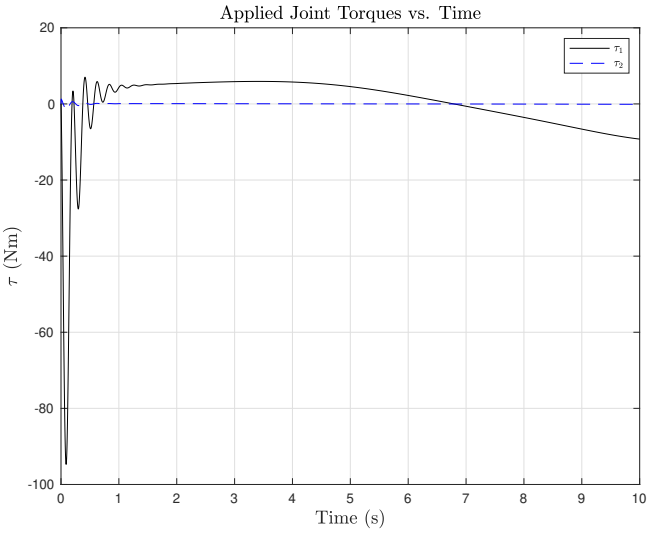

Figure 3.10: Applied torques of the adaptive task space controller.

3.5 Summary

In this chapter, two adaptive controllers, one in the joint and one in the task space, for a single manipulator that is capable of tracking a desired trajectory without any knowledge on system parameters, such as link masses and inertias, were developed and validated through simulations. Since the system parameters are assumed to be unknown, these controllers are also capable of manipulating an object of unknown mass and inertia, granted that the said object's length is not large. The task space adaptive controller is the basis of the adaptive synchronous controller presented in Chapter 4 that extends this design to dual manipulator systems.

Chapter 4

Adaptive Synchronous Controller Design

In this chapter, the controller developed in Chapter 3 will be improved to handle collaborative tasks with dual manipulators such as:

1. Manipulating an object together.
2. Accomplishing a coordinated motion where there are kinematic constraints between two end effectors.

The goal of this controller is to ensure both manipulators track their desired trajectories while ensuring that position errors between the two are minimized in the presence of parameter uncertainty. The joint space synchronization error between the manipulators i and j is defined as

$$\epsilon_{i,j} = \tilde{x}^{(i)} - \tilde{x}^{(j)} \,. \tag{4.1}$$

The superscripts denote the indices of the manipulators. For synchronized tracking, the controller design must guarantee that all the $\tilde{x}^{(i)}$ and $\epsilon_{i,j}$ terms tend to zero as time, t, tends to infinity.

The kinematics and dynamics of each manipulator in the system is the same as the single manipulator case given in Chapter 3 since the controller developed in this chapter consists of decentralized and coupled individual controllers for each manipulator. This design was first presented in [19] and [17] which expand on the adaptive control design for a single manipulator developed in [36] and [7].

4.1 Adaptive Sliding Controller Design

When determining the sliding surface for the adaptive sliding controller for a single manipulator, the difference between the desired and actual positions were considered. Similarly, for synchronization of the motions, the difference between the synchronization errors in Equation (4.1) need to be encoded along side the trajectory errors. Hence, e is defined to encapsulate this idea in Equation (4.2).

$$\begin{aligned} e_1(t) &= \tilde{x}^{(1)}(t) + \beta \int_0^t \epsilon_{1,2}(w)\mathrm{d}w \\ e_2(t) &= \tilde{x}^{(2)}(t) + \beta \int_0^t \epsilon_{2,1}(w)\mathrm{d}w \end{aligned} \tag{4.2}$$

Similar to virtual velocity concept introduced in the previous chapter, u is defined to be the virtual command signal that consists of desired velocities shifted by the accumulated synchronization error scaled by a positive-definite matrix β:

$$\begin{aligned} u_1 &= \dot{x}_d^{(1)} + \beta\epsilon_{1,2} + \Lambda e_1 \\ u_2 &= \dot{x}_d^{(2)} + \beta\epsilon_{2,1} + \Lambda e_2 \end{aligned} \tag{4.3}$$

Then, the sliding surface for each manipulator $i = 1, 2$ is defined as

$$s_i = \dot{e}_i + \Lambda e_i = u_i - \dot{x}^{(i)} = 0\,. \tag{4.4}$$

Λ is a positive-definite design matrix to be chosen. A pair of input torques for the system to slide on the surface (4.4) are given by

$$\begin{aligned} \tau_1 &= \hat{D}_1\dot{u}_1 + \hat{C}_1 u_1 + K_d s_1 + K_\epsilon \epsilon_{1,2} \\ &= Y_1\left(x_1, \dot{x}_1, u_1, \dot{u}_1\right)\hat{\Theta}_1 + K_d s_1 + K_\epsilon \epsilon_{1,2}\,, \\ \tau_2 &= \hat{D}_2\dot{u}_2 + \hat{C}_2 u_2 + K_d s_2 + K_\epsilon \epsilon_{2,1} \\ &= Y_2\left(x_2, \dot{x}_2, u_2, \dot{u}_2\right)\hat{\Theta}_2 + K_d s_2 + K_\epsilon \epsilon_{2,1}\,. \end{aligned} \tag{4.5}$$

where K_d and K_ϵ are positive definite matrices that need to be chosen as well. The adaptation parameter estimation law is chosen, similarly to Chapter 3, as

$$\dot{\hat{\Theta}}_i = -\Gamma Y_i^T s_i \,. \tag{4.6}$$

where Γ is a positive definite gain matrix. Note that the regressor matrix Y_i is defined in terms of $x^{(i)}$, $\dot{x}^{(i)}$, u_i, and $\dot{u}_i$. Substituting (4.5) into the system dynamics results in

$$\begin{aligned} D_1\dot{s}_1 + C_1 s_1 &= Y_1\left(x^{(1)}, \dot{x}^{(1)}, u_1, \dot{u}_1\right)\tilde{\Theta}_1 \,, \\ D_2\dot{s}_1 + C_2 s_2 &= Y_2\left(x^{(2)}, \dot{x}^{(2)}, u_2, \dot{u}_2\right)\tilde{\Theta}_2 \,. \end{aligned} \tag{4.7}$$

Reference velocities and accelerations can be calculated using the following equations so that the same regressor matrix shown in Equation (3.13) can be utilized.

$$\ddot{q}_{ri} = J_i^{-1}\left(\dot{u}_i - \dot{J}_i J_i^{-1} u_i\right) \tag{4.8}$$

$$\dot{q}_{ri} = J_i^{-1} u_i \tag{4.9}$$

4.2 Lyapunov Analysis

First, a positive definite Lyapunov like function is defined as

$$\begin{aligned} V = &\left[\frac{1}{2}s_1^T D_1 s_1 + \frac{1}{2}\tilde{\Theta}_1^T\Gamma_1^{-1}\tilde{\Theta}_1 + \frac{1}{2}\epsilon_{1,2}^T K_\epsilon \epsilon_{1,2}\right] \\ &+ \left[\frac{1}{2}s_2^T D_2 s_2 + \frac{1}{2}\tilde{\Theta}_2^T\Gamma_2^{-1}\tilde{\Theta}_2 + \frac{1}{2}\epsilon_{2,1}^T K_\epsilon \epsilon_{2,1}\right] \\ &+ \frac{1}{2}\left(\int_0^t \epsilon_{1,2}(w)^T \mathrm{d}w\right) K_\epsilon \Lambda\beta \int_0^t \epsilon_{1,2}(w)\mathrm{d}w \\ &+ \frac{1}{2}\left(\int_0^t \epsilon_{2,_1}(w)^T \mathrm{d}w\right) K_\epsilon \Lambda\beta \int_0^t \epsilon_{2,1}(w)\mathrm{d}w \,. \end{aligned} \tag{4.10}$$

Taking the time derivative of the Lyapunov functions yields

$$\begin{aligned}\dot{V} = & s_1^T D_1 \dot{s}_1 + \frac{1}{2} s_1^T \dot{D}_1 s_1 + \tilde{\Theta}_1^T \Gamma_1^{-1} \dot{\tilde{\Theta}}_1 + \epsilon_{1,2}^T K_\epsilon \dot{\epsilon}_{1,2} \\ & + s_2^T D_1 \dot{s}_+ \frac{1}{2} s_2^T \dot{D}_2 s_2 + \tilde{\Theta}_2^T \Gamma_2^{-1} \dot{\tilde{\Theta}}_2 + \epsilon_{2,1}^T K_\epsilon \dot{\epsilon}_{2,1} \\ & + \epsilon_{1,2}^T K_\epsilon \Lambda \beta \int_0^t \epsilon_{1,2}(w) \mathrm{d}w + \epsilon_{2,1}^T K_\epsilon \Lambda \beta \int_0^t \epsilon_{2,1}(w) \mathrm{d}w \,. \end{aligned} \tag{4.11}$$

Multiplying both sides of Equation (4.7) by s_i and substituting the resulting equations to (4.11) results in

$$\begin{aligned}\dot{V} = & - s_1^T k_d s_1 + s_1 + s_1^T Y_1 \left(x_1, \dot{x}_1, u_1, \dot{u}_1\right) \tilde{\Theta}_1 + \tilde{\Theta}_1^T \Gamma_1^{-1} \dot{\tilde{\Theta}}_1 + \epsilon_{1,2}^T K_\epsilon \dot{\epsilon}_{1,2} \\ & - s_2^T k_d s_2 + s_2 + s_2^T Y_2 \left(x_2, \dot{x}_2, u_2, \dot{u}_2\right) \tilde{\Theta}_2 + \tilde{\Theta}_2^T \Gamma_2^{-1} \dot{\tilde{\Theta}}_2 + \epsilon_{2,1}^T K_\epsilon \dot{\epsilon}_{2,1} \\ & - \left(s_1 - s_2\right)^T K_\epsilon \epsilon_{1,2} - \left(s_2 - s_1\right)^T K_\epsilon \epsilon_{2,1} - \epsilon_{1,2}^T K_\epsilon \Lambda \beta \int_0^T \epsilon_{1,2}(w) \mathrm{d}w \\ & + \epsilon_{2,1}^T K_\epsilon \Lambda \beta \int_0^T \epsilon_{2,1}(w) \mathrm{d}w \,. \end{aligned} \tag{4.12}$$

The adaptation equation (4.6) can be rewritten as

$$s_i^T Y_i \tilde{\Theta}_i + \tilde{\Theta} \Gamma_i^{-1} \dot{\tilde{\Theta}}_i = 0 \,. \tag{4.13}$$

From Equations (4.1), (4.2), (4.4), we get

$$\begin{aligned} s_1 - s_2 &= \dot{\epsilon}_{1,2} + \Lambda \epsilon_{1,2} + 2\beta \epsilon_{1,2} + 2\Lambda\beta \int_0^t \epsilon_{1,2}(w) \mathrm{d}w \,, \\ s_2 - s_1 &= \dot{\epsilon}_{2,1} + \Lambda \epsilon_{2,1} + 2\beta \epsilon_{2,1} + 2\Lambda\beta \int_0^t \epsilon_{2,1}(w) \mathrm{d}w \,. \end{aligned} \tag{4.14}$$

This results in,

$$
\begin{aligned}
&(s_1 - s_2)^T K_\epsilon \epsilon_{1,2} + (s_2 - s_1)^T K_\epsilon \epsilon_{2,1} \\
&= \epsilon_{1,2} K_\epsilon \dot{\epsilon}_{1,2} + \epsilon_{1,2}^T K_\epsilon \Lambda \epsilon_{1,2} + \epsilon_{2,1} K_\epsilon \dot{\epsilon}_{2,1} + \epsilon_{2,1}^T K_\epsilon \Lambda \epsilon_{2,1} \\
&\quad + 2\left(\epsilon_{1,2}^T K_\epsilon \beta \epsilon_{1,2} - \epsilon_{1,2}^T K_\epsilon \beta \epsilon_{2,1}\right) + 2\left(\epsilon_{2,1}^T K_\epsilon \beta \epsilon_{2,1} - \epsilon_{2,1}^T K_\epsilon \beta \epsilon_{1,2}\right) \\
&\quad + 2\left(\epsilon_{1,2}^T K_\epsilon \Lambda \beta\right) \int_0^t \epsilon_{1,2}(w)\mathrm{d}w + 2\left(\epsilon_{2,1}^T K_\epsilon \Lambda \beta\right) \int_0^t \epsilon_{2,1}(w)\mathrm{d}w \\
&\quad - \epsilon_{1,2} K_\epsilon \Lambda \beta \int_0^t \epsilon_{2,1}(w)\mathrm{d}w - \epsilon_{2,1} K_\epsilon \Lambda \beta \int_0^t \epsilon_{1,2}(w)\mathrm{d}w \,. \qquad (4.15)
\end{aligned}
$$

Substituting in (4.13) and (4.15) into (4.12) yields

$$
\begin{aligned}
\dot{V} &= -s_1^T K_d s_1 - \epsilon_{1,2}^T K_\epsilon \Lambda \epsilon_{1,2} - s_1^T K_d s_1 - \epsilon_{1,2}^T K_\epsilon \Lambda \epsilon_{1,2} \\
&\quad - (\epsilon_{1,2} - \epsilon_{2,1})^T K_\epsilon \beta (\epsilon_{1,2} - \epsilon_{2,1}) - (\epsilon_{2,1} - \epsilon_{1,2})^T K_\epsilon \beta (\epsilon_{2,1} - \epsilon_{1,2}) \qquad (4.16) \\
&\leq 0 \,.
\end{aligned}
$$

Hence, the input torques and the adaptation law guarantees that the derivative of the Lyapunov function is always negative. This means that the system will tend to the sliding surface defined in (4.4) and will stay there. This ensures that the position errors $\tilde{x}^{(i)}$ and the synchronization errors e_i will converge to zero as time, t, approaches infinity. This control system is shown in the block diagram in Figure 4.1. If K_ϵ and β are chosen to be 0, this controller effectively reduces to the adaptive task controller proposed in Chapter 3.

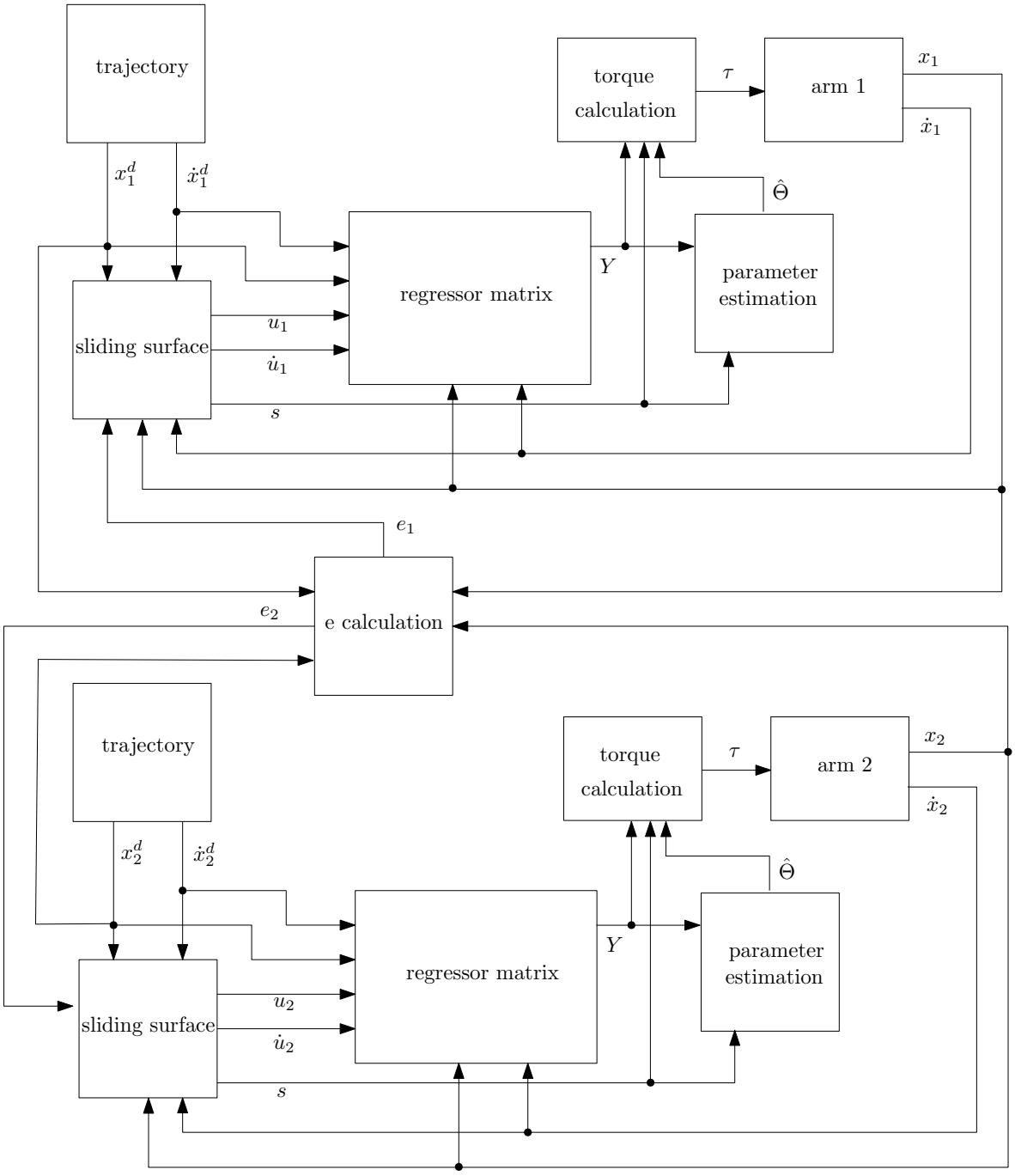

Figure 4.1: Block diagram for the adaptive synchronous controller.

4.3 Simulation and Performance Analysis

This controller design is simulated with the dual planar manipulator setup shown in Figure 4.2 with the parameters given in Table 4.1.

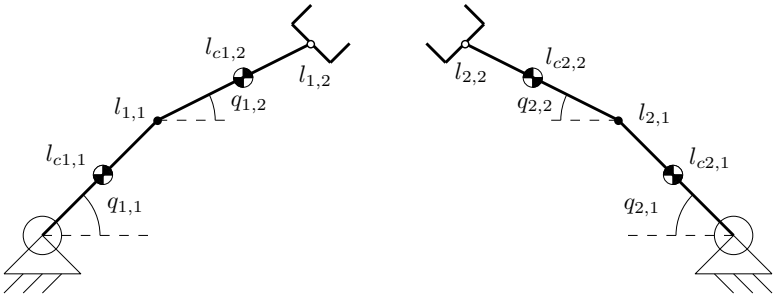

Figure 4.2: System consisting of two planar manipulators with two links.

Table 4.1: Parameters of dual planar arm system.

Parameter	**Description**	**Man. 1**	**Man. 2**	**Unit**
l_1	Length of link 1	1.0	1.0	m
l_{c1}	Length of centre of mass of link 1	0.5	0.5	m
m_1	Mass of link 1	1.0	1.0	kg
I_1	Moment of inertia of link 1	0.12	0.12	$\frac{kg}{m^2}$
l_2	Length of link 2	1.2	2.4	m
l_{c2}	Length of centre of mass of link 2	0.6	0.6	m
m_2	Mass of link 2	2	2	kg
I_2	Moment of inertia of link 2	0.25	0.50	$\frac{kg}{m^2}$

The desired trajectories for each manipulator are calculated using Equation (3.30), the same way as the adaptive task controller simulation outlined in the previous chapter. For manipulator 1, the trajectory parameters given in Equation (4.17) are plotted in Figures 4.3 and 4.4 for task and joint positions respectively. Similarly, for manipulator 2, the trajectory parameters given in Equation (4.18) are plotted in Figures 4.5 and 4.6 for task and joint positions respectively.

$$
\begin{aligned}
t_f &= 10 \\
x_d^{(1)}(0) &= \begin{bmatrix} 0.298 & 0.467 \end{bmatrix}^T \\
x_d^{(1)}(t_f) &= \begin{bmatrix} -0.023 & 0.532 \end{bmatrix}^T
\end{aligned} \tag{4.17}
$$

$$
\begin{aligned}
t_f &= 10 \\
x_d^{(2)}(0) &= \begin{bmatrix} 0.594 & 0.078 \end{bmatrix}^T \\
x_d^{(2)}(t_f) &= \begin{bmatrix} -0.115 & 0.520 \end{bmatrix}^T
\end{aligned} \tag{4.18}
$$

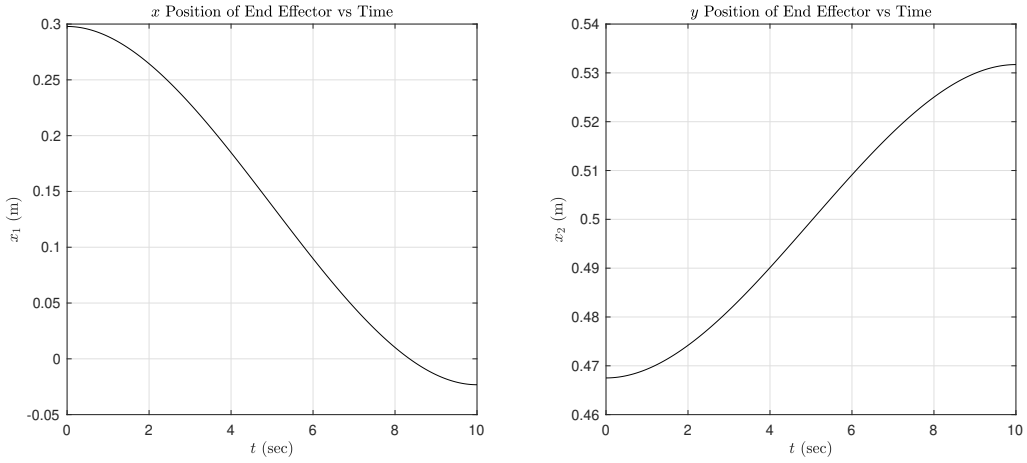

Figure 4.3: The desired end effector trajectory for manipulator 1 for the adaptive synchronous controller simulation.

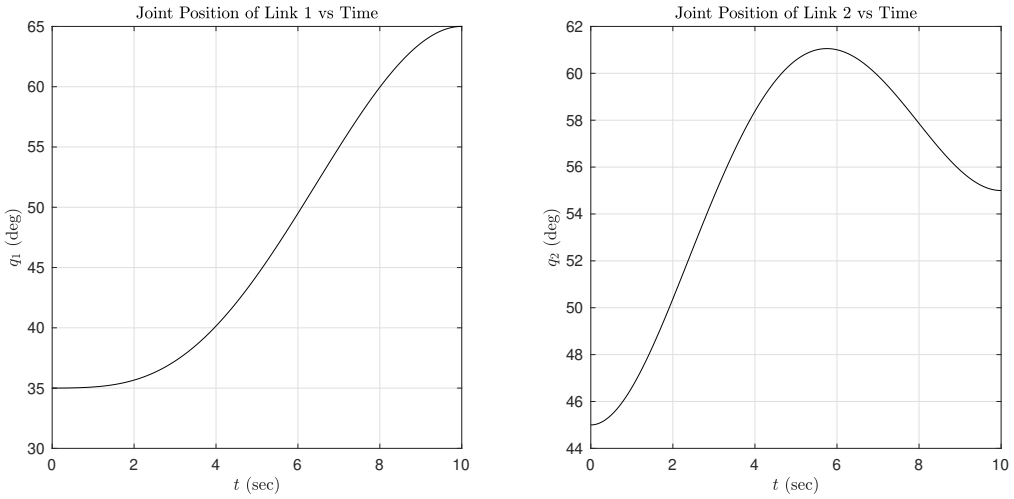

Figure 4.4: The desired joint positions for manipulator 1 for the adaptive synchronous controller simulation.

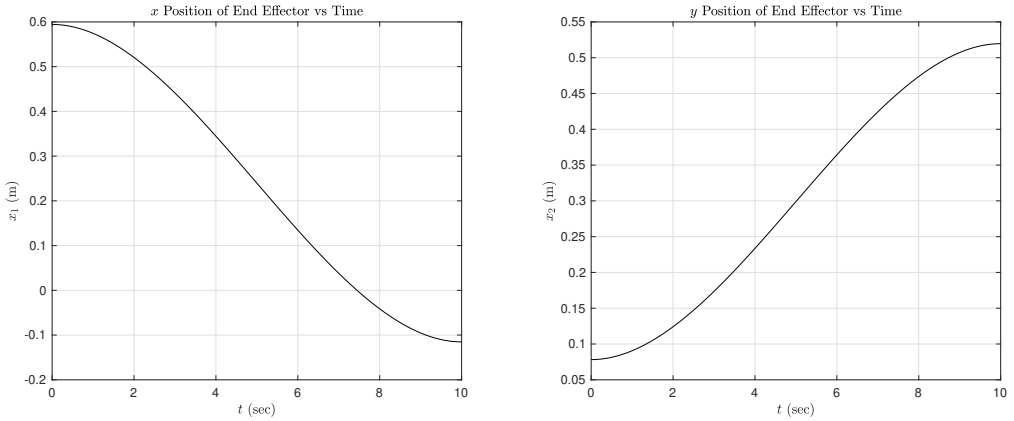

Figure 4.5: The desired end effector trajectory for manipulator 2 for the adaptive synchronous controller simulation.

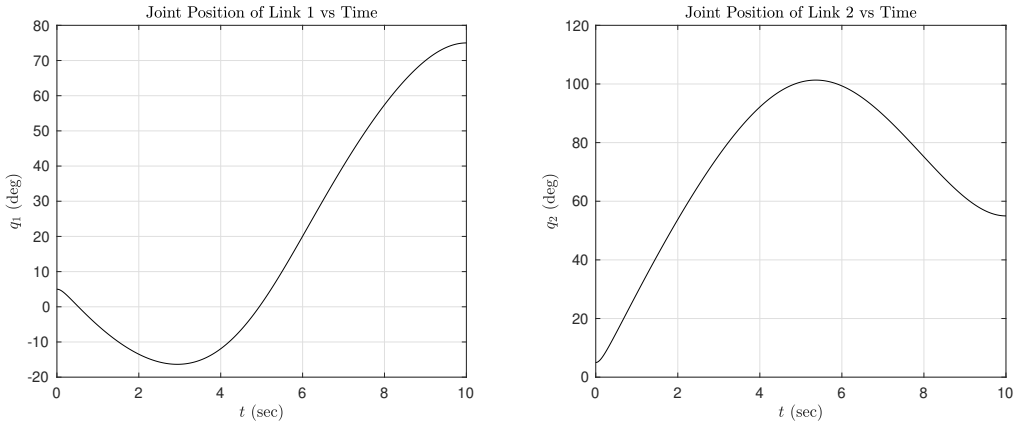

Figure 4.6: The desired joint positions for manipulator 2 for the adaptive synchronous controller simulation.

The design parameter choices are given in Equation (4.19).

$$\begin{aligned} \Lambda &= \begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix} \\ \Gamma &= \begin{bmatrix} 0.03 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \\ K_d &= \begin{bmatrix} 800 & 0 \\ 0 & 800 \end{bmatrix} \\ K_\epsilon &= \begin{bmatrix} 800 & 0 \\ 0 & 800 \end{bmatrix} \\ \beta &= \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \end{aligned} \tag{4.19}$$

The actual and desired positions of the end effector of manipulator 1 and manipulator 2 are shown in Figures 4.7 and 4.8 respectively. After a few seconds of settling time, the actual and desired trajectories are more or less identical. This shows that the synchronous adaptive controller is as effective as the adaptive task controller when it comes to individual trajectory tracking. Looking into the root mean square error (RMSE) of the end effector positions given in Table 4.2 leads to the same conclusion. Two seconds into the motion,

the RMSE is less than a millimeter except the x position of the end effector of manipulator 2. After six seconds that drops below a millimeter as well.

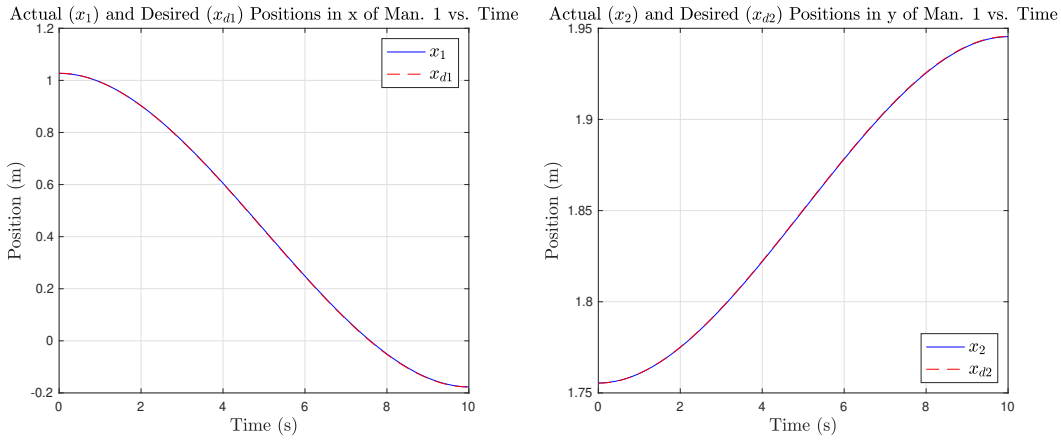

Figure 4.7: The actual and desired positions of the end effector of manipulator 1 in x and y with the adaptive synchronous controller.

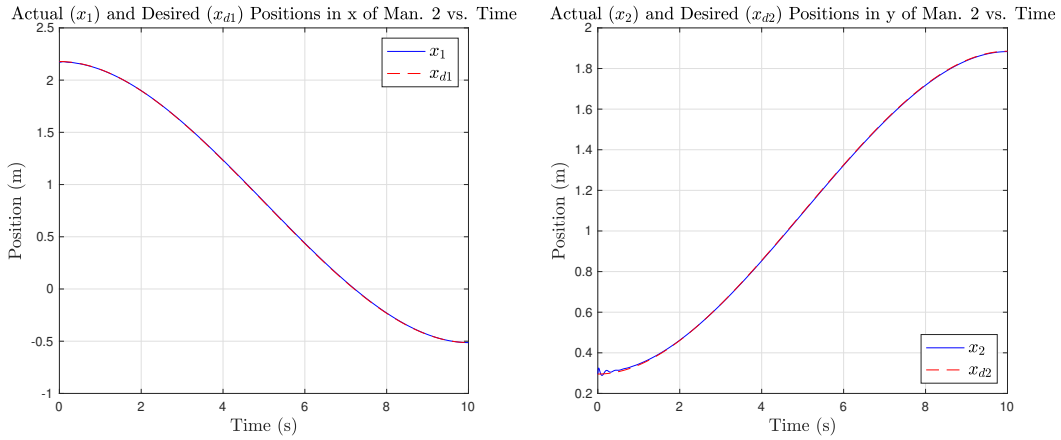

Figure 4.8: The actual and desired positions in x and y of the end effector of manipulator 2 with the adaptive synchronous controller.

The parameter estimation for both manipulators is shown in Figure 4.9. The real parameters of the system are $\Theta_1 = \begin{bmatrix}4.98 & 0.97 & 1.2\end{bmatrix}^T$ and $\Theta_2 = \begin{bmatrix}6.31 & 1.94 & 2.4\end{bmatrix}^T$ for manipulator 1 and manipulator 2 respectively. The estimated values are not converging

Table 4.2: The root mean square error of the end effector positions of the manipulators controlled by the adaptive synchronous controller.

Parameter	**RMSE**	**RMSE After 1s**	**RMSE After 2s**	**RMSE After 6s**
$\tilde{x}_1^{(1)}$	7.072×10^{-4} m	7.022×10^{-4} m	6.998×10^{-4} m	5.009×10^{-4} m
$\tilde{x}_2^{(1)}$	1.156×10^{-4} m	1.149×10^{-4} m	1.144×10^{-4} m	8.26×10^{-5} m
$\tilde{x}_1^{(2)}$	0.001 m	0.001 m	0.001 m	8.636×10^{-4} m
$\tilde{x}_2^{(2)}$	0.002 m	0.001 m	9.378×10^{-4} m	7.083×10^{-4} m

to the real values as expected. Similar to joint and task space adaptive controllers, the control design did not aim for actual parameter estimation.

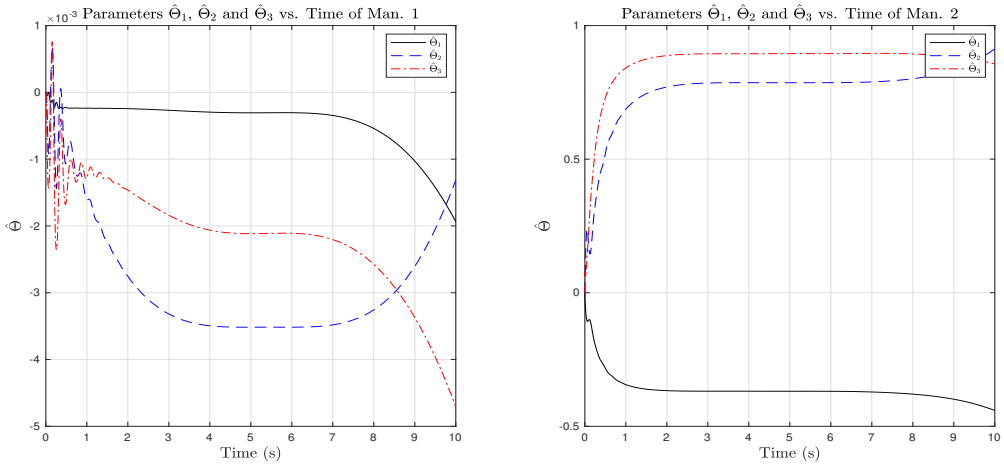

Figure 4.9: Parameter estimation of the adaptive synchronous controller.

The torques applied to the joints of each manipulator are shown in Figure 4.10. After the unstable few seconds in the beginning, the torque values are in a reasonable bound.

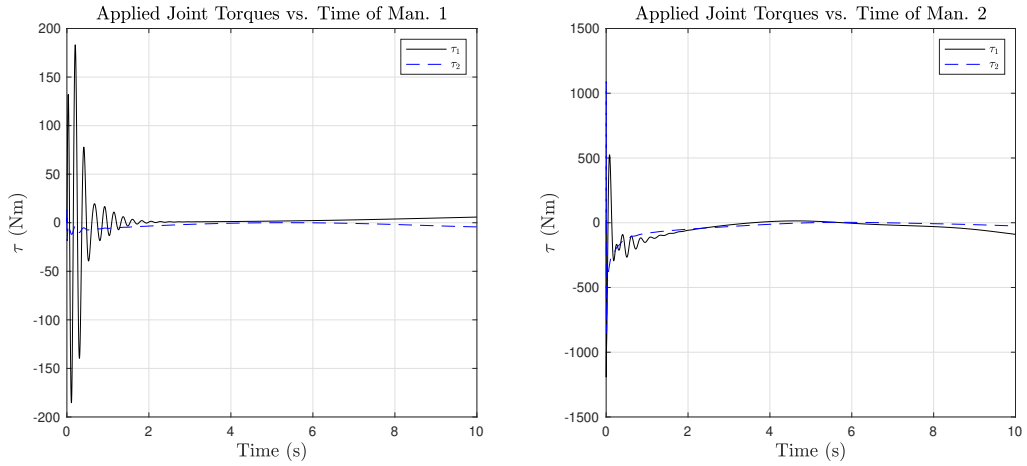

Figure 4.10: Applied torques of the adaptive synchronous controller.

The synchronization errors (e) are shown in Figures 4.11. For comparison, 4.12 demonstrates the synchronization error when the task space controller from Chapter 3 is used. The adaptive synchronous controller provides better performance, the synchronization error practically settles to zero in both axes. Whereas in the adaptive task space controller, even though the synchronization error in x is negligible, it is around 2.5 $[mm]$ in the y axis.

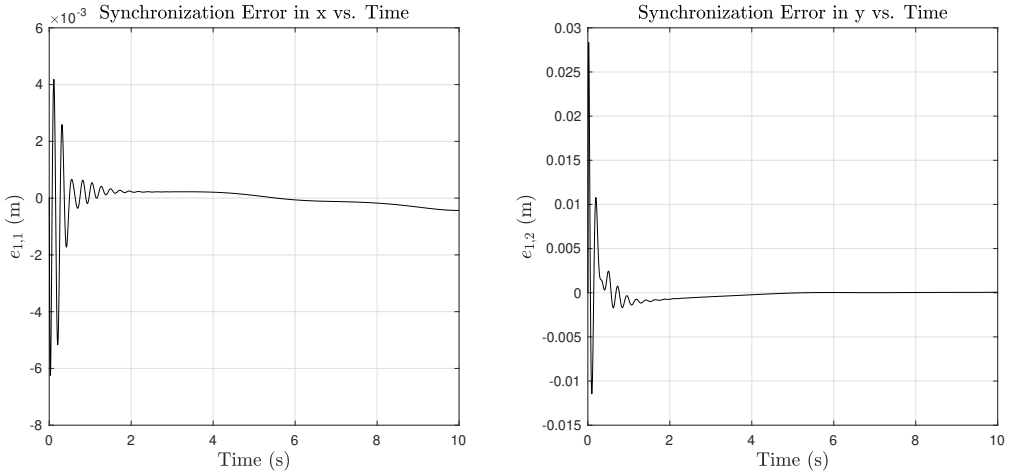

Figure 4.11: Synchronization errors between the manipulators with the adaptive synchronous controller.

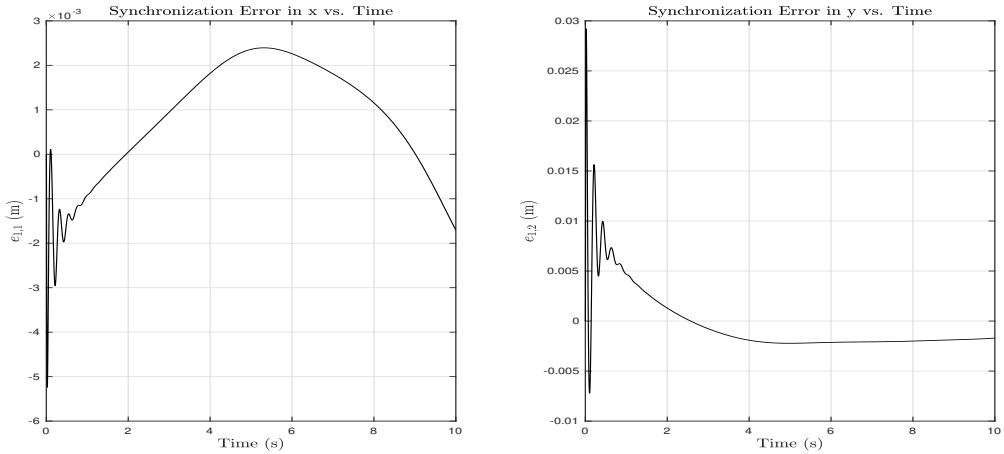

Figure 4.12: Synchronization errors between the manipulators with the adaptive task space controller.

4.4 Summary

In this chapter, an adaptive synchronous controller capable of driving two manipulators with unknown system parameters, such as link masses and inertias, along each of their desired trajectories while simultaneously minimizing the difference between their individual tracking errors was presented and validated with a simulation. This controller allows collaborative tasks to be accomplished by a dual manipulator system, which is further analyzed in Section 5.2 with a more realistic simulation. Its real-life implementation details and shortcomings are discussed in Section 5.1.

Chapter 5

Implementation and Case Study

5.1 Implementation Details and Instrumentation

In this section, the requirements for implementing the adaptive synchronous controller developed in Chapter 4 are discussed. The adaptive synchronous controller is a decentralized coupled controller meaning each manipulator is controlled by their own controller but utilize some information from the other manipulator, effectively coupling them. As seen in the block diagram in Figure 4.1, synchronization error that is needed for surface calculation requires desired and actual positions of the end effectors of both manipulators. The desired positions are straightforward to share between the controllers but for actual positions extra work is required. In addition to the end effector positions, the origins of the manipulators and their relative positions are needed for meaningful trajectory generation. The joint origin positions might also be of interest if the lengths of the links are unknown since they are needed for jacobian and forward kinematics calculations. All these locations of interest are indicated with blue dots in Figure 5.1.

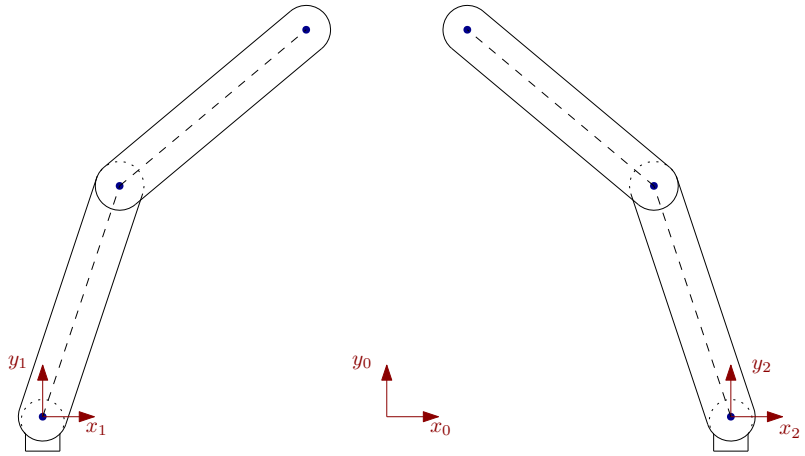

Figure 5.1: Locations of points of interest in a dual manipulator system.

There are several ways to obtain and share position information between the manipulators. All modern manipulators are equipped with encoders in each joint providing joint positions. These joint positions can be converted to cartesian positions with forward kinematics and can be exchanged with networking such as through socket messaging via ethernet. However, this would only provide cartesian positions according to the base coordinate frames (such as frame x_1, y_1 for left manipulator in Figure 5.1) and manual measurement of relative positions between the manipulators would be needed. Also, depending on the manufacturer of the manipulator the method of getting encoder measurements would vary and could introduce further complexity when different manufacturers are preferred for each manipulator. A more robust way to address this problem is to use camera systems. If the manipulators are planar, a standard camera positioned at a distance that can detect both manipulators could suffice. If the tasks require three dimensional manipulation a camera capable of providing depth information, for example Intel® RealSense™ D435 shown in Figure 5.2a, might be required. Regardless of the type of camera used, image processing is required to extract the location of points of interest in the camera frame. With the knowledge of the location and the intrinsic properties of the camera, these positions can be converted to the origin frame. Many open source libraries, such as OpenCV, allow image processing to be done with relative ease. Another option would be to use a motion capture camera such as Vicon® Vera shown in Figure 5.2b. Motion capture cameras allow direct position measurement of trackers put on points of interest according to the origin coordinate frame determined during calibration. This negates the need for developing image processing algorithms. Regardless of the method chosen for communicating the end effector positions, there will be delays in transmitting the information as it is the case in any

networked system. These delays can decrease the performance of the adaptive synchronous controller since the synchronization error used for calculating the torque input is affected by the delays.

(a) Intel® RealSense™ D435 camera with depth sensing. [37]

(b) Vicon® Vero motion capture camera. [38]

Figure 5.2: Camera options for implementing the adaptive synchronous controller.

The velocity of the end effector is another necessary measurement for this control system. Encoders only provide position data, not velocity. Numerical differentiation can be utilized as it was done in the simulations to get the velocity data from end effector position. However, numerical methods might result in unrealistic spikes that can influence the input torques and render the control system unstable. A low pass filter can be used to smoothen the velocity data and avoid instabilities.

Another source of instability could stem from the inherit jitteriness of sliding mode controllers. The inputs of sliding mode controllers, in the case of manipulators the torques, include the surface definition in them as seen in Equations (3.22) and (4.5). In turn, the definition of the surfaces in a tracking problem, Equations (3.1) and (4.4), contain errors in tracking ($\tilde{x}$ and e) which changes sign depending on under or overshooting the target trajectory. In an ideal setting, the controller derives the system to slide on the surface and it does not leave the surface once it reaches there. However in reality, due to discretization and granularity of control inputs and sensor readings, the system never slides on the surface. It rather keeps missing the surface and changing directions. This results in chatter in the control input. Chatter or jitteriness is undesirable since it creates many inefficiencies, such as heat and wear, and non-smooth motion. The chatter can be avoided by introducing a boundary layer around the surface [7], effectively thickening the surface. This is analogous to introducing a low pass filter to the surface.

5.2 Case Study: Manipulating a Common Object

5.2.1 Problem Setting

In this section the adaptive synchronous controller developed in Chapter 4 is applied to a realistic scenario as opposed to the simulations to further validate the performance of the previous controllers. In an automated kitchen application with two planar manipulators, there might be many tasks that require the collaboration of the manipulators. One of these tasks could be moving a heavy pot that is hard to grasp by a single manipulator. Such scenario is depicted in Figure 5.3.

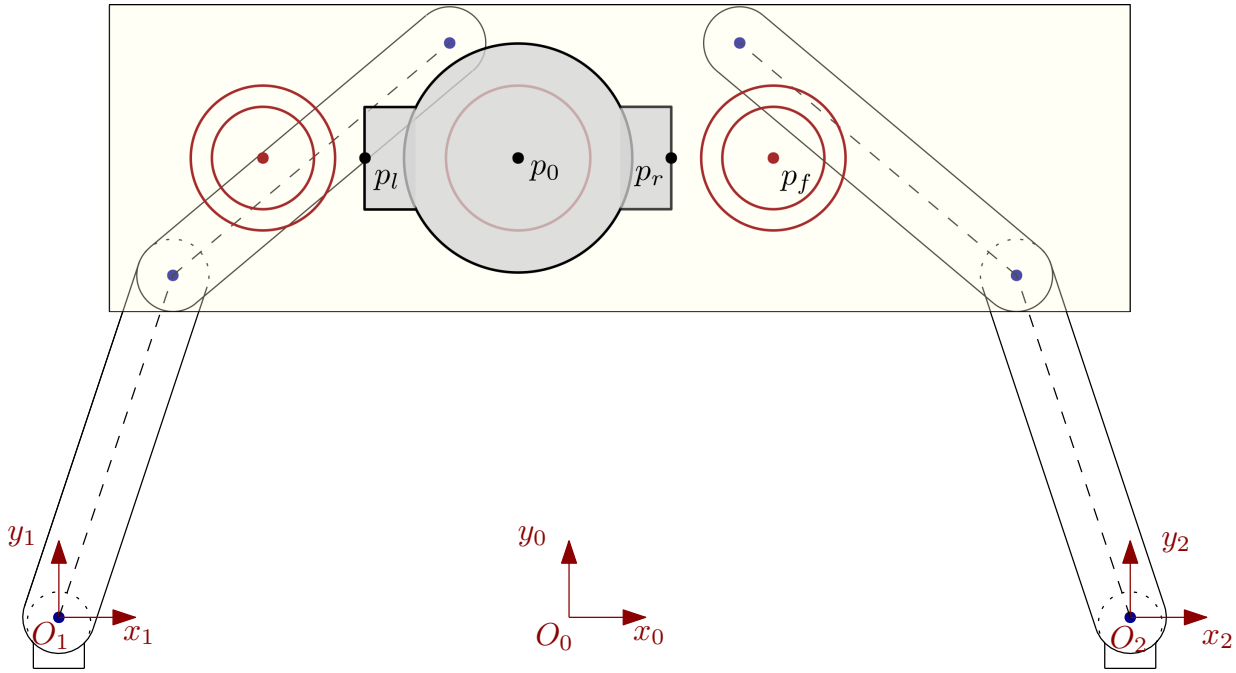

Figure 5.3: Setup for case study consisting of two planar arms in a kitchen setting.

The task is to move the pot from the heater centered at p_0 to the heater with less intensity centered at p_f to simmer the food. To accomplish the task, manipulator with its origin located at O_1 will grasp the pot at point p_l whereas the manipulator with its origin located at O_2 will grasp it at point p_r. It is assumed that the manipulators are able to rigidly grasp the pot at a single point. The coordinates of the aforementioned points according to the coordinate frame O_0, x_0, y_0 are summarized in Table 5.1.

Table 5.1: Parameters of a planar manipulator with two revolute joints.

Point	**x (m)**	**y (m)**
p_0	-0.5	0.4
p_f	0.2	0.4
p_l	-0.15	0.4
p_r	0.5	0.4
O_1	-0.4	0
O_2	0.4	0

The manipulators are modeled after the Panda arm, shown in Figure 5.4a, manufactured by Franka Emika. When only joints 2 and 4 are allowed to move, the panda manipulator becomes a planar one similar to the one considered in this case study as evident from Figure 5.4b.

(a) Panda arm.

(b) Panda arm with length information.

Figure 5.4: Panda manipulator manufactured by Franka Emika [1].

The dynamic properties of the Panda arm are obtained from [39] and with the assumption that the load of 6 $[kg]$ equally distributed to both manipulators, the parameters of the manipulators are calculated and presented in Table 5.2.

Table 5.2: Parameters of the manipulators for the case study.

Parameter	**Description**	**Value**	**Unit**
l_1	Length of link 1	0.316	m
l_{c1}	Length of centre of mass of link 1	0.158	m
m_1	Mass of link 1	3.28	kg
I_1	Moment of inertia of link 1	0.014	$\frac{kg}{m^2}$
l_2	Length of link 2	0.384	m
l_{c2}	Length of centre of mass of link 2	0.192	m
m_2	Mass of link 2	6.6	kg
I_2	Moment of inertia of link 2	0.035	$\frac{kg}{m^2}$

The trajectory of the centre of the pot given in the coordinate frame O_0, x_0, y_0 is calculated with the same method as the trajectory generation for task space adaptive controller outlined in Section 3.3. The initial and the final positions were given in Table 5.1, and $t_f = 5$ seconds.

$$\begin{aligned} x_{pot} &= -0.004t^3 + 0.03t^2 - 0.05 \\ y_{pot} &= 0.4 \end{aligned} \tag{5.1}$$

The trajectory of the end effector of manipulator 1 according to the coordinate frame O_1, x_1, y_1 is found utilizing Equation (5.1) and geometry.

$$\begin{aligned} x_{traj1} &= -0.004t^3 + 0.03t^2 + 0.25 \\ y_{traj1} &= 0.4 \end{aligned} \tag{5.2}$$

The trajectory of the end effector of manipulator 2 according to the coordinate frame O_2, x_2, y_2 calculated in the same way as the other manipulator.

$$
\begin{aligned}
x_{traj2} &= -0.004t^3 + 0.03t^2 - 0.35 \\
y_{traj2} &= 0.4
\end{aligned} \tag{5.3}
$$

This scenario is simulated with the controller developed in Chapter 4, with no prior knowledge of system parameters ($\hat{\Theta} = 0$) with the following design parameters:

$$
\begin{aligned}
\Lambda &= \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \\
\Gamma &= \begin{bmatrix} 0.03 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}, \\
K_d &= \begin{bmatrix} 80 & 0 \\ 0 & 80 \end{bmatrix}, \\
K_\epsilon &= \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \\
\beta &= \begin{bmatrix} 75 & 0 \\ 0 & 75 \end{bmatrix}.
\end{aligned} \tag{5.4}
$$

5.2.2 Results and Discussion

The simulation is also repeated with synchronization variables turned off (ie. $k_\epsilon = 0$ and $\beta = 0$). In that case, the controller is basically the same as the adaptive task controller. The synchronization error between the manipulators are plotted in Figure 5.5. The error is around $-2\,[cm]$ at the two second mark. With that much of difference in synchronization, it is safe to assume that the task of transporting the pot would fail since one of the manipulators would be smashing into the pot, causing a great amount of force to be exerted on the other. Even if this were not to cause a problem, at around four seconds in, the error is around $2\,[cm]$. This would mean that one of the manipulators would lose grasp of the pot.

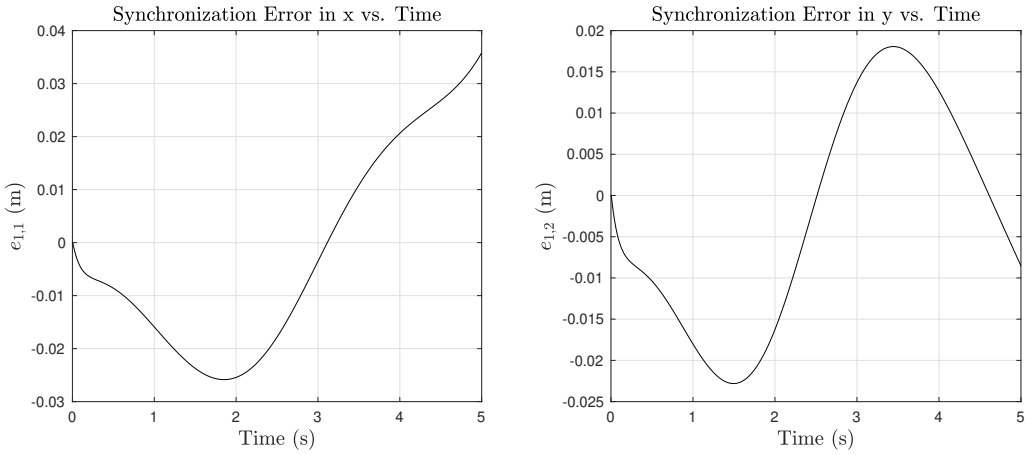

Figure 5.5: Synchronization error between the manipulators while moving the pot with adaptive task controller.

In contrast the synchronization error with the adaptive synchronous controller is less than a millimeter throughout the task. This is the same magnitude as the repeatability of the Panda manipulator. It is clear that the adaptive synchronous controller is needed for a task like this.

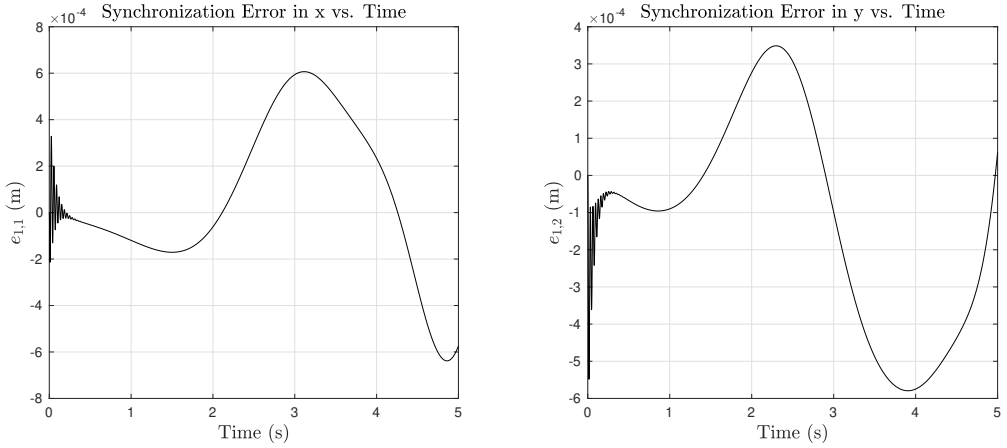

Figure 5.6: Synchronization error between the manipulators while moving the pot with adaptive synchronous controller.

To analyze how well the task was completed, the position errors of the pot location in both x and y are plotted in Figures 5.7 and 5.8 respectively. The greatest error is around

1.6 cm which would not cause an issue since the heaters are large enough to accommodate such small errors.

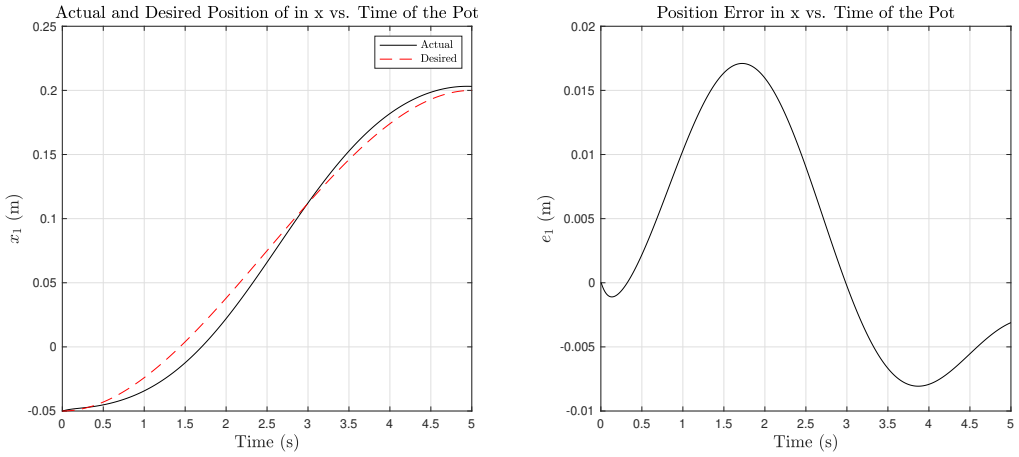

Figure 5.7: Position error in the x of the object being manipulated.

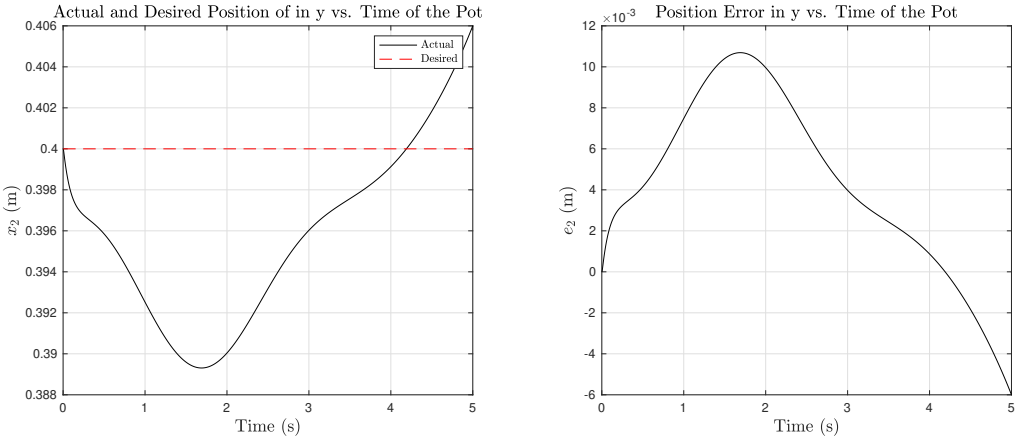

Figure 5.8: Position error in the y of the object being manipulated.

Chapter 6

Conclusion

6.1 Summary

In this thesis, an adaptive synchronous controller based on position data for a dual manipulator system to realize collaborative tasks that involve objects with unknown properties is studied. The development of the aforementioned controller is presented incrementally by building on adaptive controllers in joint and task space for single manipulators. While the joint space adaptive control design provides a good performance on trajectory tracking, its applications are limited due to the fact that real-life tasks usually involve the end effector position, which is prescribed in the task space. The task space adaptive controller overcomes this shortcoming. However, when used in a dual manipulator system, the transient period where the system parameters converge results in great synchronization errors, which is defined as the difference between individual trajectory errors of each manipulator. This synchronization error renders the usage of the adaptive controller in collaborative dual manipulator systems impractical since individual trajectories are defined in such a way that a significant deviation in synchronization could cause the task to fail. An example of a collaborative task of this nature can be the assembly of a pin into a housing in which each item is held by two separate manipulators. A substantial synchronization error during trajectory execution could cause a catastrophic failure such as the pin crashing into the housing. The adaptive synchronous controller is developed to minimize these synchronization errors by involving them in the controller design. The trajectory error of each manipulator is communicated to one another and this information is utilized in the control input (i.e. torque) calculations. This allows the controller to guarantee that the trajectory errors along with the synchronization errors approach to zero eventually. The

effectiveness of this approach is demonstrated with a simulation in which two manipulators carry a heavy load together. While the adaptive task space controller fails causing one manipulator to lose grasp, the adaptive synchronous controller is able to keep the motion of the arms synchronized while each of them track their trajectories successfully.

The main advantage of the adaptive synchronous control design presented in this thesis is that it requires minimal instrumentation to implement compared to other collaborative control designs since it is only based on the position of the end effectors. A vision system such as a camera capable of identifying and tracking the end effector of each manipulator is enough to provide the synchronization error to each manipulator. This allows the application of this controller design to a wide range of industrial manipulators which lack the extra instrumentation, such as force sensors. However, one camera can be easily blocked off by the motion of a manipulator causing the task in hand to fail. An array of cameras positioned around the manipulators solves this problem, but introduces complexity since multiple images need to be analyzed to find the position of the end effectors and these results need to be fused appropriately. Another important implementation detail to consider is the fact that the adaptive synchronous controller is a decentralized design, meaning that there are individual controllers for each manipulator rather than one central controller. The decentralized design makes it possible to use different types or configurations of manipulators without changing the controller design. However, the disadvantage of it is that the communication delays might degrade the performance since the controller of one manipulator needs to transmit its tracking error to another. Furthermore, the inherit jitteriness of sliding mode control design is another concern when it comes to real-life applications. The ever-changing control input due to this jitteriness may cause unacceptable performance in some tasks and may cause rapid degradation of the motors that drive the links of the manipulators.

6.2 Future Work

The presented adaptive synchronous control design performs satisfactorily in simulations. However, the aforementioned implementation considerations needs to be investigated as well. This can be accomplished by including either of the following in the simulations:

- Network delays due to information sharing between the manipulators
- The inherit jitteriness of the sliding mode controllers

or by applying the controller design to a real dual manipulator system. In addition, the functionality of the system is rather limited by the fact that the collaborative task trajectories for each manipulator need to be prescribed and is not dynamic. Integrating visual servoing would improve this by allowing dynamic objects to be used in the collaborative tasks and would take advantage of the already existing vision system.

References

[1] "Franka Emika." https://www.franka.de. Accessed: 2020-04-30.

[2] M. W. Spong, S. Hutchingson, and M. Vidyasagar, *Robot Modeling and Control*. John Wiley & Sons, 2006.

[3] K. M. Lynch and F. C. Park, *Modern Robotics*. Cambridge University Press, 2017.

[4] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics Modelling, Planning and Control*. Cambridge University Press, 2010.

[5] J. J. Craig, *Introduction to Robotics*. Prentice-Hall, 2005.

[6] H. K. Khalil, *Nonlinear Systems*. Pearson, 2001.

[7] J.-J. Slotine and W. Li, *Applied Nonlinear Control*. Prentice-Hall, 1991.

[8] B. Markiewicz, "Analysis of the computed torque drive method and comparison with conventional position servo for a computer-controlled manipulator," tech. rep., Jet Propulsion Laboratory, March 1973.

[9] R. P. Paul, "Modeling, trajectory calculation, and servoing of a computer controlled arm," tech. rep., Standford University Artifical Intelligence Laboratory, November 1972.

[10] V. S. Cvetkovic and M. Vukobratovic, "One robust, dynamic control algorithm for manipulation systems," *International Journal of Robotics Research*, 1982.

[11] C. Canudas de Wit, *Theory of Robot Control*. Springer Verlag, 1996.

[12] J.-J. Slotine, "The robust control of robot manipulators," *International Journal of Robotics Research*, 1985.

[13] J. J. Craig, *Adaptive Control of Mechanical Manipulators*. Addison-Wesley, 1988.

[14] R. Horowitz and M. Tomizuka, "An adaptive control scheme for mechanical manipulators - compensation of nonlinearities and decoupling control," *ASME Journal of Dynamic Systems*, 1983.

[15] H.-K. Lee and M. J. Chung, "Adaptive controller of a master-slave system for transparent teleoperation," *Journal of Robotic Systems*, vol. 15, no. 8, p. 465–475, 1998.

[16] W. Gueaieb, S. Al-Sharhan, and M. Bolic, "Robust computationally efficient control of cooperative closed-chain manipulators with uncertain dynamics," *Automatica*, vol. 43, no. 5, p. 842–851, 2007.

[17] D. Sun and J. Mills, "Adaptive synchronized control for coordination of multirobot assembly tasks," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, p. 498–510, 2002.

[18] S.-J. Chung and J.-J. E. Slotine, "Cooperative robot control and concurrent synchronization of lagrangian systems," *IEEE Transactions on Robotics*, 2009.

[19] D. Sun, "Position synchronization of multiple motion axes with adaptive coupling control," *IFAC Proceedings Volumes*, vol. 35, no. 1, p. 313–318, 2002.

[20] A. Rodriguez-Angeles and H. Nijmeijer, "Mutual synchronization of robots via estimated state feedback: A cooperative approach," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, p. 542–554, 2004.

[21] H. Wang and Y. Xie, "Passivity based adaptive jacobian tracking for free-floating space manipulators without using spacecraft acceleration," *Automatica*, vol. 45, no. 6, p. 1510–1517, 2009.

[22] L. Cheng, Z.-G. Hou, M. Tan, D. Liu, and A.-M. Zou, "Multi-agent based adaptive consensus control for multiple manipulators with kinematic uncertainties," *2008 IEEE International Symposium on Intelligent Control*, 2008.

[23] C. Cheah, C. Liu, and J. Slotine, "Adaptive jacobian tracking control of robots with uncertainties in kinematic, dynamic and actuator models," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, p. 1024–1029, 2006.

[24] H. Wang, "Passivity based synchronization for networked robotic systems with uncertain kinematics and dynamics," *Automatica*, vol. 49, no. 3, p. 755–761, 2013.

[25] Y.-C. Liu and N. Chopra, "Controlled synchronization of heterogeneous robotic manipulators in the task space," *IEEE Transactions on Robotics*, vol. 28, no. 1, p. 268–275, 2012.

[26] K. Hamajima and M. Kakikura, "Planning strategy for task of unfolding clothes," *Robotics and Autonomous Systems*, vol. 32, no. 2-3, p. 145–152, 2000.

[27] K. Salleh, H. Seki, Y. Kamiya, and M. Hikizu, "Inchworm robot grippers in clothes manipulation — optimizing the tracing algorithm," *2007 International Conference on Intelligent and Advanced Systems*, p. 1051–1055, 2007.

[28] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," *2010 IEEE International Conference on Robotics and Automation*, p. 2308–2315, 2010.

[29] K. Yamazaki, Y. Watanabe, K. Nagahama, K. Okada, and M. Inaba, "Recognition and manipulation integration for a daily assistive robot working on kitchen environments," *2010 IEEE International Conference on Robotics and Biomimetics*, p. 196–201, 2010.

[30] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mosenlechner, D. Pangercic, T. Ruhr, and M. Tenorth, "Robotic roommates making pancakes," *2011 11th IEEE-RAS International Conference on Humanoid Robots*, 2011.

[31] Y. Yamada, S. Nagamatsu, and Y. Sato, "Development of multi-arm robots for automobile assembly," *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, p. 2224–2229, May 1995.

[32] J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi, "Recognizing assembly tasks through human demonstration," *The International Journal of Robotics Research*, vol. 26, no. 7, p. 641–659, 2007.

[33] R. L. A. Shauri and K. Nonami, "Assembly manipulation of small objects by dual-arm manipulator," *Assembly Automation*, vol. 31, no. 3, p. 263–274, 2011.

[34] K. Kosuge, H. Yoshida, T. Fukuda, M. Sakai, and K. Kanitani, "Manipulation of a flexible object by dual manipulators," vol. 1, pp. 318–323 vol.1, 1995.

[35] Y. F. Zheng and M. Z. Chen, "Trajectory planning for two manipulators to deform flexible beams," *Robotics and Autonomous Systems*, vol. 12, no. 1-2, p. 55–67, 1994.

[36] J.-J. Slotine and W. Li, "Adaptive manipulator control: A case study," *IEEE Transactions on Automatic Control*, 1988.

[37] "Intel Realsense Depth Camera D435." https://www.intelrealsense.com/depth-camera-d435. Accessed: 2020-04-30.

[38] "Vicon Motion Capture Systems." https://www.vicon.com. Accessed: 2020-04-30.

[39] C. Gaz, M. Cognetti, A. Oliva, P. R. Giordano, and A. D. Luca, "Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, p. 4147–4154, 2019.

[40] F. L. Lewis, D. M. Dawson, and T. A. Chaouki, *Introduction to Robotics*. Marcel Dekker, 2004.

[41] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. John Wiley & Sons, 1995.

[42] H. Wang, "Task-space synchronization of networked robotic systems with uncertain kinematics and dynamics," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, p. 3169–3174, 2013.

[43] H. Iwata and S. Sugano, "Design of human symbiotic robot twendy-one," *2009 IEEE International Conference on Robotics and Automation*, p. 3294–3300, 2009.