MASTER THESIS

EXTERNAL FORCE ESTIMATION ON A NON-LINEAR COMPLIANT 2DOF MANIPULATOR SYSTEM

T.H.J. VAN DER HOLST

08-04-2022

UNIVERSITY OF TWENTE.

ABSTRACT

In some applications, the external forces acting on a system might be required (e.g. for the purposes of force control). It might, however, be difficult to implement a force sensor on the interaction surface or it may be too costly to do so. In this report, estimation of the external forces acting on a 2DOF manipulator with flexure hinges is considered. The estimation of the external forces is based on the system dynamics and the actuator inputs of the system. The joints of the manipulator are based on flexures, which reduce the uncertainties due to friction and hysteresis. Nevertheless, the estimation process is complicated by the non-linear nature of the 2DOF manipulator. An estimation method is developed based on linearized Kalman filtering, whereby a linearized dynamic model of the system is derived and applied to a Kalman filter, and the remaining Kalman filter parameters are set based on the expected disturbances caused by errors in the identification of the system's parameters.

The method is first evaluated on a double pendulum model through simulations. The double pendulum is a simpler system in comparison, but has key similarities with the 2DOF manipulator. From testing simulations, it is found that the accuracy of the model used by the linearized Kalman filter has a significant influence on the quality of the estimates. Any difference in the dynamic behavior of the real system with respect to the model will result in fictitious forces being estimated. This cannot be avoided by considering the modelling errors when setting the parameters of the Kalman filter. Despite the importance of the model's accuracy, neglecting the non-linear behavior in the model does not have a significant influence on the estimated external forces.

The method is then tested on an existing setup of the 2DOF manipulator. The results are agreeable, although an estimation error appears to be present based on the sensor measurements. The magnitude of the estimation errors are in general less than 10% of the magnitude of the applied external forces. Estimation errors also occur when only actuator forces are applied, again with a magnitude of at most 10% of the actuator forces. It is observed that disturbances not measured by the sensors, such as friction and hysteresis, act on the system. It is difficult to determine exactly to what extent the estimation errors are caused by these unmeasured disturbances using the current setup.

CONTENTS

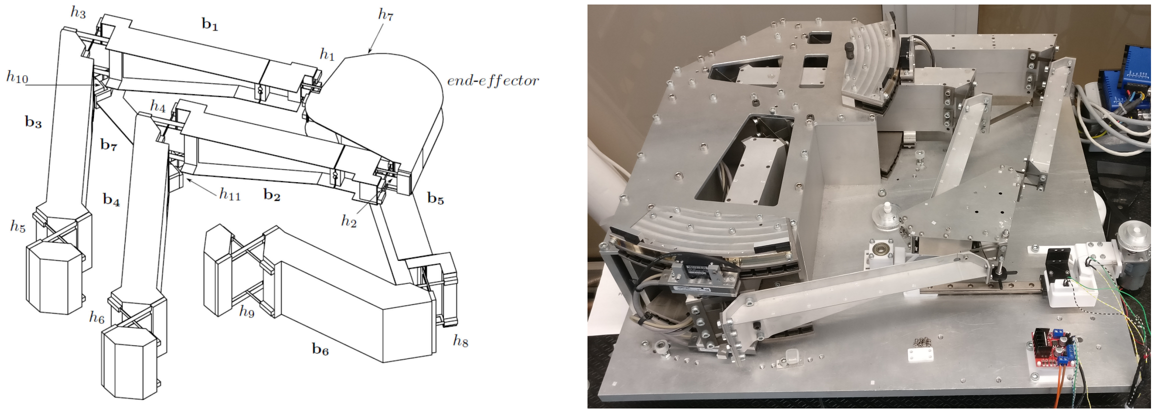
1 INTRODUCTION

1.1 Problem definition

To control the behaviour of various complex systems accurately, the external forces acting on these systems are required. These external forces can generally be measured with force sensors. However, it might be difficult to implement a force sensor on the interaction surface of the system or it may be too costly to do so. Consequently, estimation of the external forces on a system might be preferred over measuring them with force sensors. In the current study, estimation of the external forces acting on the end-effector of a 2DOF manipulator with flexure hinges is considered. The 2DOF manipulator is shown in figure 1.1.

The estimation of the external forces can be based on the system dynamics, the actuator inputs of the system, and the displacement measured by encoders. Since care was taken in the design process of the 2DOF manipulator to reduce the effects of uncertainties in the dynamic behavior of the system, estimating the external forces based on its dynamic behavior may be promising. An essential part of the design is the use of flexure hinges, which reduce the effects of friction and hysteresis, and the uncertainties caused by them. Nevertheless, the estimation process is complicated by the non-linear dynamics of the 2DOF manipulator.

This report focuses on developing and testing a method to estimate external forces based on system dynamics, actuator inputs, and measured displacements. The method will be tested on a double pendulum model in simulations, as well as on the real 2DOF manipulator in experiments.



(a) Schematic sketch [1]

(b) Photo

Figure 1.1: 2DOF manipulator

1.2 2DOF manipulator background

External force estimation will be applied to the 2DOF manipulator system depicted in figure 1.1. The conceptual design of this system was developed by Hoitzing [2], and it was improved and detailed further by Folkersma [1]. Relevant sections of their work will be briefly summarized next.

The goal of their work was to design a planar XY positioning mechanism to be used in a vacuum with high accuracy and repeatability. A key part of this process was the use of flexure hinges as opposed to friction-based components (such as bearings). Friction based components may contaminate the surroundings in the vacuum due to the the evaporation of lubricants. Furthermore, these components are subject to non-deterministic effects such as friction, wear, play, and hysteresis. These problems may be overcome by using flexure-based hinges.

The framework of the 2DOF manipulator consists of beams connected by flexure hinges, indicated by b_i and h_i respectively in figure 1.1a. The 2DOF manipulator consists of two parallel arms and a separate single arm. Most parts of the system consist of thin plates, which ensures a lightweight and stiff mechanism. To prevent internal stresses and ensure a deterministic behavior of the system, the mechanism has been designed to be exactly constrained. In their work, this is achieved by applying constraint analyses to the conceptual designs. Together, the arms exactly constrain all DOFs of the end-effector except the translational movement in x- and y-direction.

The single arm with beams b_5 and b_6 is included to prevent requiring to attach an actuator to beams b_1 or b_2 to actuate the system. Instead, the actuators and encoders are attached to beams b_4 and b_6, reducing the negative effect of their masses added to the mechanism. The placement of the actuators and encoders is illustrated in figure 1.2. The encoders measure a (translational) distance along a circular path perpendicular to beams b_4 and b_6 at a distance $r_{enc} = 0.251m$ from hinges h_6 and h_9. Similarly, the actuators exert a translational force along the circular path perpendicular to beams b_4 and b_6 at a distance $r_{act} = 0.235m$. The cables connected to the actuators and encoders are routed such that effects of hysteresis and friction caused by the cables when the system is in motion are minimized.

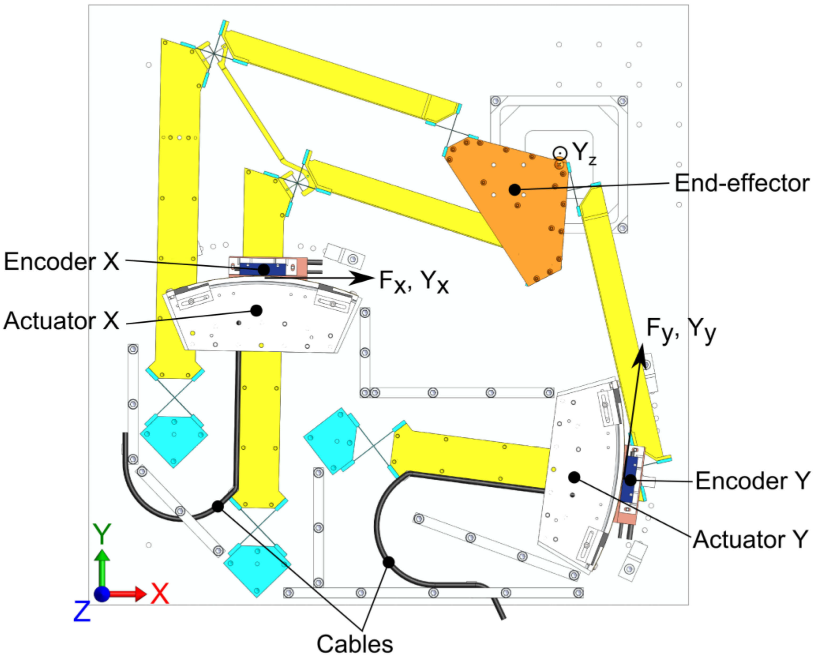


Figure 1.2: Top view of 2DOF manipulator with encoders, actuators and cables [1]

The beams are connected using flexure hinges, indicated with h_i in figure 1.1a. Flexures are compliant elements which can achieve a limited displacement or rotation due to elastic deformation. The advantage of flexures is that they reduce the effects of drawbacks like friction, wear, play and hysteresis compared to other guidance mechanisms. A downside of using flexures, however, is their small range of motion. The flexure hinges used in the 2DOF manipulator are cross-hinges consisting of five leaf springs. An example of such a flexure hinge can be seen in figure 1.4. This constrains all but one of the degrees of freedom, namely the rotation about the z axis, making it function as a hinge. The maximum deflection for each of the individual flexure hinges in the 2DOF manipulator are 20°. The resulting workspace of the end-effector is $100 \times 100mm$.

To be able to validate the estimated external forces, the 2DOF manipulator setup includes a construction with two force sensors. Each sensor is used to measure forces in direction of one of the system's degrees of freedom: one in x- and y-direction as indicated in figure 1.2. The sensors are mounted vertically in a construction secured on a rail. The rail allows the construction with sensors to move through the workspace of the end-effector in y-direction. A small protrusion connected to the end-effector is able to provide contact with the force sensors. In this way, forces exerted on the end-effector via the construction with sensors can be measured. A photo of the sensor setup is shown in figure 1.3.

The exact constraint design of the 2DOF manipulator and the use of flexure hinges decrease non-deterministic effects in its dynamic behavior. Because of its predictable dynamic behavior, applying external force estimation to the 2DOF manipulator is promising. Using the force sensors included in the setup, data can be measured to validate the estimated external forces.

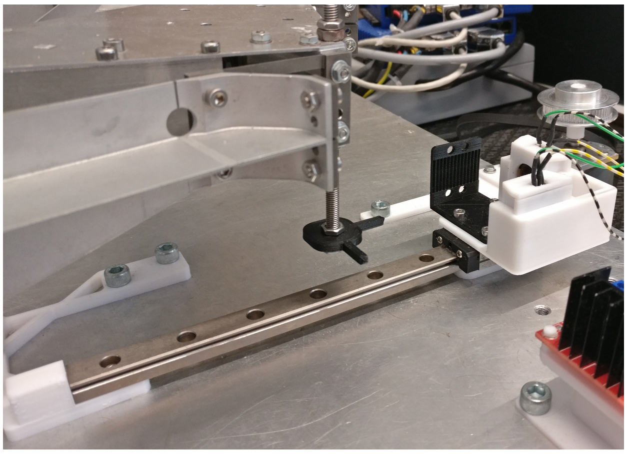


Figure 1.3: Photo of the sensor setup

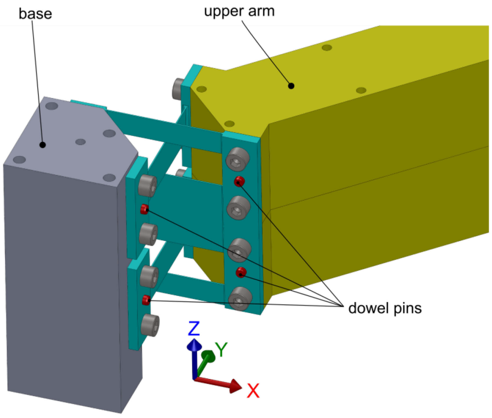


Figure 1.4: View of a cross-hinge [1]

1.3 Report outline

This thesis describes the application of external force estimation to a 2DOF manipulator. Chapter 2 will start with an overview of available methods to estimate external forces acting on a system. One of these methods, the linearized Kalman filter, will be chosen and discussed in more detail. Chapter 3 discusses the testing of the estimation method on a double pendulum in simulations. A model of a double pendulum will be defined, which will be used to test the performance of the chosen force estimation method in different conditions. In chapter 4, the chosen estimation method will be applied to the 2DOF manipulator. Its performance will be tested in experiments on an existing setup.

2 ESTIMATION METHODS

This chapter will discuss the choice of which estimation method will be used to estimate the external forces acting on the 2DOF manipulator. The chapter will start with an overview of the available estimation methods. After that, the chosen method will be discussed in more detail.

2.1 Overview

This section will give an overview of available methods to estimate external forces acting on a system. This overview is based on the work from Chien et al. [3], and Radke and Gao [4]. Chien et al. describe progress in development of state observers over the years, and Radke and Gao show a recent performance comparison of some observer-based force estimation methods. Disturbance observers or state observers are commonly used for external force estimation. State observers are systems that provide an estimate of the internal state of a given real system from measurements of the input and output of the real system. Multiple methods have been developed over the years, which rely on different principles. Different options may be appropriate for applications, depending on the particular problem. The choice of estimation method can depend on the dynamic structure of the plant, the sensors required, the available plant knowledge, and the practical implementation of the observer.

The generalized momentum approach (GMA) [5] is a model-based force estimation method. It features an alternate expression for the inertia term of the equation of motion based on the generalized momentum. As a result, it has the advantage that neither the inverse of the inertia matrix nor acceleration data is required for its implementation. The GMA method is widely used for its intuitive design and reliable performance. However, the method is found to be sensitive to modeling errors in practice. It is found that there may be cases where a false collision is detected, caused by uncertainties in the dynamic model. To deal with this, it is suggested to experimentally determine a threshold torque below which the detected collision torques are ignored. This limits the method's capability of detecting small collisions. Furthermore, the observer gain needs to be high for accurate estimations. This makes the GMA sensitive to measurement noise. Another downside is that this method requires a velocity sensor.

Another method discussed is the extended state observer (ESO) [6]. In the ESO, the external forces are estimated by augmenting the state vector to include the external forces acting on the system as an additional state. The ESO is a model-based method, where the estimated augmented state vector is adjusted based on measurements of position (and possibly velocity) as feedback. One drawback of the ESO compared to GMA is that its calculation involves the inverse of the inertia matrix, which may lead to a significant computational load. The other drawback is that the ESO may introduce a phase lag if the number of (augmented) state variables is too large.

The key principle of these observers is the use of input and output data along with plant information to estimate the states of a system. The mentioned observers do not inherently take the presence of disturbances in the measurement signals into account, however. More modern observers formulate the estimation problem with the disturbances acting on the system in mind. These methods minimize a cost function based on mathematical assumptions about disturbances. One example of such a modern method is the Kalman filter [7], which is model-based as well. It assumes that the disturbances acting on the system are stochastic, zero mean, and Gaussian, with known process and measurement noise covariances. It uses a series of measurements observed over time, including statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone. This is achieved by estimating a joint probability distribution over the variables for each timeframe.

Another modern method is the H_∞ filter. This method also optimizes a cost function based on an assumption about the disturbance. Using infinity norms, the H_∞ estimator minimizes the worst-case energy gain from noise inputs to the estimation error. The estimator is guaranteed to be optimal under a user defined upper bound. The noise acting on the system is unknown, but not necessarily random or stochastic. This is an advantage with respect to Kalman filter, as the disturbances acting on the 2DOF manipulator are not necessarily Gaussian. Disadvantages of the H_∞ filter include the level of mathematical understanding needed to apply them successfully and the need for a good model of the system.

In this report, external force estimation will be achieved by implementing a method based on Kalman filtering. Kalman filtering is preferred since it is a simple yet sophisticated estimation method, since it incorporates the disturbances acting on the system in its estimations. It is relatively simple to implement, and its widespread applicability would allow the method to be applied to other systems as well. For the Kalman filtering based method to work, an accurate dynamic model of the system will be required. Since care was taken in the design process of the 2DOF manipulator that non-deterministic effects in its dynamic behavior are minimized, this requirement seems feasible. One thing to note is that the 'basic' Kalman filter is only applicable to linear systems, while the 2DOF manipulator is a non-linear system. Fortunately, variations of the Kalman filter are available, which are capable of dealing with non-linear systems. In the following sections, the Kalman filter and some of its variations will be discussed in more detail.

2.2 Kalman filtering

This section discusses the working principles of Kalman filtering on the basis of the discrete time Kalman filter. This section is based on the insightful paper by Faragher [8], and more in-depth information about Kalman filters from the book by Crassidis and Junkins [9].

The Kalman filter is a celebrated and popular data fusion algorithm in the field of information processing. It has a wide field of applications, where it can be used to combine multiple sources of uncertain information about a system to smooth noisy data or provide estimates of parameters of interest. A great example of this is where the real state of a mechatronic system is desired to be known, but where the measurements are distorted by noise and the dynamic model of the system is imperfect. The Kalman filter is capable of combining these two types of uncertain information (model and measurements), and finding the state of the system that is most likely based on both types of information.

At its core, the Kalman filter makes use of probability theory. It is assumed that both the measurements and the model are distorted by zero-mean Gaussian white-noise. This is indicated in the truth model as

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{A}_d\boldsymbol{x}_k + \boldsymbol{B}_d\boldsymbol{u}_k + \boldsymbol{\Gamma}_d\boldsymbol{w}_k \qquad && \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_d)\,, \\ \boldsymbol{y}_k &= \boldsymbol{C}_d\boldsymbol{x}_k + \boldsymbol{v}_k && \boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_d)\,, \end{aligned} \tag{2.1}$$

where $\boldsymbol{A}_d$, $\boldsymbol{B}_d$, $\boldsymbol{C}_d$, $\boldsymbol{D}_d$, and $\boldsymbol{\Gamma}_d$ represent discrete-time state-space matrices of the system, $\boldsymbol{x}_k$ the state vector, $\boldsymbol{u}_k$ the input vector, $\boldsymbol{y}_k$ the output vector, $\boldsymbol{v}_k$ measurement noise, $\boldsymbol{w}_k$ process noise, $\boldsymbol{Q}_d$ and $\boldsymbol{R}_d$ covariance matrices of the process noise and measurement noise respectively.

(2.1) can be recognized as the familiar state-space equations extended by two sources of noise $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$. The process noise $\boldsymbol{w}_k$ represents actors on the system that are not included in the model, such as unpredictable influences or modelling errors. The measurement noise $\boldsymbol{v}_k$ represents the error caused by noise in the sensors, e.g. noise caused by the electronics in the sensor itself. By assuming the measurement and modelling noise are Gaussian distributed noises, both will have some covariance that can be determined through experiments or assumptions. An optimal estimate of the states is then found by combining the probability distributions of the model-based estimate and the sensor measurements.

Based on the state-space model of the system as in (2.1), the discrete time Kalman filter is constructed as

$$\begin{aligned} &\textbf{Update stage:} & \hat{\boldsymbol{x}}_k^+ &= \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\left[\boldsymbol{y}_k - \boldsymbol{C}_d\hat{\boldsymbol{x}}_k^-\right] \\ & & \boldsymbol{P}_k^+ &= \left[\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}_d\right]\boldsymbol{P}_k^- \\ &\textbf{Propagation stage:} & \hat{\boldsymbol{x}}_{k+1}^- &= \boldsymbol{A}_d\hat{\boldsymbol{x}}_k^+ + \boldsymbol{B}_d\boldsymbol{u}_k \\ & & \boldsymbol{P}_{k+1}^- &= \boldsymbol{A}_d\boldsymbol{P}_k^+\boldsymbol{A}_d^T + \boldsymbol{\Gamma}\boldsymbol{Q}\boldsymbol{\Gamma}^T \end{aligned} \tag{2.2}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{C}_d^T\left[\boldsymbol{C}_d\boldsymbol{P}_k^-\boldsymbol{C}_d^T + \boldsymbol{R}\right]^{-1} \tag{2.3}$$

$$\begin{aligned} \hat{\boldsymbol{x}}(t_0) &= \hat{\boldsymbol{x}}_0 \\ \boldsymbol{P}_0 &= E\left(\left[\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\right]\left[\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\right]^T\right) \end{aligned} \tag{2.4}$$

In these equations, $\boldsymbol{y}_k$ represents the measured states, $\hat{\boldsymbol{x}}_k$ represents estimates of the state vector, and $\boldsymbol{P}_k$ represents the covariance matrix of the state estimate. The superscripts '+' and '-' are used to differentiate between the variables derived in the update stage and propagation stage respectively. The variables with the superscript '-', derived in the propagation stage, represent the state estimate/covariance based on the model of the system. The variables with the superscript '+', derived in the update stage, represent the optimal estimate and its covariance matrix.

The functionality of the Kalman filter can be interpreted as a cycle split into two stages: an update stage and a propagation stage. In the propagation stage (the final stage of the cycle), an estimate for the states at the next timestep is found based on the model of the system. This can be recognized by the similarity of the equation for $\hat{\boldsymbol{x}}_{k+1}^-$ with a discrete time state-space model. This is achieved by supplying the known input $\boldsymbol{u}_k$ applied to the system, and the optimal estimate of the states $\hat{\boldsymbol{x}}_k^+$ (found in the update stage) to the model of the system. The covariance of this estimate $\boldsymbol{P}_{k+1}^-$ is calculated as well, based on the covariance of the optimal estimate $\boldsymbol{P}_k^+$ and the covariance of the process noise $\boldsymbol{Q}$.

Then, at the start of the next timestep, the optimal estimate for the states $\hat{\boldsymbol{x}}_k^+$ is found by finding the best compromise between the measured states $\boldsymbol{y}_k$ and the best estimate for the states based on the model $\hat{\boldsymbol{x}}_k^-$ (found in the previous propagation stage). This compromise is found by combining the probability distributions of the measurements $\boldsymbol{y}_k$ and estimated states based on the model $\hat{\boldsymbol{x}}_k^-$.

These two stages are consecutively repeated, resulting in an optimal estimate of the state vector for each iteration. For the first iteration, the estimate for the state and its covariance should be initialized, as indicated in (2.4). As mentioned, the Kalman filter inherently assumes that both types of noise included in the truth model are Gaussian distributed, zero-mean white noise. This is a fundamental assumption in using the Kalman filter, but may not be entirely accurate for the 2DOF manipulator. For example, the measurements will be influenced by quantization noise, which is not white. Another example is the possible positions the system can be in. In reality, the end-effector will never be able to reach positions outside of its limited workspace. In a Gaussian distribution, however, values up to infinity are possible in principle. This would imply that in the truth model, there is a chance that the positions may be outside of the real workspace of the 2DOF manipulator. Nevertheless, the Kalman filter is a powerful and elegant tool to deal with systems where uncertainties are involved.

One thing to note is that the Kalman filter estimates the state x of the system, whereas estimates of the external forces are desired. In general, the state vector of a mechatronic system consists of the system's positions and velocities. Notably, these do not include the external forces acting on the system. So, by using the Kalman filter as is, the external forces would not be estimated. This can be remedied by applying an augmented state-space model in the Kalman filter instead. Herein, the state vector is augmented with the variables which are desired to be estimated. As an example, Roveda [10] uses this method to estimate the external forces and the stiffness of springs using an extended Kalman filter. For external force estimation, the augmented state vector becomes,

$$\boldsymbol{x}_{aug} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{F}_{ext} \end{bmatrix}, \tag{2.5}$$

with x the non-augmented state vector of the system, and $\boldsymbol{F}_{ext}$ the external forces to be estimated.

2.3 Kalman filtering on non-linear systems

While the 2DOF manipulator considered in this report is a non-linear system, the discrete time Kalman filter discussed in section 2.2 can only be applied to linear systems. Variations of the Kalman filter that can be applied to non-linear systems are available. Variations that will be considered are the extended Kalman filter (EKF) and the linearized Kalman filter (LKF) [9]. Although these are capable of operating on non-linear systems, the working principles of these variations are the same as the Kalman filter described in section 2.2.

In both the EKF and LKF variants of the Kalman filter, the (non-linear) model of the system is linearized at some nominal state by applying a first-order Taylor series expansion. For a system of non-linear equations,

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{y} &= \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}), \end{aligned} \tag{2.6}$$

first-order Taylor series expansion can be applied as,

$$\begin{aligned} \dot{\widetilde{\boldsymbol{x}}} &= \boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{u}_0) + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0 \\ \boldsymbol{u}=\boldsymbol{u}_0}} \widetilde{\boldsymbol{x}} + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0 \\ \boldsymbol{u}=\boldsymbol{u}_0}} \widetilde{\boldsymbol{u}} \\ \widetilde{\boldsymbol{y}} &= \boldsymbol{g}(\boldsymbol{x}_0, \boldsymbol{u}_0) + \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0 \\ \boldsymbol{u}=\boldsymbol{u}_0}} \widetilde{\boldsymbol{x}} + \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0 \\ \boldsymbol{u}=\boldsymbol{u}_0}} \widetilde{\boldsymbol{u}}, \end{aligned} \tag{2.7}$$

with $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$ the nominal state and input vector, and with $\widetilde{\boldsymbol{x}}$ and $\widetilde{\boldsymbol{u}}$ defined as,

$$\widetilde{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{u}_0 \ , \quad \widetilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_0. \tag{2.8}$$

Each of the partial derivatives in (2.7) can be found as,

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix}, \tag{2.9}$$

where f_i represents the i'th row of the system of non-linear equations $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$, and $\boldsymbol{x}_i$ represents the i'th element of the vector $\boldsymbol{x}$.

The resulting system of linear equations can be applied to the Kalman filter described in section 2.2. For the EKF, the linear model is updated each iteration, where the nominal state and input vectors $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$ are chosen as the current optimal state estimate and current input. This is based on the fundamental assumption that the true state of the system is sufficiently close to the optimal state estimate. If that is the case, the linearized model approximates the behavior of the non-linear system at its current state well. However, since the model needs to be updated constantly, the computational burden is greater compared to the discrete time Kalman filter.

The LKF works similarly to the EKF, the only difference being that the nominal state and input vectors $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$ at which the model is linearized are chosen once beforehand. As opposed to with the EKF, the nominal state and input are not updated for the LKF. In general, the LKF is less accurate than the EKF since the nominal state is usually not as close to the true state of the system. However, the computational burden is far less since the model does not need to be updated at each iteration. Furthermore, the Kalman gain $\boldsymbol{K}_k$ can even be predetermined in experiments and stored, further reducing computational burden.

The LKF will be utilized to estimate the external forces acting on the 2DOF manipulator in this report, because it is expected that the range by which the states will vary is small. The angular deflection of the hinges of the system will not exceed 20°, and the working space of the end-effector is only $100 \times 100mm$. With a smart choice for the nominal state and input vector, these will not differ much from the true state of the system. Thus, it is expected that the LKF will sufficiently approximate the dynamic behavior of the system. Then, the LKF is preferred since it has a smaller computational burden and is easier to implement.

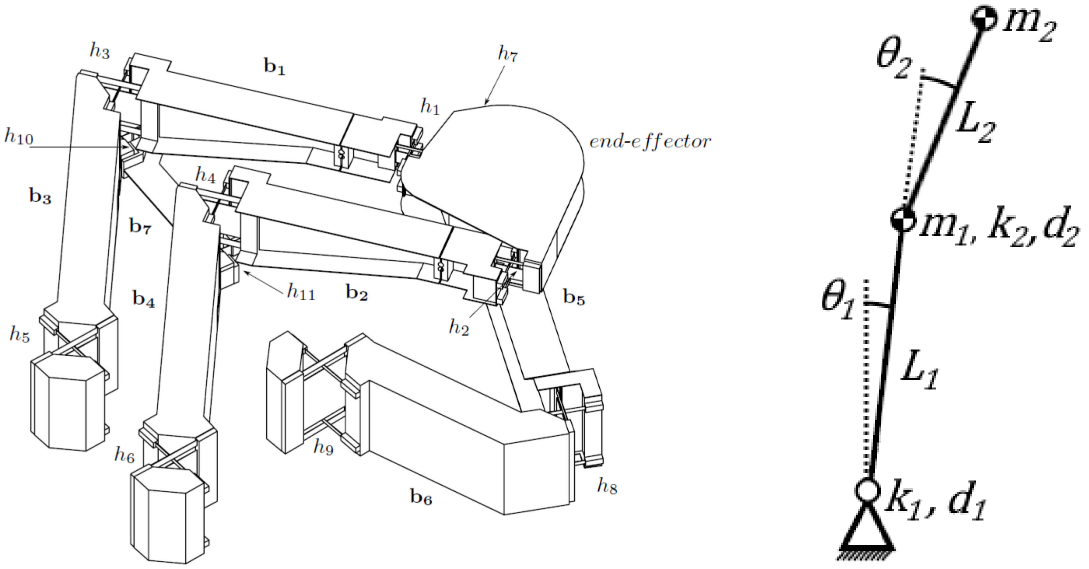
3 DOUBLE PENDULUM

In this chapter, the performance of the LKF will be tested in simulations by applying it to a double pendulum model. This will allow the LKF to be tested in a completely controlled environment. A model of a double pendulum will be significantly less complex than the model of the real system, but will still have some key similarities. Thereby the results of the performance tests on the double pendulum will give an impression of the performance to be expected from the LKF applied to the 2DOF manipulator. At the same time, these procedures will not become too complicated.

This chapter will start by giving an overview of the similarities and differences between a double pendulum and the 2DOF manipulator. Then, a model of the double pendulum will be derived. This model will be linearized and discretized such that it can be applied in the LKF. After finding a model, methods of setting up the LKF are suggested and applied to the double pendulum. Finally, the performance of the LKF set up using these methods is tested in simulations. By varying each of the LKF's parameters, the validity of the methods by which the LKF parameters were set is tested in simulations as well.

3.1 Double pendulum compared to the 2DOF manipulator

The goal in this chapter is to establish (and test the LKF on) a model of a system which approximates the dynamic behavior of the 2DOF manipulator, while being significantly less complex. To this end, the model of an ideal double pendulum will be used. The ideal double pendulum consists of two weightless rotating rigid beams with point masses connected at their ends. The beams are connected to each other and the fixed world by ideal springs and dampers, representing the flexure hinges of the 2DOF manipulator. Similar to the 2DOF manipulator, the double pendulum will operate in the horizontal plane. As such, the effects of gravity can be neglected. The double pendulum and 2DOF manipulator can be seen side-by-side in figure 3.1. In figure 3.1b, L_i represent the length of the arms, m_i the mass connected at the end of each arm, θ_i the (relative) angular displacement of each arm, and k_i and d_i respectively the stiffness and damping linked to the degrees of freedom θ_i.



(a) 2DOF manipulator [1]

(b) Double pendulum

Figure 3.1: Sketch of a double pendulum and 2DOF manipulator

If the single arm or parallel arms of the 2DOF manipulator would be inspected independent of the rest of the setup, their dynamic behavior would be almost identical to the double pendulum. The arms have two rotational degrees of freedom, each effectuated by flexure hinges connecting the beams. Similarly, the double pendulum has two rotational degrees of freedom, with each of its hinges affected by a stiffness and damping representing the flexure hinges. Despite that the parallel arms consist of two sets of arms, their combined dynamic behavior is still similar to a double pendulum. The parallel arms are connected such that their rotational deflections are always the same. As a result, their dynamic parameters (masses, stiffness of hinges, etc.) can effectively be combined. Furthermore, since the end-effector and beam b_7 do not rotate, these do not add complex dynamic behavior to the setup. Instead, these simply add an additional mass term to the rotations of the lower and upper arms.

The greatest difference between the dynamic behavior of the double pendulum and the 2DOF manipulator is caused by the kinematic relation between the parallel arms and single arm of the 2DOF manipulator. Contrary to the kinematic relation between each of the parallel arms, these are not constructed such that their rotational displacements are identical at all times. As such, their (non-linear) dynamic behavior is not 'in-sync', and their dynamic parameters cannot simply be combined. Although the parallel and single arms individually already behave non-linearly, this adds even more non-linear behavior to the 2DOF manipulator as a whole. This behavior is not included in the double pendulum model.

Furthermore, the double pendulum model assumes that the beams are rigid and the flexure hinges can be represented by ideal springs and dampers. Although the beams of the 2DOF manipulators are designed to be stiff, they are not perfectly rigid. This behavior, although small, is neglected in the double pendulum model. Additionally, the flexure hinges do not behave identical to ideal springs and dampers [2]. When the flexure hinges are deflected, they are subject to parasitic motion as well as a variable stiffness. This behavior is also not included in the double pendulum model. The effects of hysteresis, which may be caused by the cables of the actuators and encoders, is also not included in the double pendulum model.

Despite these differences in the dynamic behavior, the double pendulum is a good approximate of the 2DOF manipulator. Some of the most important aspects of the non-linear behavior of the 2DOF manipulator are included in the double pendulum model. Some aspects of its dynamic behavior are neglected, but most of these have a relatively small contribution. The biggest difference is that the 2DOF manipulator has more complex behavior included by the relative movement of the parallel and single arms. To recapitulate, the assumptions made for modelling the double pendulum are listed below.

- Each link consists of a weightless, rigid rod with a point mass connected at the end.
- The links are connected by ideal springs and dampers.
- The links are in nominal position when fully vertical.
- The double pendulum is not influenced by gravity.

3.2 Modelling

3.2.1 Equations of motion

The equations of motion of the double pendulum in figure 3.1b will be derived using the Euler-Lagrange equation,

$$\tau_i = \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}_i}\right] - \frac{\partial L}{\partial \theta_i} \tag{3.1a}$$

$$\mathcal{L} = E_k - E_p \tag{3.1b}$$

with $\mathcal{L}$ the Lagrangian, E_k the total kinetic energy of the system, E_p the total potential energy of the system, θ_i the degrees of freedom of the double pendulum, and τ_i the torque applied to each degree of freedom.

Then, an expression for the Lagrangian $\mathcal{L}$ needs to be determined. To start, an expression for the total kinetic energy of the system E_k will be found. In general, the total kinetic energy of a system includes rotational and translational kinetic energies. The double pendulum consists of two point masses, however, which inherently have zero inertia. Thus, only the translational kinetic energy needs to be considered. To this end, the kinematics relating the translational displacements and velocities of the masses to the degrees of freedom will be determined. These can be found as,

$$\begin{aligned}
x_1 &= \sin(\theta_1) L_1 & x_2 &= \sin(\theta_1) L_1 + \sin(\theta_1+\theta_2) L_2 \\
\dot{x}_1 &= \cos(\theta_1) L_1 \dot{\theta}_1 & \dot{x}_2 &= \cos(\theta_1) L_1\dot{\theta}_1 + \cos(\theta_1+\theta_2) L_2 \left[\dot{\theta}_1 + \dot{\theta}_2\right] \\
y_1 &= \cos(\theta_1) L_1 & y_2 &= \cos(\theta_1) L_1 + \cos(\theta_1+\theta_2) L_2 \\
\dot{y}_1 &= -\sin(\theta_1) L_1 \dot{\theta}_1 & \dot{y}_2 &= -\sin(\theta_1) L_1\dot{\theta}_1 - \sin(\theta_1+\theta_2) L_2 \left[\dot{\theta}_1 + \dot{\theta}_2\right].
\end{aligned} \tag{3.2}$$

The total kinetic energy of the system consists of the kinetic energies of both masses. An expression can be found by substituting the kinematic relations (3.2) into the classical expression for the kinetic energy as,

$$\begin{aligned}
E_k &= E_{k1} + E_{k2} \\
&= \frac{1}{2}m_1\dot{x}_1^2 + \frac{1}{2}m_1\dot{y}_1^2 + \frac{1}{2}m_2\dot{x}_2^2 + \frac{1}{2}m_2\dot{y}_2^2 \\
&= \left[\frac{1}{2}\left[m_1+m_2\right]L_1^2 + \frac{1}{2}m_2L_2^2 + m_2L_1L_2\cos(\theta_2)\right]\dot{\theta}_1^2 + \frac{1}{2}m_2L_2^2\dot{\theta}_2^2 \\
&\quad + \left[m_2L_2^2 + m_2L_1L_2\cos(\theta_2)\right]\dot{\theta}_1\dot{\theta}_2.
\end{aligned} \tag{3.3}$$

Then, an expression for the total potential energy of the system should be found. Since the assumption was made that the double pendulum is not influenced by gravity, the only factors contributing to the total potential energy of the system are the springs. The total potential energy of the system can be found from the classical expression for the energy stored in springs as,

$$\begin{aligned} E_p &= E_{p1} + E_{p2} \\ &= \frac{1}{2} k_1 \theta_1^2 + \frac{1}{2} k_2 \theta_2^2. \end{aligned} \tag{3.4}$$

Substituting (3.3) and (3.4) into (3.1b) gives an expression for the Lagrangian,

$$\begin{aligned} \mathcal{L} = &\left[\frac{1}{2}\left[m_1 + m_2\right] L_1^2 + \frac{1}{2} m_2 L_2^2 + m_2 L_1 L_2 \cos\left(\theta_2\right)\right] \dot{\theta}_1^2 + \frac{1}{2} m_2 L_2^2 \dot{\theta}_2^2 \\ &+ \left[m_2 L_2^2 + m_2 L_1 L_2 \cos\left(\theta_2\right)\right] \dot{\theta}_1 \dot{\theta}_2 - \frac{1}{2} k_1 \theta_1^2 - \frac{1}{2} k_2 \theta_2^2. \end{aligned} \tag{3.5}$$

The equations of motion can be found by substituting (3.5) into (3.1a) and solving. The resulting expression for each τ_i becomes,

$$\begin{aligned} \tau_1 &= \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}_1}\right] - \frac{\partial L}{\partial \theta_1} \\ &= \left[\left[m_1 + m_2\right] L_1^2 + m_2 L_2^2 + 2 m_2 L_1 L_2 \cos\left(\theta_2\right)\right] \ddot{\theta}_1 + \left[m_2 L_2^2 + m_2 L_1 L_2 \cos\left(\theta_2\right)\right] \ddot{\theta}_2 \\ &\quad - m_2 L_1 L_2 \sin\left(\theta_2\right) \left[2 \dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2\right] + k_1 \theta_1 \end{aligned} \tag{3.6a}$$

$$\begin{aligned} \tau_2 &= \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}_2}\right] - \frac{\partial L}{\partial \theta_2} \\ &= \left[m_2 L_2^2 + m_2 L_1 L_2 \cos\left(\theta_2\right)\right] \ddot{\theta}_1 + m_2 L_2^2 \ddot{\theta}_2 + m_2 L_1 L_2 \sin\left(\theta_2\right) \dot{\theta}_1^2 + k_2 \theta_2. \end{aligned} \tag{3.6b}$$

So far, the forces caused by the dampers d_1 and d_2 have not been included in the equations of motion. Unfortunately, including forces caused by damping is not a straightforward task using Lagrangian mechanics. It is noted that these damping terms are included such that the resulting model is stable, not to accurately represent some real system. As a pragmatic solution, the damping forces are added to the equations of motion (3.6) in their conventional form: $d_i \dot{\theta}_i$. This approach is much more convenient, and should be sufficiently accurate for the purposes of testing.

Lastly, the torques τ_i in the equations of motion are split into input torques $\tau_{in,i}$ and external torques $\tau_{ext,i}$ such that these can be distinguished. The resulting equations of motion in matrix form become,

$$\boldsymbol{\tau}_{in} + \boldsymbol{\tau}_{ext} = \boldsymbol{M}(\boldsymbol{\theta})\,\ddot{\boldsymbol{\theta}} + \boldsymbol{C}\left(\dot{\boldsymbol{\theta}},\boldsymbol{\theta}\right)\dot{\boldsymbol{\theta}} + \boldsymbol{K}\boldsymbol{\theta} \tag{3.7a}$$

$$\boldsymbol{M}(\boldsymbol{\theta}) = \begin{bmatrix} [m_1+m_2]\,L_1^2 + m_2L_2^2 + 2m_2L_1L_2\cos(\theta_2) & m_2L_2^2 + m_2L_1L_2\cos(\theta_2) \\ m_2L_2^2 + m_2L_1L_2\cos(\theta_2) & m_2L_2^2 \end{bmatrix} \tag{3.7b}$$

$$\boldsymbol{C}\left(\dot{\boldsymbol{\theta}},\boldsymbol{\theta}\right) = \begin{bmatrix} -2m_2L_1L_2\sin(\theta_2)\,\dot{\theta}_2 + d_1 & -m_2L_1L_2\sin(\theta_2)\,\dot{\theta}_2 \\ m_2L_1L_2\sin(\theta_2)\,\dot{\theta}_1 & d_2 \end{bmatrix} \tag{3.7c}$$

$$\boldsymbol{K} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \tag{3.7d}$$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \quad , \quad \dot{\boldsymbol{\theta}} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad , \quad \ddot{\boldsymbol{\theta}} = \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} \quad , \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{3.7e}$$

3.2.2 Augmented state-space model

The LKF works based on a state-space model, and thus the equations of motion (3.7) need to be converted to state-space form,

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x}, \end{aligned} \tag{3.8}$$

with $\boldsymbol{x}$ the state vector, $\dot{\boldsymbol{x}}$ its derivative with respect to time, $\boldsymbol{y}$ the output vector, $\boldsymbol{u}$ the input vector, and $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ state-space matrices.

In accordance with the 2DOF manipulator, which only has position sensors, the output vector consists of the displacements $\boldsymbol{\theta}$. Furthermore, since the LKF is used to find an estimate the state $\boldsymbol{x}$ of the system, the state vector should be augmented to include the external torques on the double pendulum. The resulting state-, output-, and input-vector of the augmented state-space model become,

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\theta} \\ \dot{\boldsymbol{\theta}} \\ \boldsymbol{\tau}_{ext} \end{bmatrix} \quad , \quad \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \ddot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{\tau}}_{ext} \end{bmatrix} \quad , \quad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{\theta} \end{bmatrix} \quad , \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{\tau}_{in} \end{bmatrix} \tag{3.9}$$

The state-space matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ can be found by writing expressions for each of the elements in the $\dot{\boldsymbol{x}}$ and $\boldsymbol{y}$ vectors as a function of $\boldsymbol{x}$ and $\boldsymbol{u}$. Most of these are trivial to find, except for $\dot{\boldsymbol{\tau}}_{ext}$ and $\ddot{\boldsymbol{\theta}}$. Unfortunately, it is not possible to find an expression for $\dot{\boldsymbol{\tau}}_{ext}$, since the external torques are unknown. It can be assumed that these are not a function of the state- and input-vector, however. As such, the corresponding rows of the $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices will be set to zero.

An expression for the accelerations $\ddot{\boldsymbol{\theta}}$ can be found by rewriting the equations of motion (3.7) as,

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{M}^{-1}(\boldsymbol{\theta})\,\boldsymbol{\tau}_{in} + \boldsymbol{M}^{-1}(\boldsymbol{\theta})\,\boldsymbol{\tau}_{ext} - \boldsymbol{M}^{-1}(\boldsymbol{\theta})\,\boldsymbol{C}\left(\dot{\boldsymbol{\theta}},\boldsymbol{\theta}\right)\dot{\boldsymbol{\theta}} - \boldsymbol{M}^{-1}(\boldsymbol{\theta})\,\boldsymbol{K}\boldsymbol{\theta}. \tag{3.10}$$

All matrices in (3.10) are known except the inverse of the inertia matrix $\boldsymbol{M}^{-1}$. The inertia matrix is a 2×2 matrix, which means its inverse can be found analytically as,

$$\begin{aligned}\boldsymbol{M}^{-1}(\boldsymbol{\theta}) &= \frac{1}{M_{11}M_{22} - M_{12}M_{21}}\begin{bmatrix} M_{22} & -M_{12} \\ -M_{21} & M_{11}\end{bmatrix} \\ &= \begin{bmatrix} \frac{m_2L_2^2}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\,\text{cos}(\theta_2)} & -\frac{m_2L_2^2+m_2L_1L_2\,\text{cos}(\theta_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\,\text{cos}(\theta_2)} \\ -\frac{m_2L_2^2+m_2L_1L_2\,\text{cos}(\theta_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\,\text{cos}(\theta_2)} & \frac{[m_1+m_2]L_1^2+m_2L_2^2+2m_2L_1L_2\,\text{cos}(\theta_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\,\text{cos}(\theta_2)}\end{bmatrix},\end{aligned} \tag{3.11}$$

where M_{ij} represents the element (i, j) of the inertia matrix $\boldsymbol{M}$.

Then, the augmented state-space model becomes,

$$\begin{aligned}\dot{\boldsymbol{x}} &= \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{M}^{-1}(\boldsymbol{\theta})\,\boldsymbol{K} & -\boldsymbol{M}^{-1}(\boldsymbol{\theta})\,\boldsymbol{C}\left(\dot{\boldsymbol{\theta}}, \boldsymbol{\theta}\right) & \boldsymbol{M}^{-1}(\boldsymbol{\theta}) \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{x} + \begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{M}^{-1}(\boldsymbol{\theta}) \\ \boldsymbol{0}\end{bmatrix}\boldsymbol{u} \\ \boldsymbol{y} &= \begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{x}.\end{aligned} \tag{3.12}$$

3.2.3 Linearization

To be able to apply the augmented state-space model (3.12) to the LKF, it needs to be linearized. This will be achieved by applying first-order Taylor series expansion as shown in section 2.3. The augmented state-space model can be written according to (2.6) as,

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{y} &= \boldsymbol{g}(\boldsymbol{x})\,.\end{aligned} \tag{3.13}$$

The nominal state- and input-vector $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$ should be chosen such that they are as close as possible to the true state- and input-vector at any time. These will be set as,

$$\boldsymbol{x}_0 = \boldsymbol{0}\,, \quad \boldsymbol{u}_0 = \boldsymbol{0}, \tag{3.14}$$

which represents the double pendulum when it is in its nominal position, with zero velocity, external torque, and input force. This is the optimal choice, since the expected range of each of these states and inputs is centered around zero. Thus, the expected value of each state and input is 0.

With this choice for the nominal state and input vector, some of the terms in (2.7) become much simpler. Applying these values for $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$ to (3.12) and (2.8) gives,

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{u}_0) &= \boldsymbol{0} \\ \boldsymbol{g}(\boldsymbol{x}_0) &= \boldsymbol{0},\end{aligned} \tag{3.15}$$

$$\begin{aligned}\widetilde{\boldsymbol{x}} &= \boldsymbol{x} \\ \widetilde{\boldsymbol{u}} &= \boldsymbol{u}.\end{aligned} \tag{3.16}$$

Then, (2.7) applied to the augmented state-space model of the double pendulum becomes,

$$\begin{aligned}\dot{\tilde{\boldsymbol{x}}} &= \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} \boldsymbol{x} + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} \boldsymbol{u} \\ \tilde{\boldsymbol{y}} &= \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_0} \boldsymbol{x},\end{aligned} \tag{3.17}$$

with the partial derivatives,

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_6} \\ \vdots & & \vdots \\ \frac{\partial f_6}{\partial x_1} & \cdots & \frac{\partial f_6}{\partial x_6} \end{bmatrix}, \quad \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \vdots & \\ \frac{\partial f_6}{\partial u_1} & \frac{\partial f_6}{\partial u_2} \end{bmatrix}, \quad \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_6} \\ \frac{\partial g_2}{\partial x_1} & & \frac{\partial g_2}{\partial x_6} \end{bmatrix}. \tag{3.18}$$

What remains is to find expressions for the partial derivatives. Fortunately, a significant number of the equations f_i and g_i is already linear. This means the partial derivatives of these equations can simply be read from the matrices in (3.12). The partial derivatives to which this applies are,

$$\begin{aligned} \frac{\partial f_1}{\partial \boldsymbol{x}} &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} & \frac{\partial f_1}{\partial \boldsymbol{u}} &= \begin{bmatrix} 0 & 0 \end{bmatrix} \\ \frac{\partial f_2}{\partial \boldsymbol{x}} &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} & \frac{\partial f_2}{\partial \boldsymbol{u}} &= \begin{bmatrix} 0 & 0 \end{bmatrix} \\ \frac{\partial f_5}{\partial \boldsymbol{x}} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \frac{\partial f_5}{\partial \boldsymbol{u}} &= \begin{bmatrix} 0 & 0 \end{bmatrix} \\ \frac{\partial f_6}{\partial \boldsymbol{x}} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \frac{\partial f_6}{\partial \boldsymbol{u}} &= \begin{bmatrix} 0 & 0 \end{bmatrix}, \end{aligned} \tag{3.19}$$

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}. \tag{3.20}$$

The only non-linear functions of which the partial derivatives need to be found are f_3 and f_4, which correspond to the rewritten equations of motion (3.10). These functions can be written as,

$$\begin{aligned} f_3 = &-\left[\boldsymbol{M}^{-1}\boldsymbol{K}\right]_{11} x_1 - \left[\boldsymbol{M}^{-1}\boldsymbol{K}\right]_{12} x_2 - \left[\boldsymbol{M}^{-1}\boldsymbol{C}\right]_{11} x_3 - \left[\boldsymbol{M}^{-1}\boldsymbol{C}\right]_{12} x_4 \\ &+ \left[\boldsymbol{M}^{-1}\right]_{11} x_5 + \left[\boldsymbol{M}^{-1}\right]_{12} x_6 + \left[\boldsymbol{M}^{-1}\right]_{11} u_1 + \left[\boldsymbol{M}^{-1}\right]_{12} u_2 \\ = &-\frac{m_2 L_2^2}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} k_1 x_1 + \frac{m_2 L_2^2 + m_2 L_1 L_2 \cos(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} k_2 x_2 \\ &- \frac{m_2 L_2^2 d_1}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} x_3 + \frac{m_2^2 L_1 L_2^3 \sin(x_2) + \frac{1}{2} m_2^2 L_1^2 L_2^2 \sin(2x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} x_3^2 \\ &+ \frac{2 m_2^2 L_1 L_2^3 \sin(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} x_3 x_4 + \frac{m_2 L_2^2 d_2 + m_2 L_1 L_2 d_2 \cos(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} x_4 \\ &+ \frac{m_2^2 L_1 L_2^3 \sin(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} x_4^2 + \frac{m_2 L_2^2}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} x_5 \\ &- \frac{m_2 L_2^2 + m_2 L_1 L_2 \cos(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} x_6 + \frac{m_2 L_2^2}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} u_1 \\ &- \frac{m_2 L_2^2 + m_2 L_1 L_2 \cos(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} u_2, \end{aligned} \tag{3.21a}$$

$$
\begin{aligned}
f_4 = &- \left[\boldsymbol{M}^{-1}\boldsymbol{K}\right]_{21} x_1 - \left[\boldsymbol{M}^{-1}\boldsymbol{K}\right]_{22} x_2 - \left[\boldsymbol{M}^{-1}\boldsymbol{C}\right]_{21} x_3 - \left[\boldsymbol{M}^{-1}\boldsymbol{C}\right]_{22} x_4 \\
&+ \left[\boldsymbol{M}^{-1}\right]_{21} x_5 + \left[\boldsymbol{M}^{-1}\right]_{22} x_6 + \left[\boldsymbol{M}^{-1}\right]_{21} u_1 + \left[\boldsymbol{M}^{-1}\right]_{22} u_2 \\
= &\frac{m_2L_2^2 + m_2L_1L_2\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}k_1x_1 - \frac{\left[m_1+m_2\right]L_1^2 + m_2L_2^2 + 2m_2L_1L_2\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}k_2x_2 \\
&- \frac{2m_2^2L_1L_2^3\sin(x_2) + m_2^2L_1^2L_2^2\sin(2x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}x_3x_4 \\
&+ \frac{m_2L_2^2d_1}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}x_3 + \frac{m_2L_1L_2d_1\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}x_3 \\
&- \frac{\left[m_1m_2+m_2^2\right]L_1^3L_2\sin(x_2) + m_2^2L_1L_2^3\sin(x_2) + m_2^2L_1^2L_2^2\sin(2x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}x_3^2 \\
&- \frac{\left[m_1+m_2\right]L_1^2d_2 + m_2L_2^2d_2}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}x_4 - \frac{2m_2L_1L_2d_2\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}x_4 \\
&- \frac{m_2^2L_1L_2^3\sin(x_2) + \frac{1}{2}m_2^2L_1^2L_2^2\sin(2x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}x_4^2 - \frac{m_2L_2^2 + m_2L_1L_2\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}x_5 \\
&+ \frac{\left[m_1+m_2\right]L_1^2 + m_2L_2^2 + 2m_2L_1L_2\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}x_6 - \frac{m_2L_2^2 + m_2L_1L_2\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}u_1 \\
&+ \frac{\left[m_1+m_2\right]L_1^2 + m_2L_2^2 + 2m_2L_1L_2\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)}u_2,
\end{aligned}
\tag{3.21b}
$$

where the notation $[\boldsymbol{X}]_{ij}$ represents the (i,j)'th element of a matrix $\boldsymbol{X}$, and the matrix products $\boldsymbol{M}^{-1}\boldsymbol{K}$ and $\boldsymbol{M}^{-1}\boldsymbol{C}$ can be expressed as,

$$
\boldsymbol{M}^{-1}\boldsymbol{K} = \begin{bmatrix} \frac{m_2L_2^2}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}k_1 & -\frac{m_2L_2^2+m_2L_1L_2\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}k_2 \\ -\frac{m_2L_2^2+m_2L_1L_2\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}k_1 & \frac{\left[m_1+m_2\right]L_1^2+m_2L_2^2+2m_2L_1L_2\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}k_2 \end{bmatrix} \tag{3.22a}
$$

$$
\left[\boldsymbol{M}^{-1}\boldsymbol{C}\right]_{11} = \frac{m_2L_2^2d_1 - m_2^2L_1L_2^3\sin(x_2)x_3 - \frac{1}{2}m_2^2L_1^2L_2^2\sin(2x_2)x_3 - 2m_2^2L_1L_2^3\sin(x_2)x_4}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)} \tag{3.22b}
$$

$$
\left[\boldsymbol{M}^{-1}\boldsymbol{C}\right]_{12} = -\frac{m_2L_2^2d_2 + m_2L_1L_2d_2\cos(x_2) + m_2^2L_1L_2^3\sin(x_2)x_4}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)} \tag{3.22c}
$$

$$
\begin{aligned}
\left[\boldsymbol{M}^{-1}\boldsymbol{C}\right]_{21} = &\frac{1}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)} \cdot \Bigg[- m_2L_2^2d_1 - m_2L_1L_2d_1\cos(x_2) \\
&+ \left[m_1m_2+m_2^2\right]L_1^3L_2\sin(x_2)x_3 + m_2^2L_1L_2^3\sin(x_2)x_3 + m_2^2L_1^2L_2^2\sin(2x_2)x_3 \\
&+ 2m_2^2L_1L_2^3\sin(x_2)x_4 + m_2^2L_1^2L_2^2\sin(2x_2)x_4\Bigg]
\end{aligned}
\tag{3.22d}
$$

$$
\begin{aligned}
\left[\boldsymbol{M}^{-1}\boldsymbol{C}\right]_{22} = &\frac{1}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2 - m_2^2L_1^2L_2^2\cos(x_2)} \cdot \Bigg[\left[m_1+m_2\right]L_1^2d_2 + m_2L_2^2d_2 \\
&+ 2m_2L_1L_2d_2\cos(x_2) + m_2^2L_1L_2^3\sin(x_2)x_4 + \frac{1}{2}m_2^2L_1^2L_2^2\sin(2x_2)x_4\Bigg].
\end{aligned}
\tag{3.22e}
$$

Then, the partial derivatives of f_3 and f_4 can be found with respect to each state x_i and input u_i. Due to the size of the resulting expressions, these have been listed in appendix A. Substituting $\boldsymbol{x}_0 = \mathbf{0}$ and $\boldsymbol{u}_0 = \mathbf{0}$ into these expressions gives,

$$
\begin{aligned}
\left.\frac{\partial f_3}{\partial x_1}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= -\frac{k_1}{m_1 L_1^2} & \left.\frac{\partial f_4}{\partial x_1}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= \frac{[L_1+L_2]\,k_1}{m_1 L_1^2 L_2} \\
\left.\frac{\partial f_3}{\partial x_2}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= \frac{[L_1+L_2]\,k_2}{m_1 L_1^2 L_2} & \left.\frac{\partial f_4}{\partial x_2}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= -\frac{k_2}{m_2 L_2^2} - \frac{[L_1+L_2]^2\,k_2}{m_1 L_1^2 L_2^2} \\
\left.\frac{\partial f_3}{\partial x_3}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= -\frac{d_1}{m_1 L_1^2} & \left.\frac{\partial f_4}{\partial x_3}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= \frac{[L_1+L_2]\,d_1}{m_1 L_1^2 L_2} \\
\left.\frac{\partial f_3}{\partial x_4}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= \frac{[L_1+L_2]\,d_2}{m_1 L_1^2 L_2} & \left.\frac{\partial f_4}{\partial x_4}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= -\frac{d_2}{m_2 L_2^2} - \frac{[L_1+L_2]^2\,d_2}{m_1 L_1^2 L_2^2} \\
\left.\frac{\partial f_3}{\partial x_5}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= \frac{1}{m_1 L_1^2} & \left.\frac{\partial f_4}{\partial x_5}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= -\frac{L_1+L_2}{m_1 L_1^2 L_2} \\
\left.\frac{\partial f_3}{\partial x_6}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= -\frac{L_1+L_2}{m_1 L_1^2 L_2} & \left.\frac{\partial f_4}{\partial x_6}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= \frac{1}{m_2 L_2^2} + \frac{[L_1+L_2]^2}{m_1 L_1^2 L_2^2} \\
\left.\frac{\partial f_3}{\partial u_1}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= \frac{1}{m_1 L_1^2} & \left.\frac{\partial f_4}{\partial u_1}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= -\frac{L_1+L_2}{m_1 L_1^2 L_2} \\
\left.\frac{\partial f_3}{\partial u_2}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= -\frac{L_1+L_2}{m_1 L_1^2 L_2} & \left.\frac{\partial f_4}{\partial u_2}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}_0\\ \boldsymbol{u}=\boldsymbol{u}_0}} &= \frac{1}{m_2 L_2^2} + \frac{[L_1+L_2]^2}{m_1 L_1^2 L_2^2}.
\end{aligned}
\tag{3.23}
$$

Finally, the linearized system can be found by substituting (3.19), (3.20), and (3.23) into (3.17). This linearized system can easily be written as a state-space system such as (3.8). The state-space matrices of this linearized augmented-state model of the double pendulum then become,

$$
\begin{aligned}
\boldsymbol{A}_{lin} &= \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
-\frac{k_1}{m_1 L_1^2} & \frac{[L_1+L_2]k_2}{m_1 L_1^2 L_2} & -\frac{d_1}{m_1 L_1^2} & \frac{[L_1+L_2]d_2}{m_1 L_1^2 L_2} & \frac{1}{m_1 L_1^2} & -\frac{L_1+L_2}{m_1 L_1^2 L_2} \\
\frac{[L_1+L_2]k_1}{m_1 L_1^2 L_2} & -\frac{k_2}{m_2 L_2^2} - \frac{[L_1+L_2]^2 k_2}{m_1 L_1^2 L_2^2} & \frac{[L_1+L_2]d_1}{m_1 L_1^2 L_2} & -\frac{d_2}{m_2 L_2^2} - \frac{[L_1+L_2]^2 d_2}{m_1 L_1^2 L_2^2} & -\frac{L_1+L_2}{m_1 L_1^2 L_2} & \frac{1}{m_2 L_2^2} + \frac{[L_1+L_2]^2}{m_1 L_1^2 L_2^2} \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}, \\
\boldsymbol{B}_{lin} &= \begin{bmatrix}
0 & 0 \\
0 & 0 \\
\frac{1}{m_1 L_1^2} & -\frac{L_1+L_2}{m_1 L_1^2 L_2} \\
-\frac{L_1+L_2}{m_1 L_1^2 L_2} & \frac{1}{m_2 L_2^2} + \frac{[L_1+L_2]^2}{m_1 L_1^2 L_2^2} \\
0 & 0 \\
0 & 0
\end{bmatrix}, \\
\boldsymbol{C}_{lin} &= \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}.
\end{aligned}
\tag{3.24}
$$

3.2.4 Discretization

The LKF as defined in section 2.3 operates on a discrete-time model. Thus, the (continuous-time) model derived previously should be converted to a state-space model in discrete-time as,

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{A}_d \boldsymbol{x}_k + \boldsymbol{B}_d \boldsymbol{u}_k, \\ \boldsymbol{y}_k &= \boldsymbol{C}_d \boldsymbol{x}_k, \end{aligned} \tag{3.25}$$

where the subscript k represents the time at which the signal is evaluated as $k \cdot T_s$.

The state-space model will be discretized using the forward Euler method, where the derivatives of the states are estimated as,

$$\dot{\boldsymbol{x}} \approx \frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{T_s}. \tag{3.26}$$

The discrete-time equivalent of a continuous-time state-space model (3.8) can be found by substituting (3.26), and converting all continuous vectors to discrete vectors. Rewriting the result to the standard discrete state-space form as in (3.25) gives,

$$\begin{aligned} \frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{T_s} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k, \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k, \end{aligned} \qquad \Longrightarrow \qquad \begin{aligned} \boldsymbol{x}_{k+1} &= \left(\boldsymbol{I} + T_s \boldsymbol{A}\right) \boldsymbol{x}_k + T_s \boldsymbol{B}\boldsymbol{u}_k, \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k. \end{aligned} \tag{3.27}$$

This gives expressions by which the discrete-time state-space matrices can be found from the continuous-time state-space matrices:

$$\begin{aligned} \boldsymbol{A}_d &= \left(\boldsymbol{I} + T_s \boldsymbol{A}\right), \\ \boldsymbol{B}_d &= T_s \boldsymbol{B}, \\ \boldsymbol{C}_d &= \boldsymbol{C}. \end{aligned} \tag{3.28}$$

Then, the discrete-time state-space matrices of the (linearized, augmented-state) double pendulum model can be found by substituting the continuous state-space matrices (3.8) into (3.28). The resulting matrices are:

$$\begin{aligned}
\boldsymbol{A}_d &= \begin{bmatrix}
1 & 0 & T_s & 0 & 0 & 0 \\
0 & 1 & 0 & T_s & 0 & 0 \\
-\frac{k_1 T_s}{m_1 L_1^2} & \frac{[L_1+L_2]k_2 T_s}{m_1 L_1^2 L_2} & 1-\frac{d_1 T_s}{m_1 L_1^2} & \frac{[L_1+L_2]d_2 T_s}{m_1 L_1^2 L_2} & \frac{T_s}{m_1 L_1^2} & -\frac{[L_1+L_2]T_s}{m_1 L_1^2 L_2} \\
\frac{[L_1+L_2]k_1 T_s}{m_1 L_1^2 L_2} & -\frac{k_2 T_s}{m_2 L_2^2}-\frac{[L_1+L_2]^2 k_2 T_s}{m_1 L_1^2 L_2^2} & \frac{[L_1+L_2]d_1 T_s}{m_1 L_1^2 L_2} & 1-\frac{d_2 T_s}{m_2 L_2^2}-\frac{[L_1+L_2]^2 d_2 T_s}{m_1 L_1^2 L_2^2} & -\frac{[L_1+L_2]T_s}{m_1 L_1^2 L_2} & \frac{T_s}{m_2 L_2^2}+\frac{[L_1+L_2]^2 T_s}{m_1 L_1^2 L_2^2} \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}, \\
\boldsymbol{B}_d &= \begin{bmatrix}
0 & 0 \\
0 & 0 \\
\frac{T_s}{m_1 L_1^2} & -\frac{[L_1+L_2]T_s}{m_1 L_1^2 L_2} \\
-\frac{[L_1+L_2]T_s}{m_1 L_1^2 L_2} & \frac{T_s}{m_2 L_2^2} + \frac{[L_1+L_2]^2 T_s}{m_1 L_1^2 L_2^2} \\
0 & 0 \\
0 & 0
\end{bmatrix}, \\
\boldsymbol{C}_d &= \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}.
\end{aligned} \tag{3.29}$$

3.3 Implementing the linearized Kalman filter

To implement the LKF its parameters need to be set, namely the process noise covariance matrix $\boldsymbol{Q}_d$, the process noise transfer matrix $\boldsymbol{\Gamma}_d$, the measurement noise covariance matrix $\boldsymbol{R}_d$, the initial state-estimate $\hat{\boldsymbol{x}}_0$, and the initial state-estimate covariance matrix $\boldsymbol{P}_0$. One of the advantages of testing the LKF on a fully controlled environment is that the optimal values for these parameters can be found, such that the LKF can be tested in ideal circumstances. This is especially true for the initial state-estimate $\boldsymbol{x}_0$ and its covariance matrix $\boldsymbol{P}_0$. The initial states of the system are exactly known, so the initial estimate can be set to the true values if desired. Since this initial estimate is guaranteed to be 100% accurate, the initial state-estimate covariance matrix $\boldsymbol{P}_0$ can even be set to zero.

Then, the measurement noise covariance matrix $\boldsymbol{R}_d$ should be set such that the measurement noise $\boldsymbol{v}_k$ is accurately included. It is assumed that there is no cross-correlation between the noise acting on each of the sensors, which gives:

$$\boldsymbol{R}_d = \begin{bmatrix} \sigma_{v1}^2 & 0 \\ 0 & \sigma_{v2}^2 \end{bmatrix} \tag{3.30}$$

with σ_{vi}^2 the variance of the measurement noise of sensor i.

The measurement noise will be split into two parts as,

$$\boldsymbol{v}_k = \boldsymbol{v}_{quant,k} + \boldsymbol{v}_{rest,k}, \tag{3.31}$$

with $\boldsymbol{v}_{quant,k}$ noise caused by quantization, and $\boldsymbol{v}_{rest,k}$ any remaining contributions. It is assumed that there is no correlation between the quantization noise and the remaining contributions. As such, the variance of the total measurement noise σ_{vi}^2 can be found by adding the variance of the quantization noise $\sigma_{quant,vi}^2$ and the variance of the remaining contributions $\sigma_{rest,vi}^2$:

$$\sigma_{vi}^2 = \sigma_{quant,vi}^2 + \sigma_{rest,vi}^2. \tag{3.32}$$

Quantization noise is caused by the (finite) resolution of the encoders. This will cause an error, as the real value will be rounded up or down to the closest incremental step. Suppose the distance between each increment of the encoder is d_{enc}. Given an encoder reading, the real displacement value can be assumed to lie somewhere within a range of d_{enc} of that value. Furthermore, each position within that range for the real value is equally probable. Thus, it is assumed that the error caused by quantization has a uniform distribution with a range d_{enc}. For a uniform distribution between two points a and b, the variance can be calculated as,

$$\sigma^2 = \frac{[b-a]^2}{12}. \tag{3.33}$$

Substituting d_{enc} into (3.33) gives an approximate for the variance of the quantization noise:

$$\sigma_{quant,vi}^2 = \frac{d_{enc}^2}{12}. \tag{3.34}$$

It should be noted that while the quantization noise has a uniform distribution, the measurement noise in the LKF is assumed to be Gaussian. Thus, it can be expected that applying the variance calculated using (3.34) will not give a fully accurate representation of the quantization noise.

The measurement noise caused by the remaining contributions is exactly known in the simulations. These will be included using a 'band-limited white noise' block. This block generates normally distributed random numbers, and its variance $\sigma_{rest,vi}^2$ can be chosen by the user. Thus, the variance $\sigma_{rest,vi}^2$ can be set to the true value if desired.

What remains is to include the process noise $\boldsymbol{w}_k$ accurately. This noise represents any distortion in $\boldsymbol{x}_{k+1}$ caused by errors in the model used by the LKF. The errors that will be included in $\boldsymbol{w}_k$ in this chapter are the parameter identification errors, and the change of the external torques. To this end, the process noise $\boldsymbol{w}_k$ will be split as,

$$\boldsymbol{w}_k = \begin{bmatrix} \boldsymbol{w}_{par,k} \\ \boldsymbol{w}_{F,k} \end{bmatrix}, \tag{3.35}$$

with $\boldsymbol{w}_{par,k}$ the noise caused by errors in the parameters, and $\boldsymbol{w}_{F,k}$ noise caused by changes in the external torques.

It is assumed that there is no correlation between the noise caused by errors in the parameters $\boldsymbol{w}_{par,k}$ and the noise caused by changes in the external torques $\boldsymbol{w}_{F,k}$. Thus, the covariance matrix of the total process noise $\boldsymbol{Q}_d$ can be written as,

$$\boldsymbol{Q}_d = \begin{bmatrix} \boldsymbol{Q}_{d,par} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_{d,F} \end{bmatrix}, \tag{3.36}$$

with $\boldsymbol{Q}_{d,par}$ the covariance matrix of $\boldsymbol{w}_{par,k}$, and $\boldsymbol{Q}_{d,F}$ the covariance matrix of $\boldsymbol{w}_{F,k}$.

Although the external torques are probably not constant, they are represented as constant states in the model of the system. This can be recognized from the state-space equations in section 3.2.4. The relevant state-space equation is repeated in (3.37).

$$\boldsymbol{\tau}_{ext,k+1} = \boldsymbol{\tau}_{ext,k} \tag{3.37}$$

Naturally, this is not accurate. Since the change of the external torques is not included in the model, it will be represented by the noise $\boldsymbol{w}_{F,k}$ instead. The covariance matrix of this noise should then be determined based on the expected change of the external torques for each time step. It is assumed that there is no correlation between the change in external torques on each arm. This gives for the covariance matrix of the noise $\boldsymbol{w}_{F,k}$,

$$\boldsymbol{Q}_{d,F} = \begin{bmatrix} \sigma^2_{F,w1} & 0 \\ 0 & \sigma^2_{F,w2} \end{bmatrix}, \tag{3.38}$$

with $\sigma^2_{F,wi}$ the variance of the noise caused by changes in the external torque acting on beam i.

For the simulations in this chapter, the external torques are known exactly beforehand. Then, the variances $\sigma^2_{F,wi}$ can be calculated at the start of each simulation. This will be achieved by analyzing the change in the external torques for each time step within some range of the external torque signals. The range chosen should be the most 'dynamic' range in the external torque signals. Otherwise, the possible amount by which the external torques can change between each time step may be underestimated. The variances $\sigma^2_{F,wi}$ can then be found as,

$$\sigma^2_{F,wi} = \mathit{var}(\Delta\boldsymbol{\tau}_{ext,i}), \tag{3.39}$$

with $\Delta\boldsymbol{\tau}_{ext,i}$ The vector containing the change in the external torque on beam i between each time step over the given range.

Next, the noise $\boldsymbol{w}_{par,k}$ will be discussed. This noise represents the errors caused by wrongly identified parameters. These are more complicated to include, as these errors depend on the states. For example, if the stiffness of a spring was not identified correctly, the error this causes in a model will be greater the more this spring is deflected. In Kalman filters, the noise is not a function of the states, however. As an approximation, the covariance matrix of the noise $\boldsymbol{w}_{par,k}$ will be found based on data gathered from simulations using the setup shown in figure 3.2.

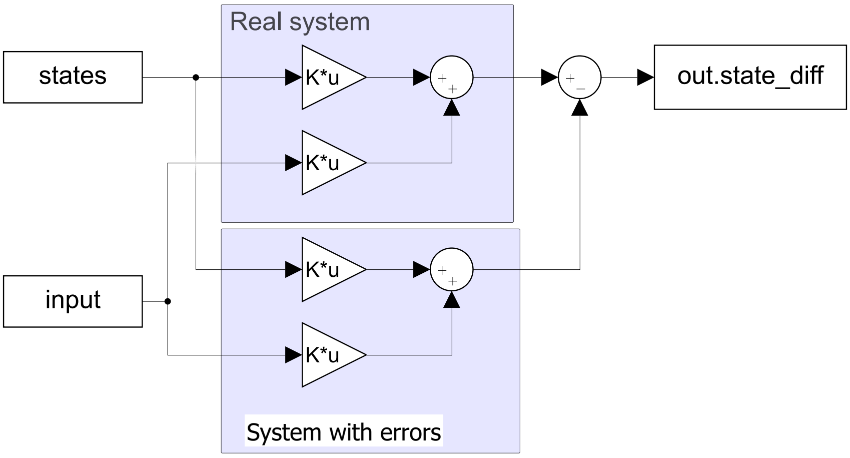


Figure 3.2: Simulink model to derive covariance matrix of $\boldsymbol{w}_{par,k}$

Suppose the range between which the states and inputs vary are known, as well as the error range of the parameters. Then, random state- and input-vectors and a model based on random 'wrong' parameters can be generated within their respective ranges. Using the setup shown in figure 3.2, the errors caused by this 'wrong' model can be found for the randomly generated state- and input-vectors. This process can be repeated a number of times, where for each simulation a new 'wrong' model and a new set of random state- and input-vectors are generated. If this process is repeated sufficient times, an approximate of the variance of $\boldsymbol{w}_{par,k}$ for the given ranges of the states, inputs, and parameter errors can be found based on the generated data. It is noted that the only state-space equations where the model's parameters are present are those for the velocities $\dot{\boldsymbol{\theta}}_i$. Thus, these will be the only non-zero elements of the output signal of the system in figure 3.2. Then, the covariance matrix $\boldsymbol{Q}_{d,par}$ can be found as,

$$\boldsymbol{Q}_{d,par} = \begin{bmatrix} var(\boldsymbol{e}_{x_3}) & 0 \\ 0 & var(\boldsymbol{e}_{x_4}) \end{bmatrix}, \tag{3.40}$$

with $\boldsymbol{e}_{x_i}$ the error data collected in all simulations for state x_i.

The range between which the model's parameters can differ from their real values are known, and the range between which the states and inputs vary can be found from simulations as well. This method does assume that the probability that some state $\boldsymbol{x}_k$ occurs is uniformly distributed, however. In reality this is probably not true. Nevertheless, this method gives an approximate for the covariance matrix of $\boldsymbol{w}_{par,k}$.

It should be noted that not all errors in the model have been accounted for in the process noise $\boldsymbol{w}_k$. For one, the effects of linearizing the model were not taken into account. While the real double pendulum is not linear, the LKF will operate based on a linearized model. The further the states and inputs are from the nominal state- and input-vectors (3.14), the larger the error caused by the linearization will be. As discussed before, the non-linearity of the system is assumed to be very small within the ranges the double pendulum will operate. Another neglected aspect that may cause process errors in the 2DOF manipulator is the noise in the actuators.

Finally, the process noise transfer matrix $\boldsymbol{\Gamma}$ can be found. This matrix relates each element of the noise vector $\boldsymbol{w}_k$ to the states $\boldsymbol{x}_{k+1}$. The noise caused by parameter errors $\boldsymbol{w}_{par,k}$ relates specifically to $x_{3,k+1}$ and $x_{4,k+1}$, which corresponds to the velocities. The noise caused by the change in external torques $\boldsymbol{w}_{F,k}$ relates specifically to $x_{5,k+1}$ and $x_{6,k+1}$, which corresponds to the external torques. Then, the process noise transfer matrix $\boldsymbol{\Gamma}$ becomes,

$$\mathbf{\Gamma} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.41}$$

3.4 Simulation setup

In this section, the simulation setup for testing the LKF on the double pendulum will be discussed. The goal of these simulations is to find the performance that can be expected for the real system in a controlled environment. As such, it will be attempted to mimic the circumstances of the 2DOF manipulator realistically.

The equation of motion (3.7) of the double pendulum will be used as the plant model in the simulations. This choice was made because the double pendulum has key similarities with the 2DOF manipulator, while also being less complex. The input of the plant are the torques applied to the system. These can be split in two parts: the known actuator torques, and the unknown external torques. The output of the plant is the angular displacement of the beams. Based on this output, a sensor measurement signal will be created by adding white noise and applying quantization. This should give a somewhat realistic representation of the real measurement signal. The plant and its input and output signals as implemented in Simulink are illustrated in figure 3.3.

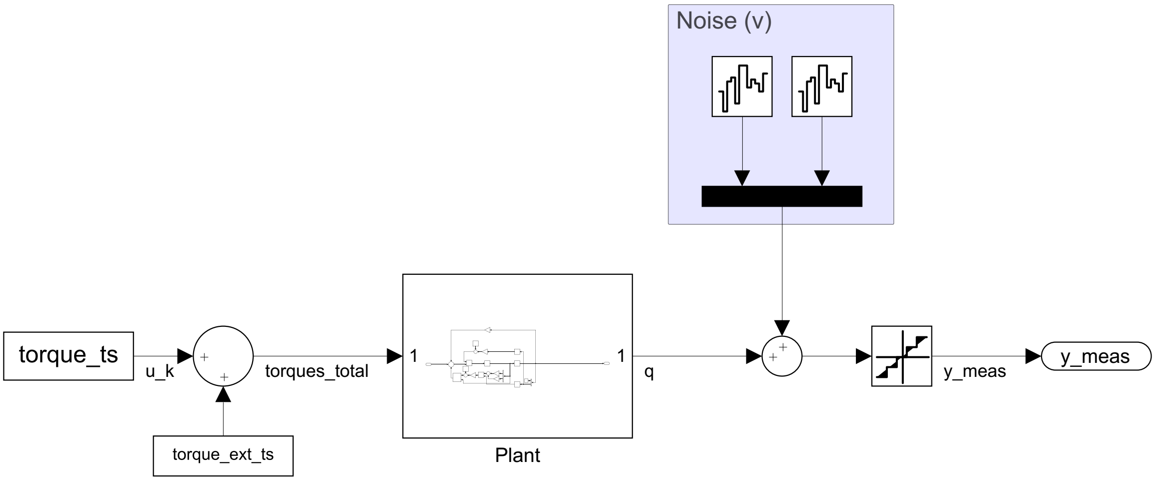


Figure 3.3: Implementation of the double pendulum model in Simulink

The implementation of the LKF in Simulink as discussed in section 3.3 is shown in figure 3.4. This Simulink model is in accordance with the definition of the Kalman filter 2.2 to 2.4, where the linearized and discretized augmented-state model derived in section 3.2.4 was used. The input signal u_k in the figure consists only of the known actuator forces, the external torques applied as input to the plant in figure 3.3 are not included in this signal. As such, the external torques are not 'known' by the LKF. The input signal y_{meas} represents the measurement signal as in figure 3.3. That is, it corresponds to the angular positions of the beams distorted by white noise and quantization.

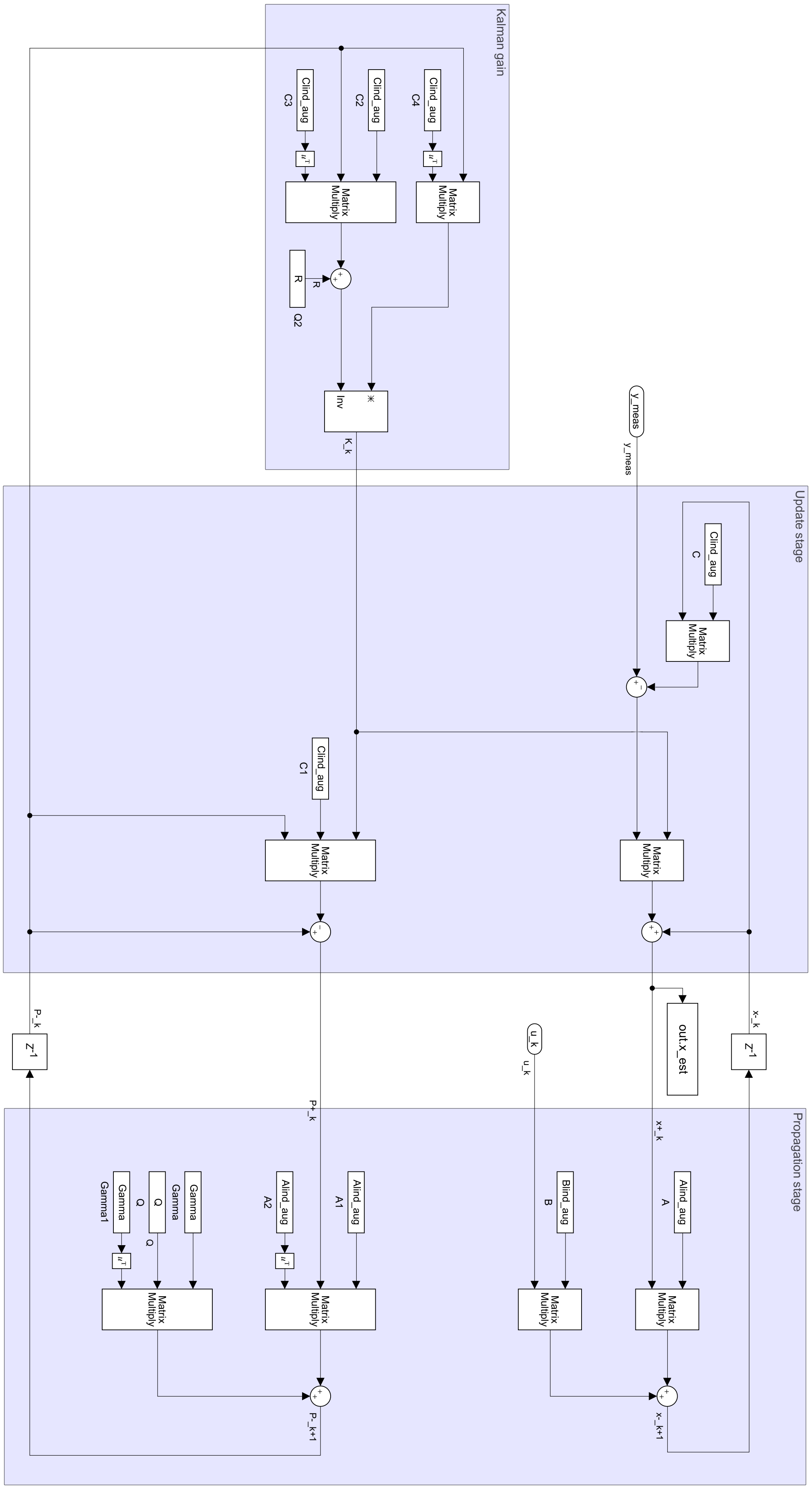


Figure 3.4: Implementation of the Kalman filter in Simulink

3.5 Simulations

Simulation conditions

The LKF has been tested in simulations under multiple conditions, with e.g. different actuator torque signals, external torque signals, and parameter errors. However, to make sure the results shown in the following sections can be compared, the conditions in which these were executed have been kept the same for the largest part. This includes the parameters of the plant, the actuator torques, external torques, and the noise signals. The general values used for these parameters will be discussed in this section. In the following sections, it will be indicated if any of the parameters are changed from these general values.

The magnitudes of the actuator and external torques are chosen such that the resulting ranges of the angular deflection are about 10°, because the 2DOF manipulator will have a similar range of motion. This will result in a comparable error caused by excluding the non-linearities in the model. The actuator torques will consist of two low frequency sinusoids, and the external torque signals will be built up by some low frequency sinusoids, an offset and a step at some point in time. To be precise, the actuator torque signals and the offset and frequency components of the external torque signals (so excluding the steps) are chosen as,

$$\begin{aligned} \tau_{act,1} &= 3\,\text{sin}\,(0.16t) + 3\,\text{sin}\,(0.48t)\,, \\ \tau_{act,2} &= 3\,\text{sin}\,(0.1t) + 3\,\text{sin}\,(0.66t)\,, \\ \tau_{ext,1} &= 3 + 3\,\text{cos}\,(0.15t) + 0.4\,\text{cos}\,(1.1t)\,, \\ \tau_{ext,2} &= 3 + 3\,\text{cos}\,(0.1t) + 0.3\,\text{cos}\,(0.9t)\,. \end{aligned} \tag{3.42}$$

Besides the initial step in the external torque (caused by the offset and cosines), there is an additional step added to the external torque on each of the arms. On the first arm, this external torque is applied at $t = 2.5s$ with a magnitude of $-5.4N$. On the second arm, this external torque is applied at $t = 5s$ with a magnitude of $3.6N$. The actuator and external torque signals are illustrated in figure 3.5. Choosing the torque signals in this way tests the performance of the LKF for low frequent external torques as well as steps. Moreover, the steps will induce higher frequency oscillations in the positions, which might be a challenge for the LKF as well.

Parameter	Value
$\sigma^2_{meas,1}$	10^{-8}
$\sigma^2_{meas,2}$	10^{-8}
d_{res}	$10^{-3}\ m$
T_s	$10^{-4}\ s$
m_1	$1.5\,kg$
m_2	$0.9\,kg$
L_1	$0.4\,m$
L_2	$0.3\,m$
k_1	$110\ {}^{Nm}/_{rad}$
k_2	$130\ {}^{Nm}/_{rad}$
d_1	$0.1\ {}^{Nm\cdot s}/_{rad}$
d_2	$0.1\ {}^{Nm\cdot s}/_{rad}$

Table 3.1: Simulation parameters for testing methods of setting LKF parameters

To set the sub-matrix $\boldsymbol{Q}_{d,F}$ as in (3.38), the external torque signals should be analyzed as indicated in section 3.3. The ranges where the change in the external torque signals for each time step are analyzed are shown in figure 3.6. These are the ranges where the signal is most 'dynamic', beside the steps. The steps are excluded because these could give an unreasonably

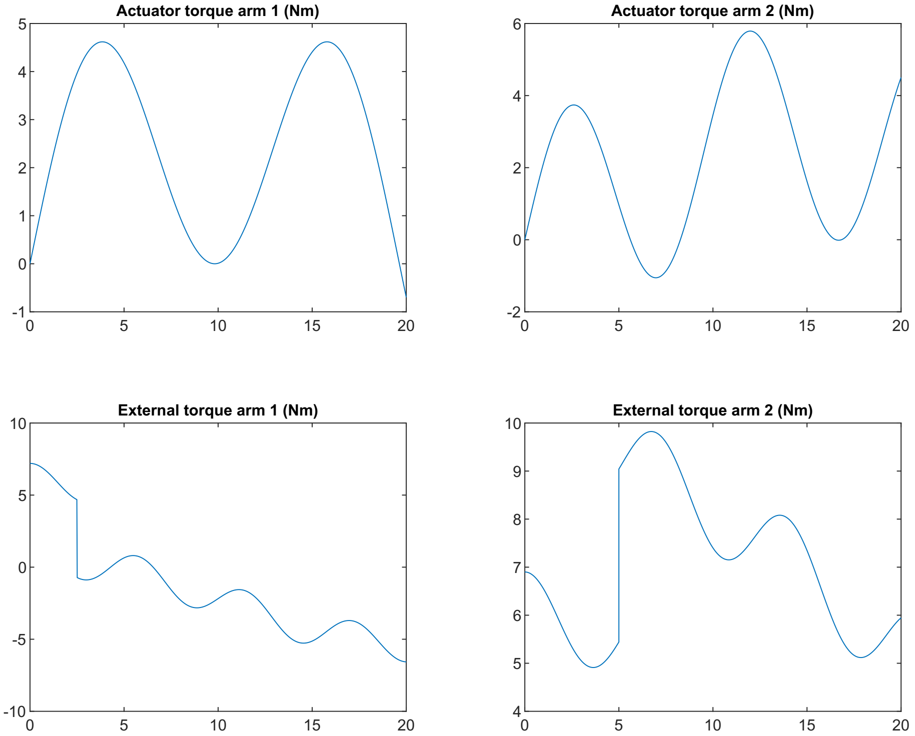


Figure 3.5: Actuator and external torque signals applied to double pendulum

high expected change in the external torque per time step (up to infinity, depending on the range).

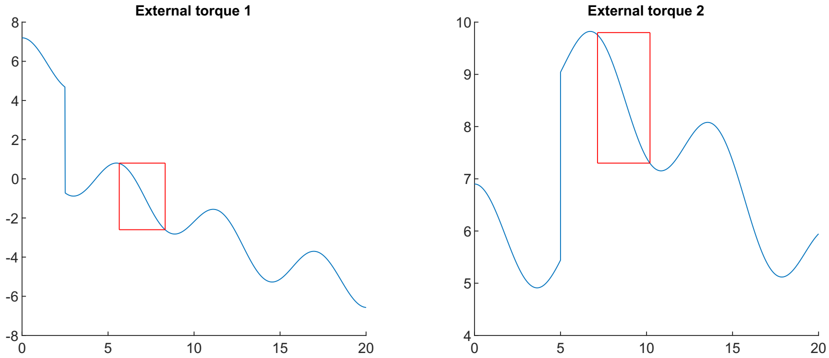


Figure 3.6: Measured displacement of the arms and the white noise applied

The remaining parameters used in the simulations are listed in table 3.1. These ensure that the conditions for these simulations are comparable but slightly worse than the conditions in which the experiments on the 2DOF manipulator will be executed. The measurement noise is slightly greater, the encoder resolution is slightly lower, and the sample time is higher. In figure 3.7, the resulting measurement signal can be seen, which will be used by the LKF. Figure 3.7 also shows the white noise that was applied to the measurement signal. In the following sections, the results of estimating the external force based on this signal will be discussed for different conditions.

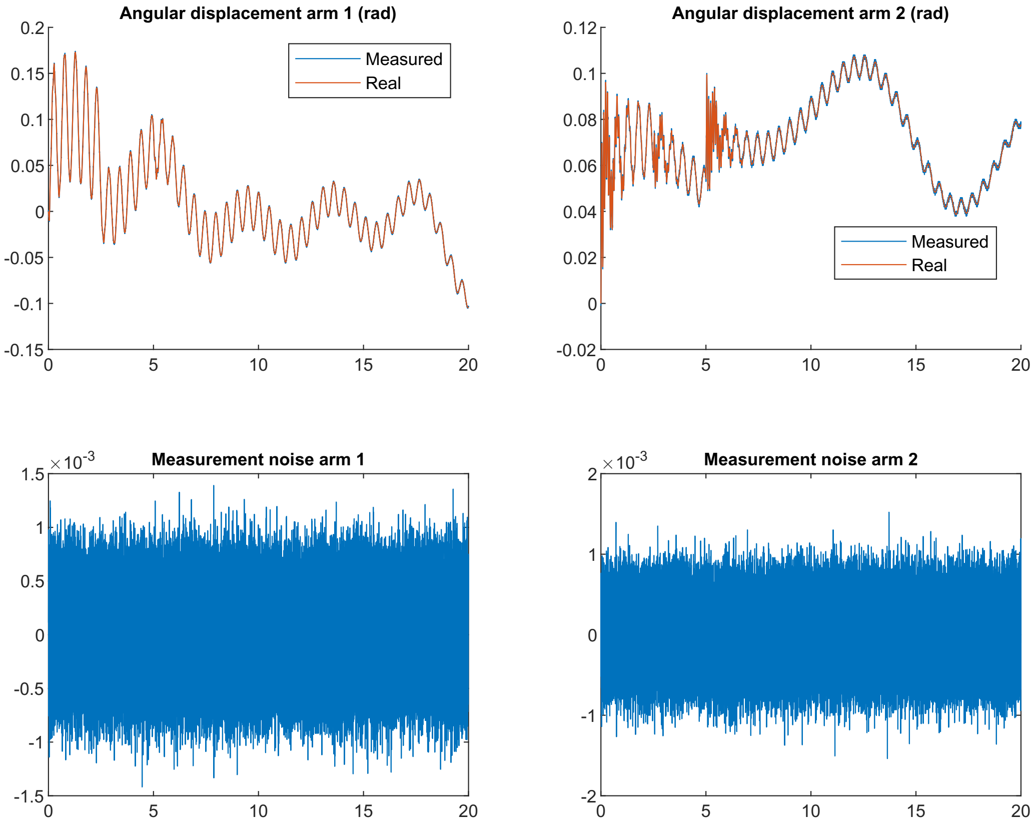


Figure 3.7: Measured displacement of the arms and the white noise applied

Although the initial state of the double pendulum is known exactly, it is deliberately set to the wrong value to test the effect of an error in the initial estimate $\boldsymbol{x}_0$. To indicate the uncertainty in the initial error, the initial state-estimate covariance matrix $\boldsymbol{P}_0$ is set to a high value as well. The error in the initial estimate and initial state-estimate covariance matrix are chosen as,

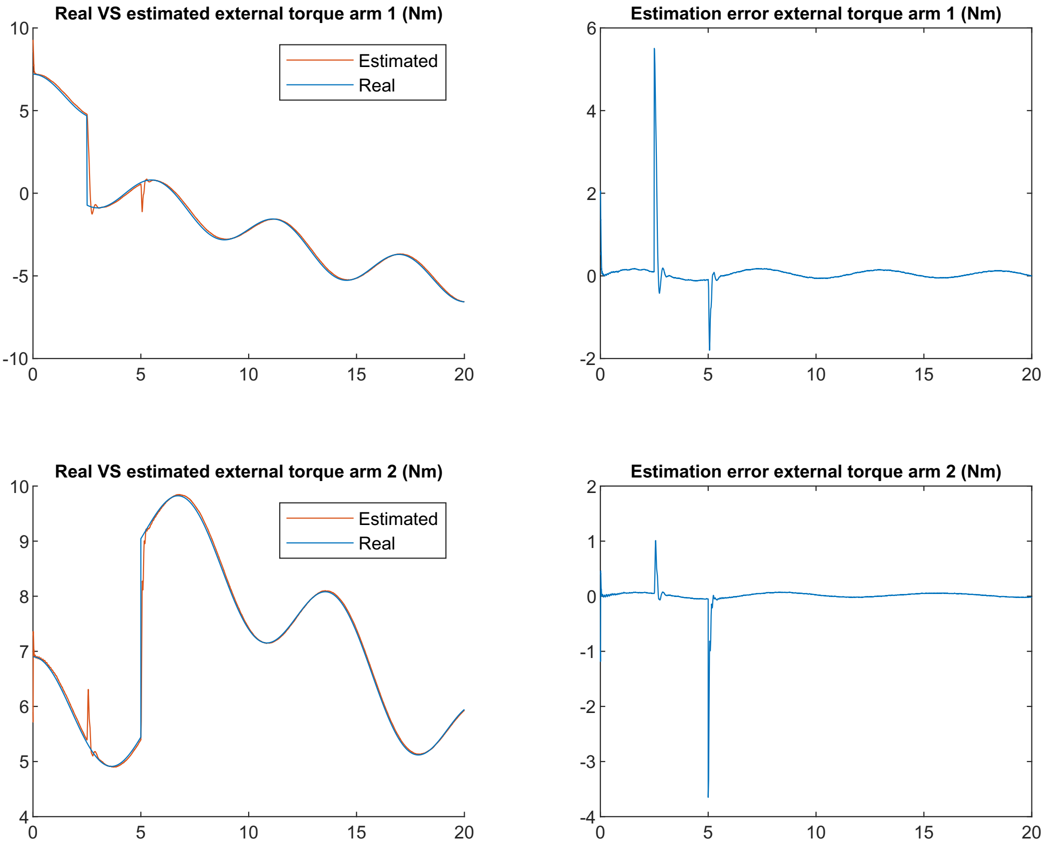
$$\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_0 + \begin{bmatrix} -1 \\ -2 \\ -0.1 \\ 0.15 \\ 2 \\ -1 \end{bmatrix} \quad , \quad \boldsymbol{P}_0 = 0.1 \cdot \boldsymbol{I}_{6\times 6}. \tag{3.43}$$

3.5.1 Simulations without parameter errors

To start, the LKF will be tested in simulations where there are no errors in the parameters. Then, the model used by the LKF should be (almost) identical to the plant. The simulations in this section should therefore give the best results that can be expected from the LKF.

Simulation results: LKF parameters set according to section 3.3

The results of running a simulation with the LKF parameters set exactly as indicated in section 3.3 are shown in figure 3.8.



Real VS estimated external torque arm 1 (Nm)

Estimation error external torque arm 1 (Nm)

Real VS estimated external torque arm 2 (Nm)

Estimation error external torque arm 2 (Nm)

Figure 3.8: Estimated external torques and estimation errors, parameters set as in 3.3

It can be seen that the resulting estimated torques converge to the real values of the external torques very well. The estimation errors oscillate slightly, which is caused by a delay in the estimates with respect to the real torques. This can be explained by the manner in which the external torques are included in the state-space model. The external torques are constant in the model, and the change in torques is represented by the noise $\boldsymbol{w}_{F,k}$. Then, the estimated torques will only be adjusted when enough data suggests that the true value of the torques has changed with respect to estimates at previous time steps, causing a delay.

The estimation error for each arm also spikes when a torque step is applied to the other arm. However, any estimation errors are corrected very quickly. Furthermore, the effects of neglecting the non-linear behavior of the double pendulum are barely noticeable. Despite that the torque estimates are based on a linear model, the LKF performance is very satisfactory in ideal circumstances.

Simulation results: scaled process noise covariance submatrix $\boldsymbol{Q}_{d,F}$

This subsection will compare the performance of the LKF for different values of the process covariance submatrix $\boldsymbol{Q}_{d,F}$, which relates to the noise representing the changes in the external torques $\boldsymbol{w}_{F,k}$. The results for different $\boldsymbol{Q}_{d,F}$ are tested by scaling the values found using section 3.3. It is noted that the process covariance submatrix $\boldsymbol{Q}_{d,par}$ is set to zero in these simulations, since there are no parameter errors included in these simulations.

Figure 3.9 shows the result of scaling $\boldsymbol{Q}_{d,F}$ by a factor of 10^{-3}. The estimated torques no longer follow the real torques fast enough. This makes sense, as the variance of the noise $\boldsymbol{w}_{F,k}$, which represents the changes in external torque, was reduced. Effectively, the equation for $\boldsymbol{\tau}_{k+1}$ in the state-space model was marked as more reliable. This equation represents the external torques as a constant, however. As a result, the LKF will not allow the estimated external torques to be adjusted as quickly as before. Even more, if $\boldsymbol{Q}_{d,F}$ would be set to zero, the estimated external torques would remain constant. Obviously this would not be an improvement to the performance of the LKF.

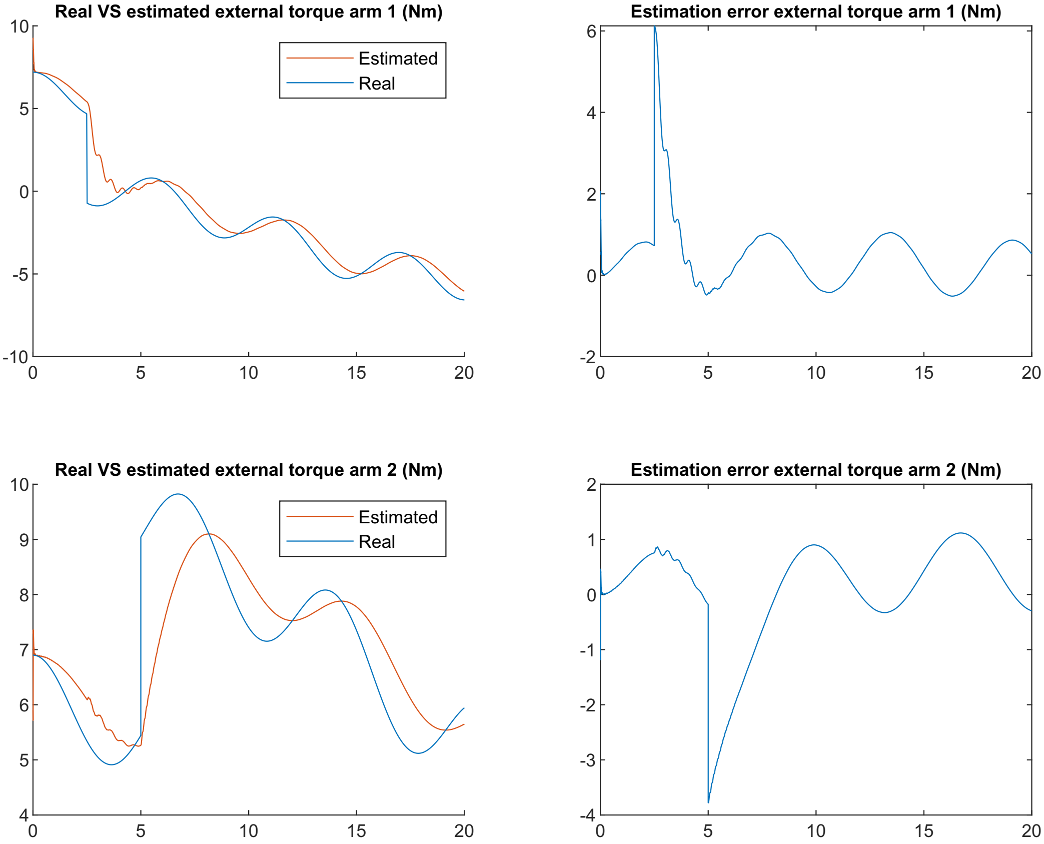


Figure 3.9: Estimated external torques and estimation errors, $\boldsymbol{Q}_{d,F}$ scaled by 10^{-3}

Figure 3.10 shows the result of scaling $\boldsymbol{Q}_{d,F}$ by a factor of 10^3. This action has introduced more noise into the estimated external torques, which is not an improvement to the performance of the LKF either. By increasing the process covariance submatrix $\boldsymbol{Q}_{d,F}$, the equation for $\boldsymbol{\tau}_{k+1}$ in the state-space model was marked as less reliable. While this allows the estimated external torques to be adjusted faster, this also makes the LKF rely more on the (noisy) measurement data.

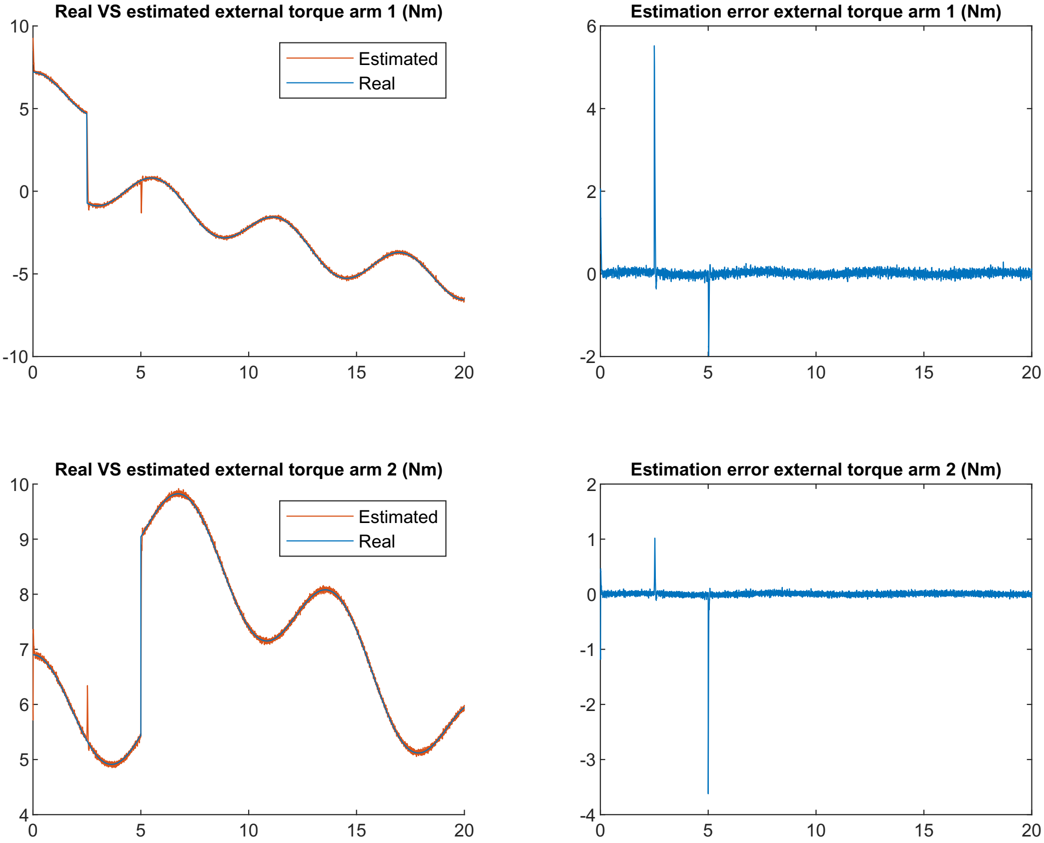


Figure 3.10: Estimated external torques and estimation errors, $\boldsymbol{Q}_{d,F}$ scaled by 10^3

In conclusion, the values found for $\boldsymbol{Q}_{d,F}$ using section 3.3 seem valid in ideal conditions. $\boldsymbol{Q}_{d,F}$ can be scaled to adjust the performance of the LKF if desired. The trade-off would be faster tracking of the external torque estimates at the cost of introducing more noise.

Simulation results: scaled measurement noise covariance matrix $\boldsymbol{R}_d$

In this section, the influence of varying the measurement noise covariance matrix $\boldsymbol{R}_d$ on the performance of the EKF is tested. The results for different $\boldsymbol{R}_d$ are tested by scaling the values found using section 3.3. Figure 3.11 shows the estimated torques found by scaling $\boldsymbol{R}_d$ by 10^{-3}, and figure 3.12 shows the estimated torques found by scaling $\boldsymbol{R}_d$ by 10^3.

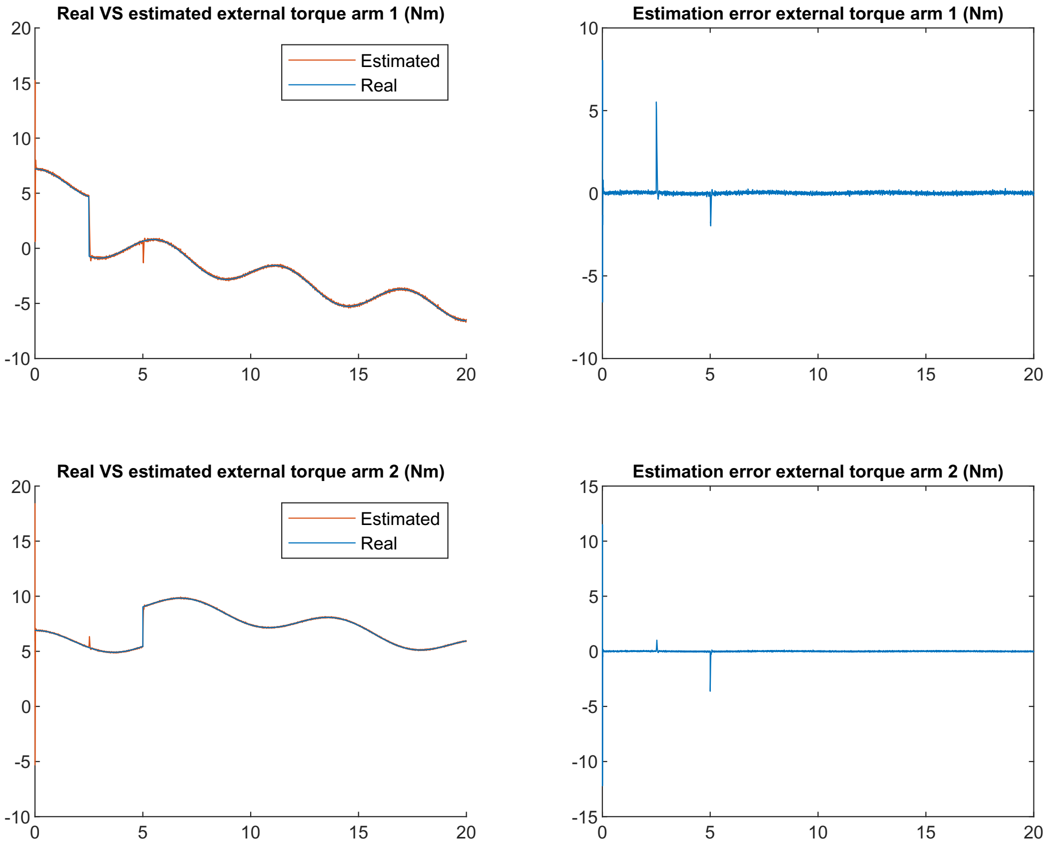


Figure 3.11: Estimated external torques and estimation errors, $\boldsymbol{R}_d$ scaled by 10^{-3}

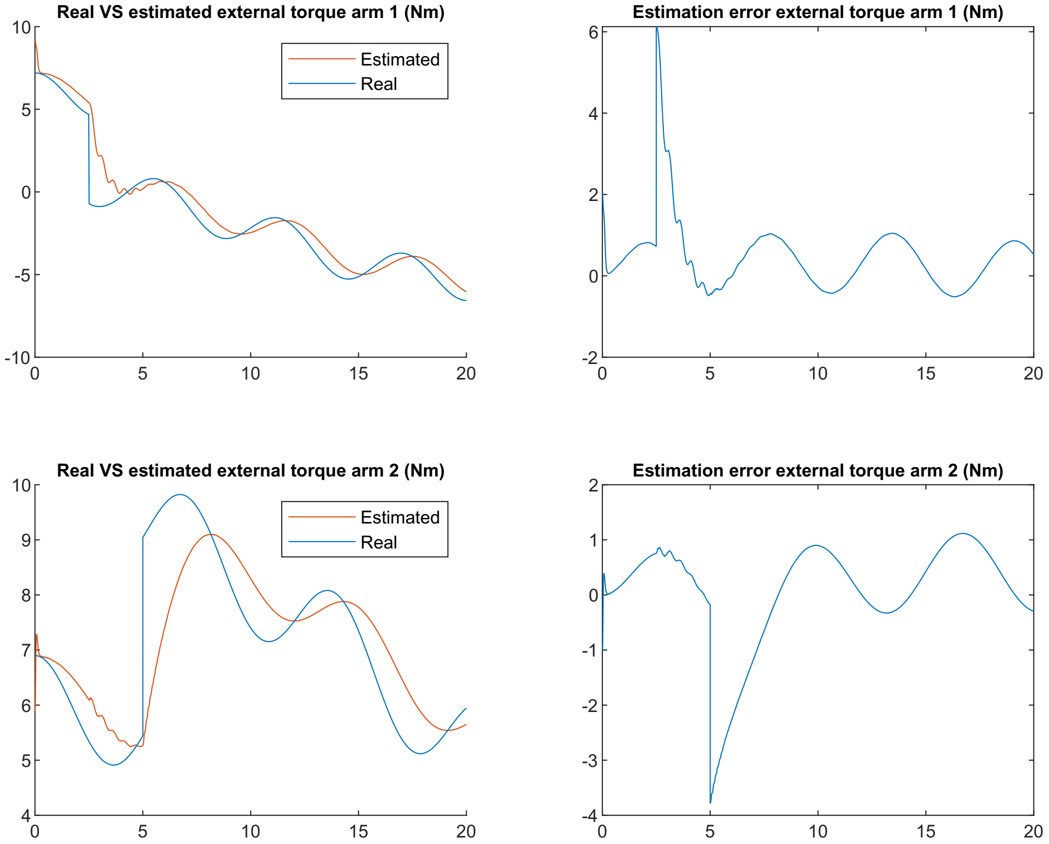


Figure 3.12: Estimated external torques and estimation errors, $\boldsymbol{R}_d$ scaled by 10^{3}

Figures 3.11 and 3.12 show almost exactly the same results as in figures 3.9 and 3.10, where $\boldsymbol{Q}_{d,F}$ was scaled by the same amounts. These covariance matrices indicate the reliability of the measurements and model respectively. What can be seen is that their absolute values are not necessarily too important. The relative values of these matrices are what mostly dictates the performance of the LKF. Based on their relative size, the LKF will rely more on either the measurement data or the model for its estimated states. Hence, scaling $\boldsymbol{R}_d$ by 10^3 or $\boldsymbol{Q}_{d,F}$ by 10^{-3} gives similar results, since this gives the same change in their relative sizes. The submatrix $\boldsymbol{Q}_{d,par}$ may also play a role, but not in these simulations as it was set to zero.

Simulation results: non-zero process noise covariance submatrix $\boldsymbol{Q}_{d,par}$

This subsection will test the performance of the LKF when the value of the process noise covariance submatrix $\boldsymbol{Q}_{d,par}$ is not set to zero. In the previous simulations, this matrix was set in accordance with section 3.3. Since there were no errors in the parameters in these simulations, so far this resulted in $\boldsymbol{Q}_{d,par} = \boldsymbol{0}$. In this section, $\boldsymbol{Q}_{d,par}$ will be given non-zero values despite that there are still no parameter errors included in the simulation. In doing so, the possible negative effects of including $\boldsymbol{Q}_{d,par}$ are investigated.

To start, figure 3.13 shows the results of setting $\boldsymbol{Q}_{d,par}$ as if there were 10% parameter errors. Compared to figure 3.8, the estimated torques follow the real external torques significantly slower. By including a non-zero $\boldsymbol{Q}_{d,par}$, the state-space equations for the velocities $\dot{\boldsymbol{\theta}}_{k+1}$ will be interpreted by the LKF as less reliable. As a result, the estimate for the states will rely more on the measurements as well as the other state-space equations. The estimated torques cannot directly be found from measurements, however. Instead, the LKF will base its estimates for the torques more on the state-space equations for $\boldsymbol{\tau}_{k+1}$. As discussed previously, these equations do not account for a possible change in the external torques. As a result, the estimated torques will track the real values significantly slower.

To illustrate the cause of the reduced performance, equation 3.44 shows the values used for $\boldsymbol{Q}_{d,par}$ and $\boldsymbol{Q}_{d,F}$ in figure 3.13. While $\boldsymbol{Q}_{d,F}$ was intentionally set to allow the estimated external torques to vary, $\boldsymbol{Q}_{d,par}$ has significantly higher values. Because the values in $\boldsymbol{Q}_{d,F}$ are low in comparison, the LKF will rely more on the corresponding state-space equation: $\boldsymbol{\tau}_{k+1} = \boldsymbol{\tau}_k$. Again, this prevents the external torque estimates to be adjusted quickly.

$$\boldsymbol{Q}_{par} = 10^{-5} \cdot \begin{bmatrix} 0.0466 & 0 \\ 0 & 0.4720 \end{bmatrix} \quad , \qquad \boldsymbol{Q}_F = 10^{-5} \cdot \begin{bmatrix} 0.0023 & 0 \\ 0 & 0.0008 \end{bmatrix} \tag{3.44}$$

In an attempt to solve this problem, the process noise submatrix $\boldsymbol{Q}_{d,F}$ is scaled up. The results for scaling $\boldsymbol{Q}_{d,F}$ by 100, 1000, and 10000 are shown in figures 3.14, 3.15, and 3.16. As expected, the estimated external torques track the real values significantly faster as a result of the scaling. On the other hand, the amount of noise in the estimates increases as well.

In conclusion, by adding non-zero values for $\boldsymbol{Q}_{d,par}$, the performance of the LKF has deteriorated. This may still be necessary when the model actually has errors in its parameters, however. These simulations have also shown that it may be detrimental to overestimate the inaccuracy of the model. Furthermore, the simulations have also shown that care should be that the values for $\boldsymbol{Q}_{d,F}$ are (significantly) larger than those of $\boldsymbol{Q}_{d,par}$. Otherwise the risk exists that the estimates for the external torques are not allowed to be adjusted sufficiently fast.

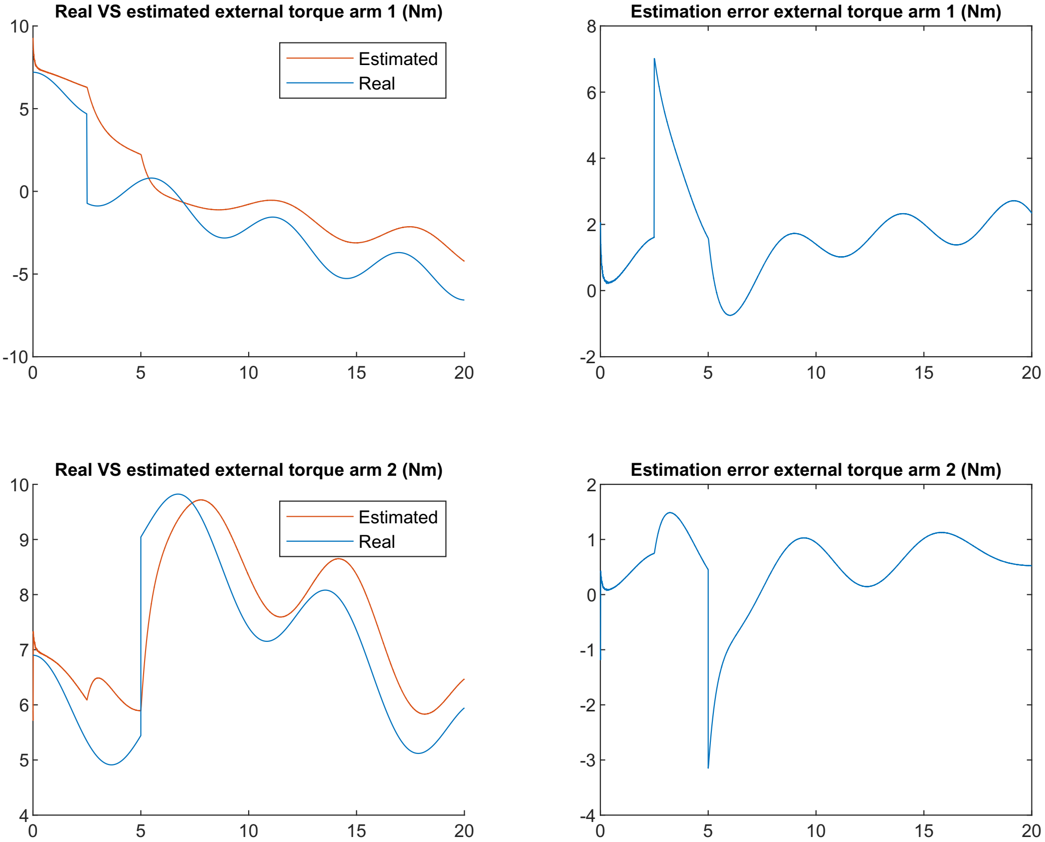


Figure 3.13: Estimated external torques and estimation errors, $\boldsymbol{Q}_{par}$ set for 10% parameter errors

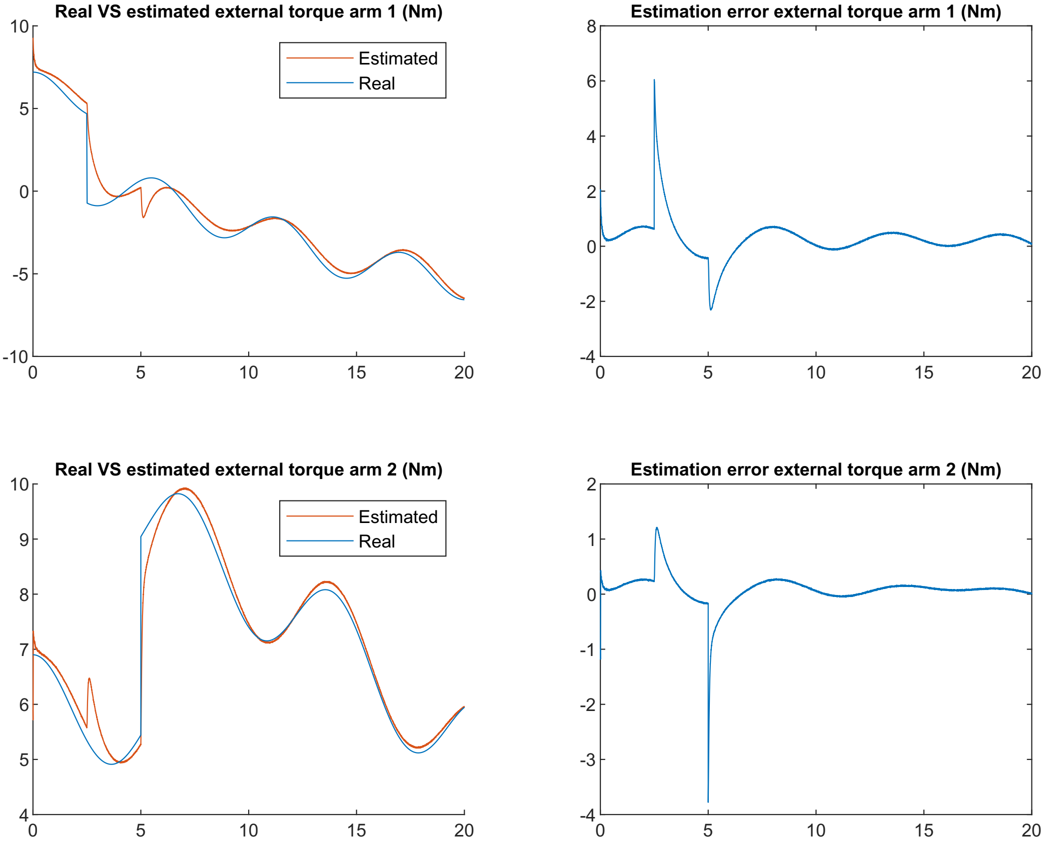


Figure 3.14: Estimated external torques and estimation errors, $\boldsymbol{Q}_F$ scaled by 10^2

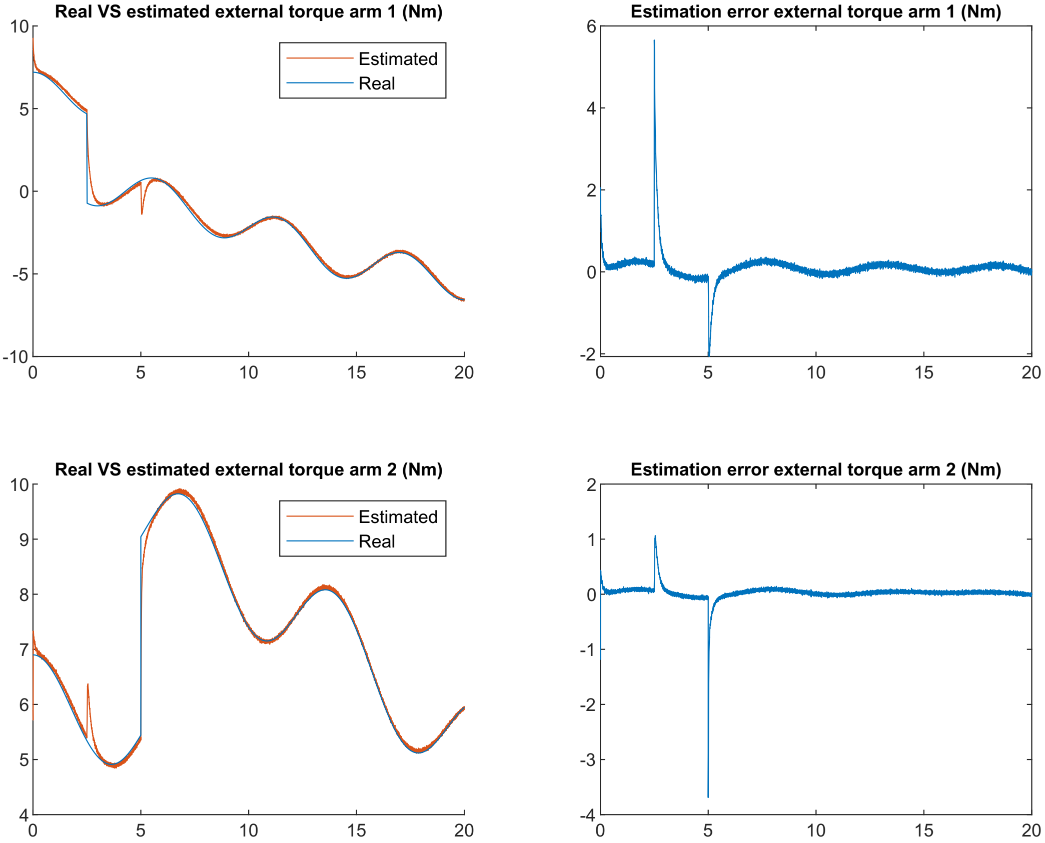


Figure 3.15: Estimated external torques and estimation errors, Q_F scaled by 10^3

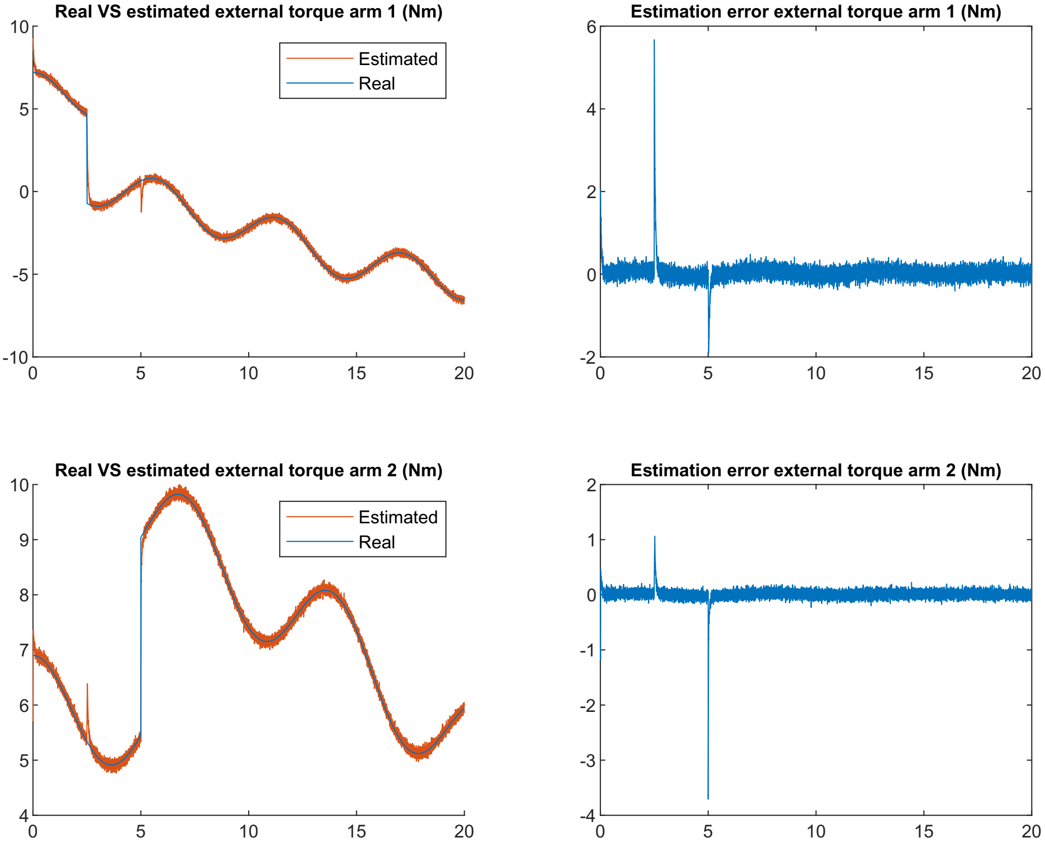


Figure 3.16: Estimated external torques and estimation errors, Q_F scaled by 10^4

3.5.2 Simulations with 10% parameter errors

In this section, the performance of the LKF will be tested when parameter errors are introduced. This will be achieved by generating a model based on randomly adjusted parameters. Values for the parameters are randomly generated within a range of 10% of the real parameter values (as used by the plant). These parameters are then applied to the state-space model defined in section 3.2.4, and the resulting 'wrong' model will be applied to the LKF as described in section 2.2. To be able to compare the results of each simulation in this section, the values for the parameters were randomly generated once and loaded each simulation thereafter. Different sets of randomized parameters have also been tried for the same conditions, which gave similar results. The randomly generated error factors used in these simulations for the parameters are shown in table 3.2.

Parameter	Error factor
m_1	1.0180
m_2	1.0517
k_1	0.9513
k_2	0.9702
d_1	0.9279
d_2	0.9760

Table 3.2: Factor by which the parameters differ from their real values

To start, the performance of the LKF is tested when its parameters are set according to section 3.3, but with $\boldsymbol{Q}_{d,F}$ scaled by 10^3 to allow the estimated external torques to be adjusted fast enough (as discussed in section 3.5.1). The resulting estimates are shown in figure 3.17.

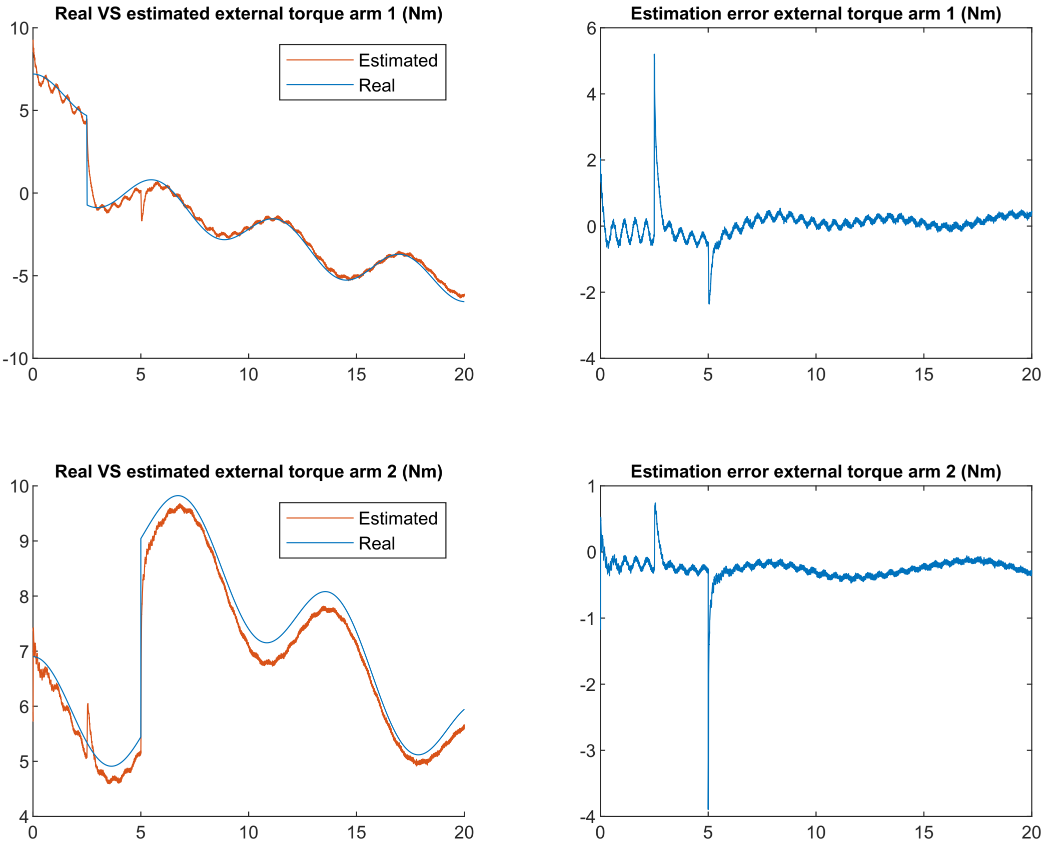


Figure 3.17: Estimated external torques with 10% parameter errors

Clearly, the parameter errors have deteriorated the performance of the LKF. After each torque step the estimated torques oscillate around their real values, and a constant deviation from the real values can also be recognized. These errors showing up in the estimated external torques can be explained. In this simulation, the LKF is based on a model with wrong parameters. As a result, the external torques estimated by the LKF do not represent the external torques that, when applied to the real system, would lead to the measured displacements. Instead, these external torques represent those that would lead to these displacements when applied to the system represented by the erroneous model. As a result, the estimates show 'fictitious' external torques, caused by the difference in the dynamic behavior of the real system with respect to the system represented by the erroneous model.

This is further illustrated in figure 3.18, where the actuator torques were set to zero, and only the steps in the external torque signals remain. Beside the oscillations in the estimated torques, a constant estimation error can be recognized after the steps. This constant error can be attributed to the error in the stiffness parameters k_i used by the LKF. As can be seen from table 3.2, the values of the stiffness parameters used by the LKF are lower than the real values. As a result, the LKF underestimates the external torques. The oscillations can be attributed to errors in the parameters as well: the modelled system has different resonance frequencies than the real system. Hence, 'fictitious' external torques are estimated because of the difference in their dynamic behavior. To illustrate this, figure 3.19 compares the Bode plots of the model with true parameters and the model with erroneous parameters.

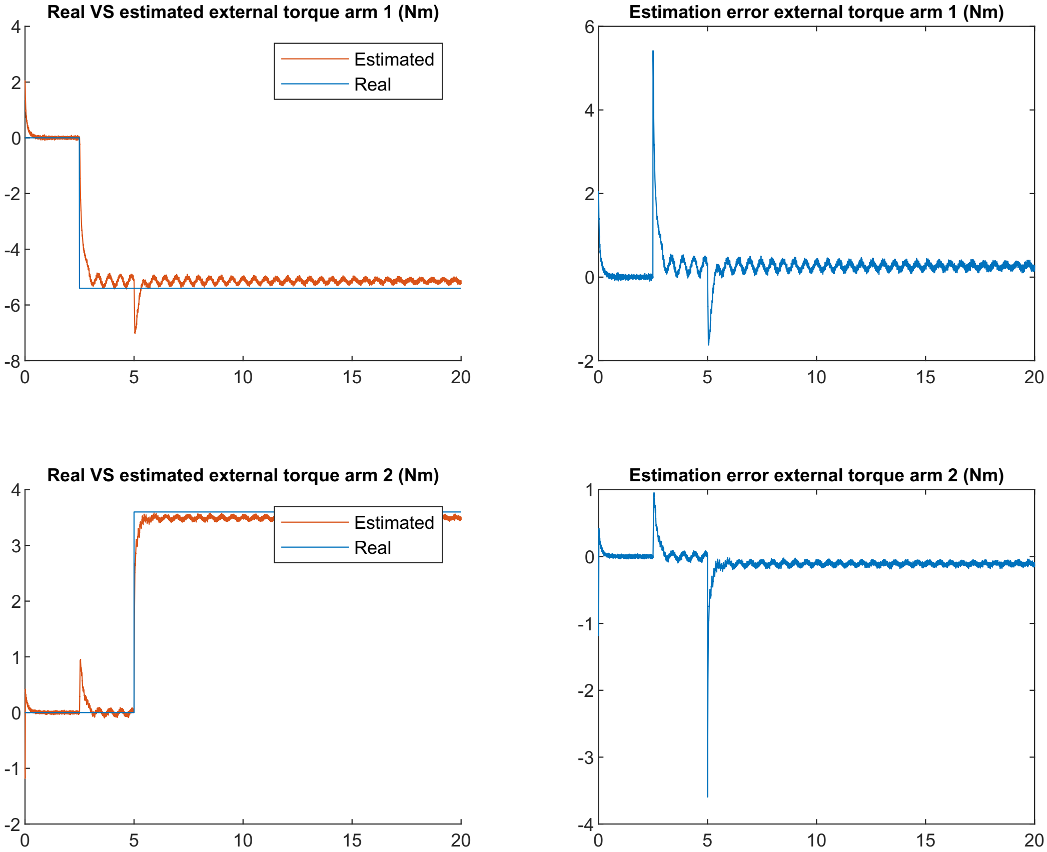


Figure 3.18: Estimated external torques with only steps in τ_{ext} acting on the system

If the actuator torques are reintroduced in the simulation, the estimates of the external torques deteriorate further. This is illustrated in figure 3.20. The system is excited by the actuators, but the LKF attributes some of the dynamic response this causes to the external torques. Again, this is caused by the erroneous model used by the LKF.

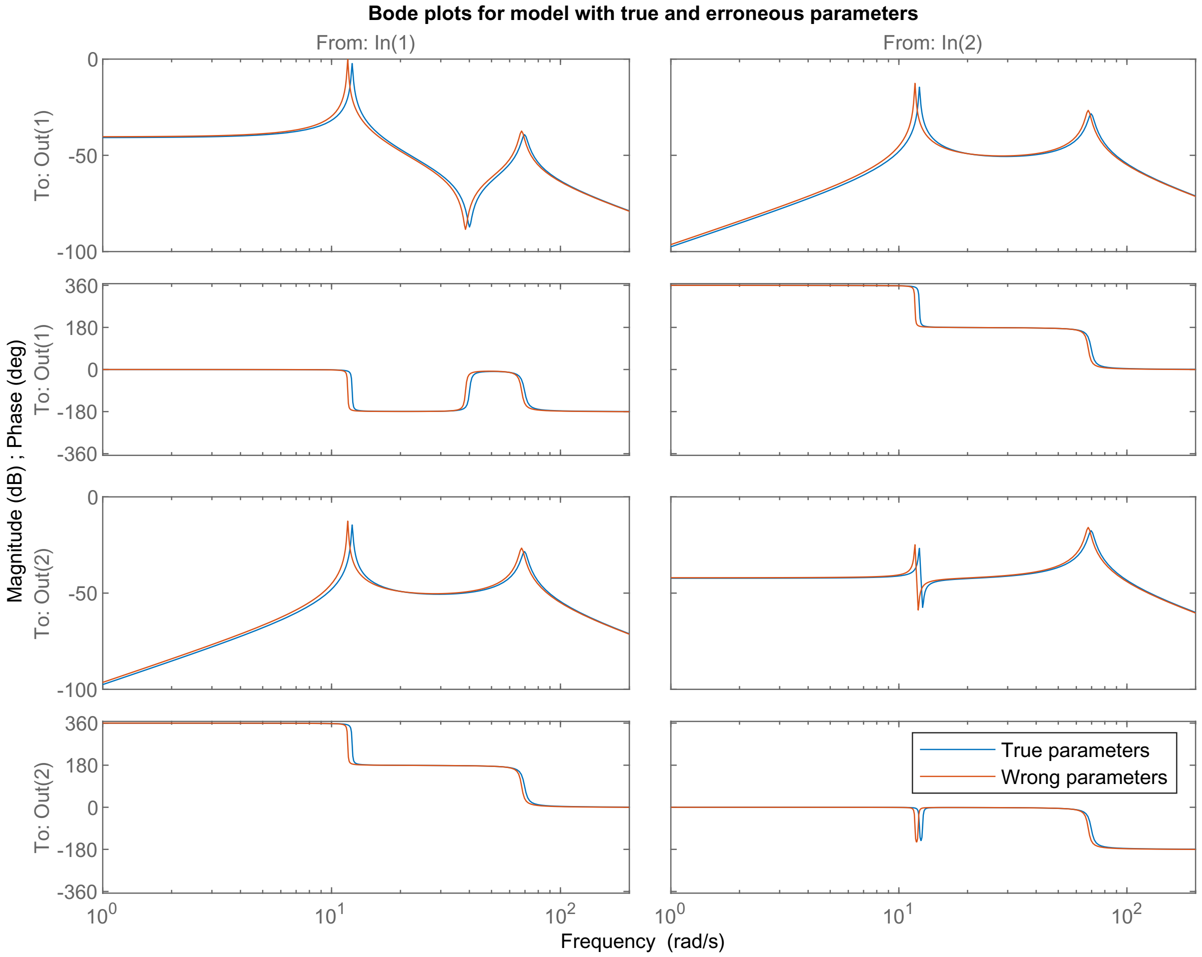


Figure 3.19: Bode plots of the double pendulum model with true and erroneous parameters

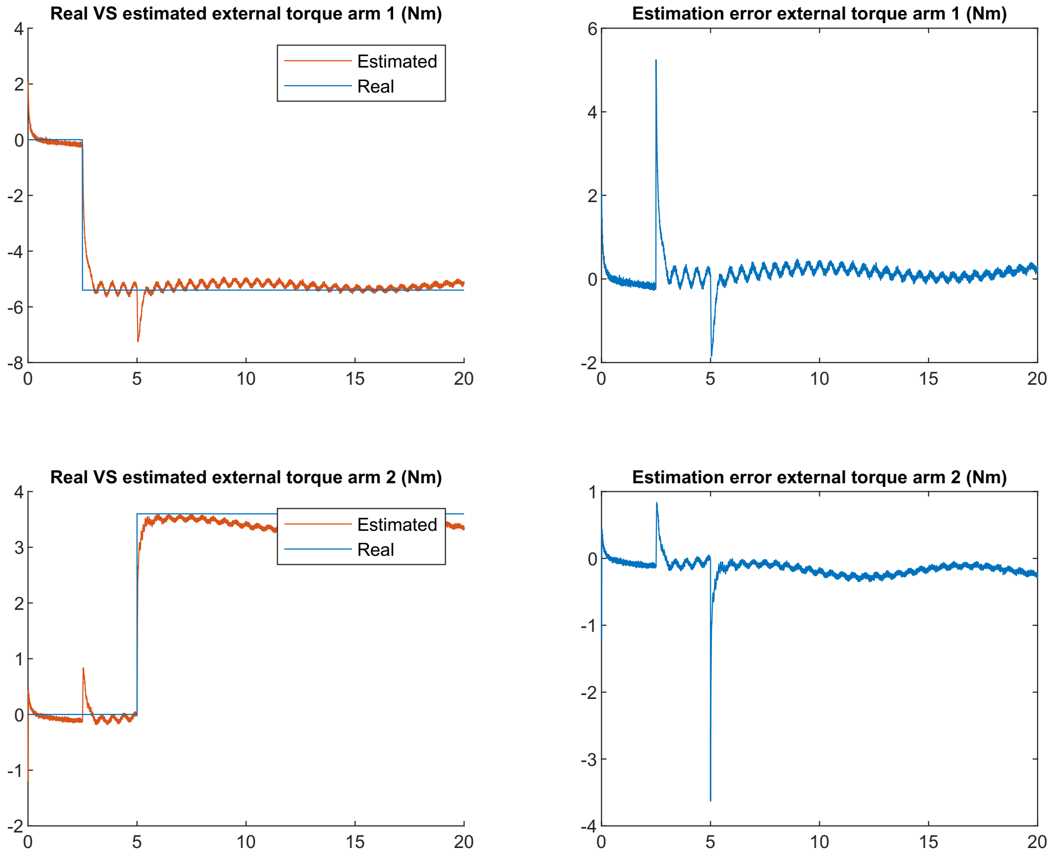


Figure 3.20: Estimated external torques with steps and actuator torques

It may be attempted to improve the estimates by increasing the scaling of $\boldsymbol{Q}_{d,par}$, indicating that the equation of motion are less reliable. As discussed in section 3.5.1, $\boldsymbol{Q}_{d,F}$ should then be scaled as well as otherwise the external torque estimates will respond too slowly to changes. The results of scaling $\boldsymbol{Q}_d$ by 10 and 100 are shown in figures 3.21 and 3.22. Unfortunately, the only effect of scaling the process noise covariance matrix $\boldsymbol{Q}_d$ is that the estimates become more noisy. Besides the increased amount of noise, the same estimation errors can be recognized. Again, this is caused by the change in relative size between the covariance matrices $\boldsymbol{R}_d$ and $\boldsymbol{Q}_d$. The LKF will rely more on the measurements for its estimates, making the results more noisy.

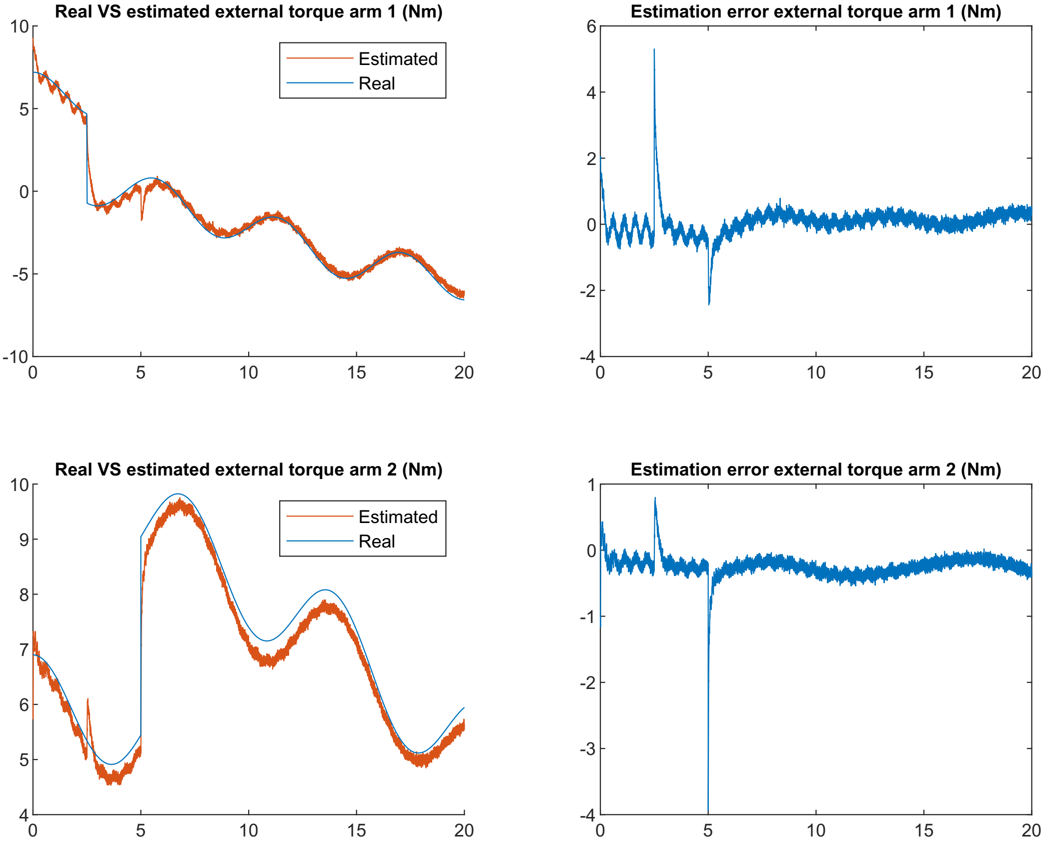


Figure 3.21: Estimated external torques and estimation errors, Q_d scaled by 10

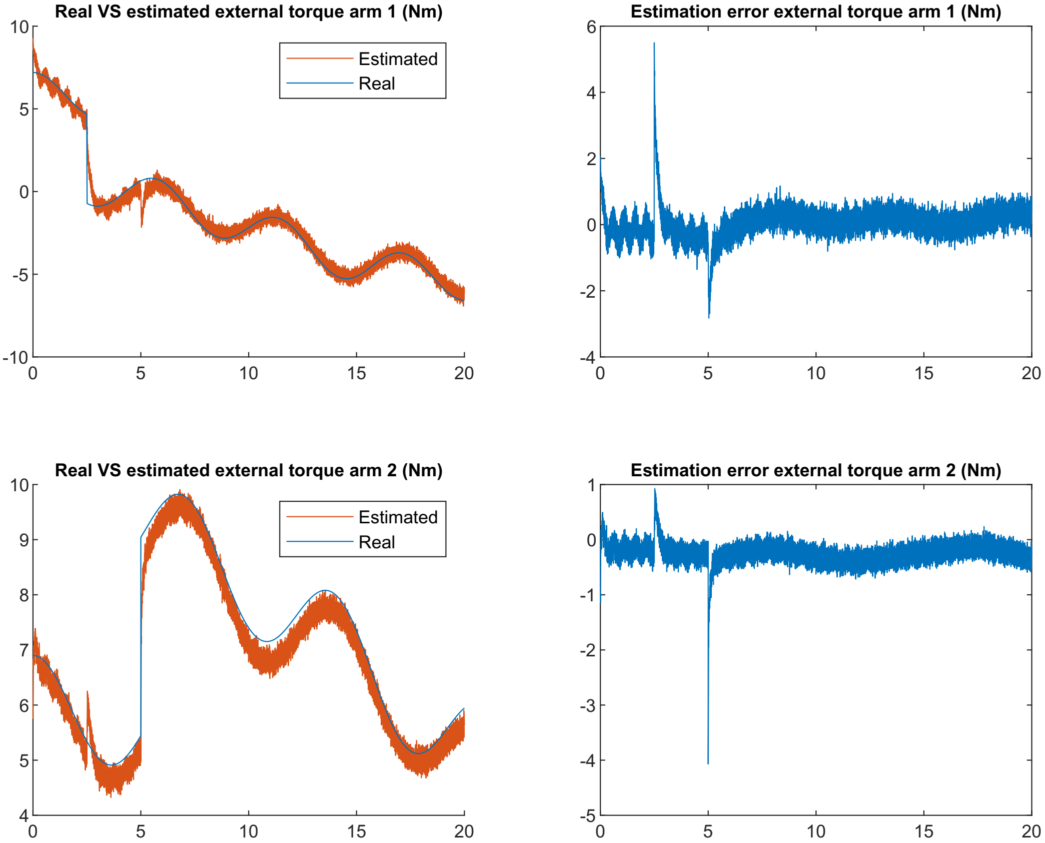


Figure 3.22: Estimated external torques and estimation errors, Q_d scaled by 100

Then, it may be wondered whether including non-zero values for $\boldsymbol{Q}_{d,par}$ has helped at all. Figure 3.23 shows the results for setting $\boldsymbol{Q}_{d,par}$ to zero. The estimated torques are almost identical to those found with the original values for $\boldsymbol{Q}_{d,par}$ shown in figure 3.17. To illustrate the differences, table 3.3 shows the mean value of the absolute error of the estimated states in both cases. Additionally, the table shows the mean of the absolute estimation errors for when the entire $\boldsymbol{Q}$ matrix was scaled by 0.1. Table 3.3 and figure 3.23 show that including $\boldsymbol{Q}_{d,par}$ barely makes a difference.

States	$\boldsymbol{Q}_{d,par}$	$\boldsymbol{Q}_{d,par} = \mathbf{0}$	$0.1 \cdot \boldsymbol{Q}_d$
θ_1	$4.99 \cdot 10^{-5}$	$4.34 \cdot 10^{-5}$	$5.20 \cdot 10^{-5}$
θ_2	$6.57 \cdot 10^{-5}$	$5.95 \cdot 10^{-5}$	$6.46 \cdot 10^{-5}$
$\dot{\theta}_1$	$5.9 \cdot 10^{-3}$	$4.2 \cdot 10^{-3}$	$5.5 \cdot 10^{-3}$
$\dot{\theta}_2$	$12.3 \cdot 10^{-3}$	$9.1 \cdot 10^{-3}$	$9.8 \cdot 10^{-3}$
$\tau_{ext,1}$	0.271	0.257	0.273
$\tau_{ext,2}$	0.258	0.274	0.256

Table 3.3: Means of the absolute estimation errors

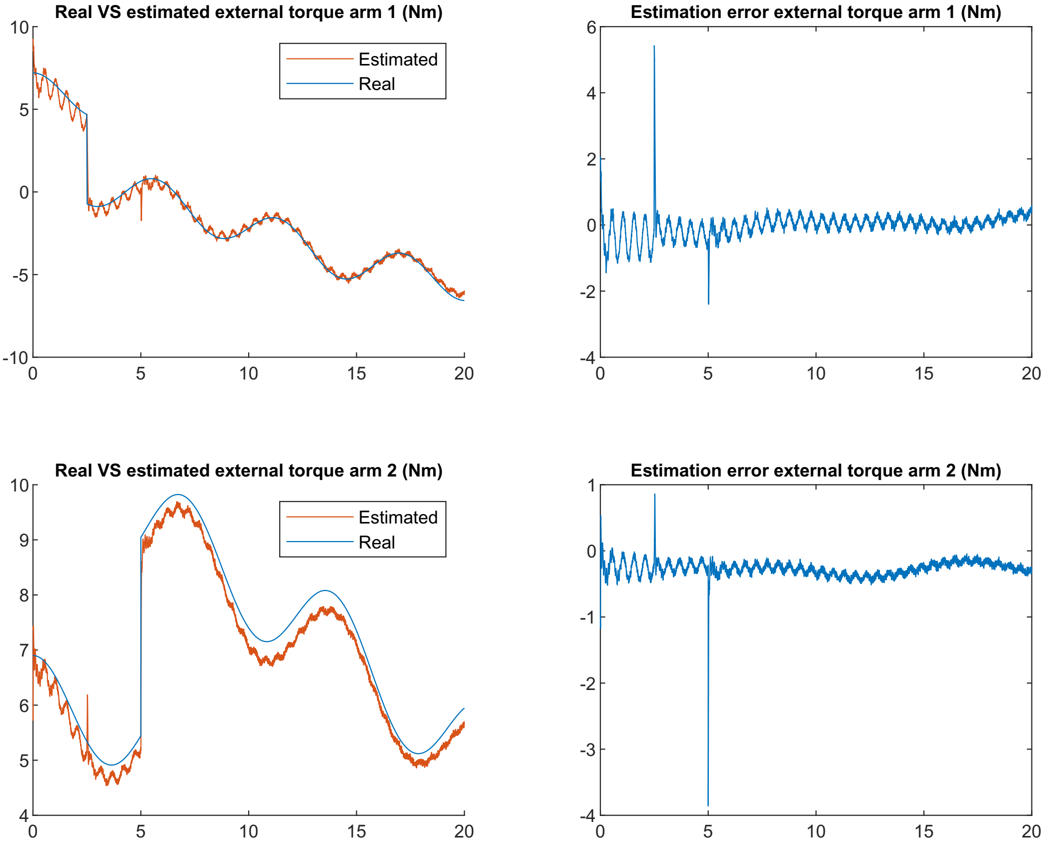


Figure 3.23: Estimated external torques and estimation errors, $\boldsymbol{Q}_{d,par}$ set to 0

In conclusion, any errors in the model will deteriorate the estimation of the external torques, because any unexpected dynamic behavior (caused by using a wrong model) will be attributed to the external torques. Unfortunately, this problem cannot be solved by including $\boldsymbol{Q}_{d,par}$. The model is the only source based on which the external torques can be estimated, since there is no feedback from a sensor. As such, it is critical that the model represents the real system accurately.

3.5.3 Simulation with 10% parameter errors and velocity sensors

In the previous section, the impact of errors in the model's parameters were investigated. The parameter errors deteriorate the quality of the estimates, and they cannot be improved significantly by tuning the LKF parameters. This section investigates if the estimates can be improved if a velocity sensor would be included in the setup.

The velocity sensors are included in the same way as the position sensors: they are influenced by white noise and quantization. The measurement covariance matrix is adjusted to include the measured velocities in the same fashion as the position sensors in section 3.3. The state-space matrix $\boldsymbol{C}_d$ has also been adjusted to include the velocities. The updated matrices become,

$$\boldsymbol{R}_d = \begin{bmatrix} \sigma^2_{quant,v1} + \sigma^2_{rest,v1} & 0 & 0 & 0 \\ 0 & \sigma^2_{quant,v2} + \sigma^2_{rest,v2} & 0 & 0 \\ 0 & 0 & \sigma^2_{quant,v3} + \sigma^2_{rest,v3} & 0 \\ 0 & 0 & 0 & \sigma^2_{quant,v4} + \sigma^2_{rest,v4} \end{bmatrix},$$

$$\boldsymbol{C}_d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{3.45}$$

Besides the changes to $\boldsymbol{R}_d$ and $\boldsymbol{C}_d$, the LKF parameters are set according to section 3.3 as usual. The resulting estimated torques are shown in figure 3.24. By adding velocity sensors, the estimated velocities have become much less noisy. As a result, the estimated external torques have become less noisy as well. Unfortunately, the same estimation errors caused by using an inaccurate model still remain.

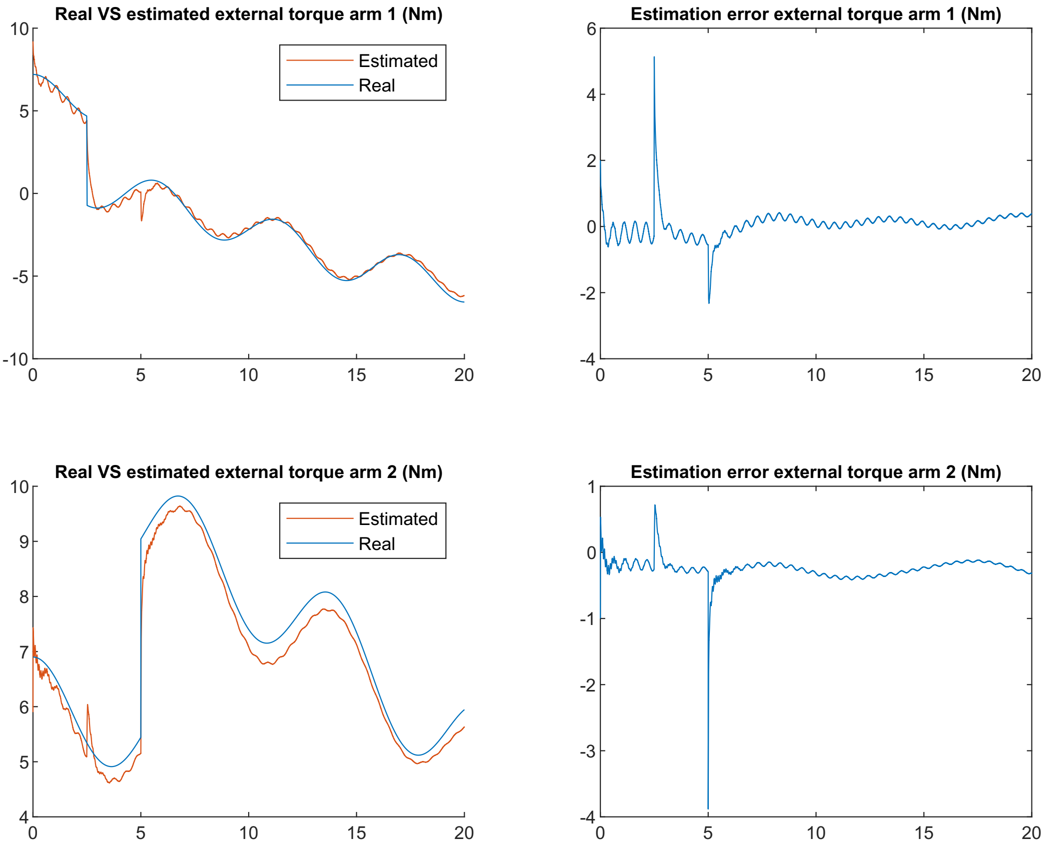


Figure 3.24: Estimated external torques and estimation error, with velocity sensors

3.6 Discussion: double pendulum testing

In this chapter methods for setting up the LKF parameters were suggested in section 3.3, and tested in section 3.5. The performance of the LKF was tested on a double pendulum in ideal circumstances, as well as with the presence of parameter errors in the model used in the LKF.

The methods used to set the LKF parameters discussed in section 3.3 lead to good results in ideal circumstances. Care must be taken that the values for the submatrix $\boldsymbol{Q}_{d,F}$ are larger than those of submatrix $\boldsymbol{Q}_{d,par}$. Otherwise, the external torque estimates may not be adjusted quickly enough by the LKF. It was also found that the performance of the LKF heavily depends on the correctness of the model used. When parameter errors are present, the LKF will include 'fictitious' forces in its estimates, caused by the difference in the dynamic behavior of the modelled system with respect to the real system. There is no solution that makes the detrimental effects of parameter errors disappear. Including the process noise $\boldsymbol{w}_{par,k}$ barely made any difference, and adding a velocity sensor did not significantly improve the estimates either. The model plays a very important role to the performance of the LKF, since this is the only source based on which the external torques are estimated.

The effects of neglecting the non-linear behavior of the double pendulum in the model used by the LKF were not significant. In ideal circumstances, though without including the non-linear dynamic behavior, the LKF still showed excellent performance. Based on the results in this chapter, applying a LKF to the 2DOF manipulator seems promising, although the performance will heavily depend on the accuracy of the model.

4 2DOF MANIPULATOR

In this chapter, the performance of the LKF will be tested in experiments on a real 2DOF manipulator system. As opposed to the double pendulum in chapter 3, the 2DOF manipulator will be tested in real world conditions. The 2DOF manipulator is a real system, where unknown noise sources and undetermined dynamic effects may play a role. While the dynamic behavior of the double pendulum was known exactly, this cannot be stated for the 2DOF manipulator.

The chapter will start by summarizing important properties of the 2DOF manipulator setup. Then, a model of the 2DOF manipulator is found based on the work of Aarts [11]. This includes a derivation of the equations of motion and the identification of the system's parameters. Next, the LKF parameters are set based on the model and data measured from experiments. Finally, the LKF is tested on the real 2DOF manipulator setup in different conditions.

4.1 2DOF manipulator overview

To recapitulate, the 2DOF manipulator discussed in section 1.2 consists of a set of parallel arms and a single arm which are made up of beams connected by flexure hinges. A schematic overview of the 2DOF manipulator is shown in figure 4.1. In the figure, the flexure hinges are numbered 1-11 and the beams are indicated with the letters A-G. The center of the end-effector is indicated with 'eff', and its workspace is shown by a red dashed outline as well. The encoders and actuators are fixed to beams C and A at a distance $r_{enc} = 0.251m$ and $r_{act} = 0.235m$ respectively. These measure distance/exert force along the circular path perpendicular to the beams, indicated with black arrows. Table 4.1 shows the coordinates of the hinges and end-effector when the system is in its nominal position [1].

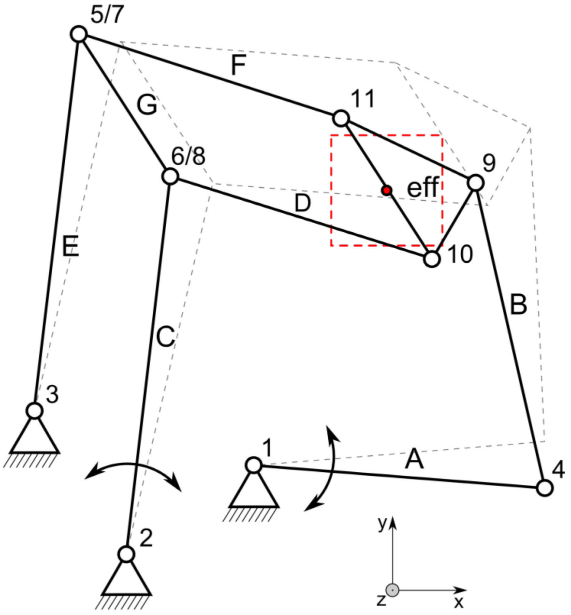


Figure 4.1: Schematic overview [1]

Hinge	x-position (m)	y-position (m)
1	0.27988	0.14908
2	0.16427	0.06801
3	0.08093	0.19738
4	0.54385	0.12809
5	0.12031	0.54114
6	0.20365	0.41177
7	0.12031	0.54114
8	0.20365	0.41177
9	0.48029	0.40561
10	0.44167	0.33531
11	0.35833	0.46468
eff	0.4	0.4

Table 4.1: Hinge coordinates nominal position [1]

4.2 Modelling

The system identification and modelling process on the 2DOF manipulator in this section was performed by Aarts [11]. The results have not been published, but were exchanged in private correspondence. Since these were not published, the process is repeated in this section.

4.2.1 Overview of the generalized coordinate approach

This section contains a brief overview of the method used to define the equations of motion, the generalized coordinate approach [12]. In this method, the dynamics are evaluated from the perspective of each individual body, and then reflected to a set of independent coordinates $\boldsymbol{q}$ equal to the number of degrees of freedom. A set of generalized coordinates $\boldsymbol{x}$ is defined, which describe the positions and orientations of the individual bodies. The relation between these generalized coordinates $\boldsymbol{x}$ and the independent coordinates $\boldsymbol{q}$ are defined by the geometric transfer function

$$\boldsymbol{x} = \boldsymbol{\mathcal{F}}(\boldsymbol{q}). \tag{4.1}$$

Expressions for the velocities and accelerations of the generalized coordinates as a function of the independent coordinates $\boldsymbol{q}$ are found from the time derivative of (4.1), which become

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{\mathcal{F}}_{,q}(\boldsymbol{q})\,\dot{\boldsymbol{q}}, \\ \ddot{\boldsymbol{x}} &= \boldsymbol{\mathcal{F}}_{,q}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \left(\boldsymbol{\mathcal{F}}_{,qq}(\boldsymbol{q})\,\dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}}, \end{aligned} \tag{4.2}$$

where the Jacobian $\boldsymbol{\mathcal{F}}_{,q}(\boldsymbol{q})$ can be found as

$$\boldsymbol{\mathcal{F}}_{,q}(\boldsymbol{q}) = \begin{bmatrix} \frac{\partial \mathcal{F}_1(\boldsymbol{q})}{\partial q_1} & \cdots & \frac{\partial \mathcal{F}_1(\boldsymbol{q})}{\partial q_{n_{dof}}} \\ \vdots & & \vdots \\ \frac{\partial \mathcal{F}_{n_x}(\boldsymbol{q})}{\partial q_1} & \cdots & \frac{\partial \mathcal{F}_{n_x}(\boldsymbol{q})}{\partial q_{n_{dof}}} \end{bmatrix}, \tag{4.3}$$

and the product $\left(\boldsymbol{\mathcal{F}}_{,qq}(\boldsymbol{q})\,\dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}}$ can be found as

$$\left(\boldsymbol{\mathcal{F}}_{,qq}(\boldsymbol{q})\,\dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\boldsymbol{q}}^T \boldsymbol{\mathcal{F}}_{1,qq}(\boldsymbol{q})\,\dot{\boldsymbol{q}} \\ \vdots \\ \dot{\boldsymbol{q}}^T \boldsymbol{\mathcal{F}}_{n_x,qq}(\boldsymbol{q})\,\dot{\boldsymbol{q}} \end{bmatrix}, \tag{4.4a}$$

$$\boldsymbol{\mathcal{F}}_{i,qq}(\boldsymbol{q}) = \begin{bmatrix} \frac{\partial^2 \mathcal{F}_i(\boldsymbol{q})}{\partial q_1 \partial q_1} & \cdots & \frac{\partial^2 \mathcal{F}_i(\boldsymbol{q})}{\partial q_1 \partial q_{n_{dof}}} \\ \vdots & & \vdots \\ \frac{\partial^2 \mathcal{F}_i(\boldsymbol{q})}{\partial q_{n_{dof}} \partial q_1} & \cdots & \frac{\partial^2 \mathcal{F}_i(\boldsymbol{q})}{\partial q_{n_{dof}} \partial q_{n_{dof}}} \end{bmatrix}. \tag{4.4b}$$

The forces acting on each body in the system, expressed in terms of $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, and $\ddot{\boldsymbol{x}}$, can than be transformed by substituting (4.1) and (4.2). In doing so, expressions for the forces acting on each individual body are found in terms of the independent coordinates $\boldsymbol{q}$. These forces can then be reflected to the independent coordinates by applying,

$$\boldsymbol{F}_q = \boldsymbol{\mathcal{F}}_{,q}^T(\boldsymbol{q})\,\boldsymbol{F}_x, \tag{4.5}$$

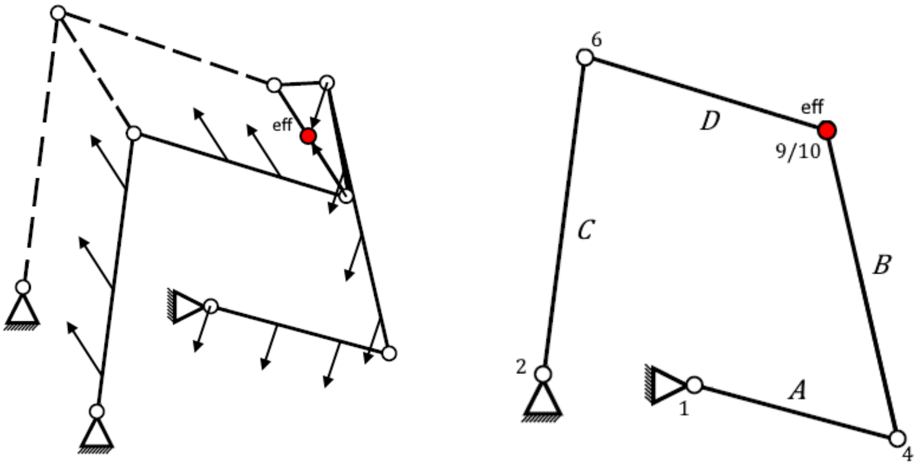
where $\boldsymbol{F}_x$ represents the forces acting on the generalized coordinates, and $\boldsymbol{F}_q$ represents the same forces reflected to the independent coordinates. Similar to a transmission ratio, $\boldsymbol{\mathcal{F}}_{,q}(\boldsymbol{q})$ defines the relation between the velocities $\dot{\boldsymbol{q}}$ and $\dot{\boldsymbol{x}}$ as well as the forces $\boldsymbol{F}_x$ and $\boldsymbol{F}_q$.

This method can be used to define the forces acting on a multi-body system relatively straightforwardly, and find the equations of motion based on these definitions by applying the transformations described above.

4.2.2 Simplified system representation

To simplify the modelling and identification process, the 2DOF manipulator as shown in figure 4.1 can be adjusted such that a simpler, dynamically equivalent system is found. This process was performed by Aarts [11]. The parallel arms are connected such that their rotational deflections are always equal. As a result, their dynamic parameters (masses, stiffness of hinges, etc.) can effectively be combined. Even more, their dynamic parameters cannot be distinguished in the identification process. This can be utilized to find a simplified representation of the system. The parallel arms can be combined into one (dynamically equivalent) single arm. To this end, beams E, G, and F, and the flexure hinges 3, 5, 7, and 11 are neglected. Instead, their dynamic behavior will be represented by the remaining parallel arm. Similarly, the hinge 8 is neglected. Since hinge 6 and 8 operate on the same rotation, these cannot be distinguished in the identification. Instead, these will combined into a single stiffness parameter.

Additionally, the positions of the remaining arms can be shifted such that these converge on the same point on the end-effector. This step does not change the kinematic relation between the arms, but does simplify the derivation of the kinematic relations. These simplification steps are illustrated in figure 4.2. The nominal positions of the system's hinges in the simplified representation are shown in table 4.2, where these have been expressed with respect to the nominal position of the end-effector.



(a) Removing parallel arm and shifting

(b) Result

Figure 4.2: Shifting the arms

Hinge	x-position (m)	y-position (m)
2	-0.2774	-0.2673
6	-0.2380	0.0765
10	0	0
1	-0.2004	-0.2565
4	0.0636	-0.2775
9	0	0

Table 4.2: Nominal position of hinges after shifting

The inertia properties of the system will be represented using the lumped mass representation [13]. In this idealization, concentrated masses are placed at the end nodes of each body. It can be applied on the condition that the total mass and inertia of the bodies remain the same, and the resulting center of mass coincides with the original. This is illustrated in figure 4.3. Note that an additional negative inertia is included on the right such that the total inertia is equivalent.

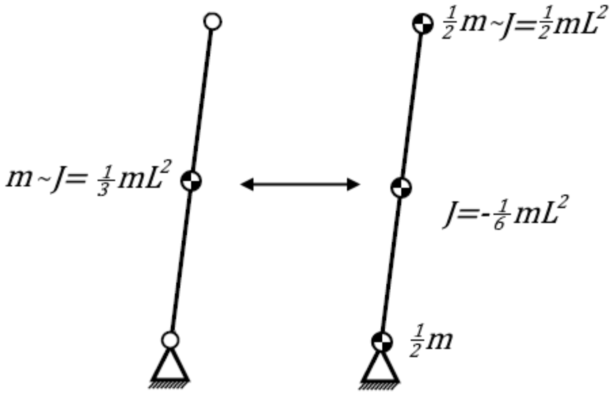

$m \sim J = \frac{1}{3}mL^2$
$\frac{1}{2}m \sim J = \frac{1}{2}mL^2$
$J = -\frac{1}{6}mL^2$
$\frac{1}{2}m$

Figure 4.3: Illustration of the lumped mass representation

Using the lumped mass representation, it is assumed that the body is rigid between its nodes. Furthermore, dynamical coupling between the translational and rotational motions are neglected. This leads to a purely diagonal inertia matrix. The representation may be less accurate as a result, but using this representation has computational advantages.

In the simplified representation of the 2DOF manipulator, the masses of the beams will be evenly split between its two nodes similar to figure 4.3. The centers of gravity will then be 'placed' in the correct location automatically in the identification process resulting from the identified values for m_A, m_B, m_C, m_D, and m_{eff}. The result of applying the lumped mass representation to the system is illustrated in figure 4.4. Each of the inertias in the lumped mass representation can be expressed in terms of the original mass parameters as,

$$\begin{aligned} m_1 &= \frac{m_A}{2}, & J_A &= -\frac{1}{6}m_A L_A^2, \\ m_2 &= m_C, & J_B &= -\frac{1}{6}m_B L_B^2, \\ m_4 &= \frac{m_A + m_B}{2}, & J_C &= -\frac{2}{6}m_C L_C^2, \\ m_6 &= m_C + m_D, & J_D &= -\frac{2}{6}m_D L_D^2. \\ m_{10} &= m_D + \frac{m_B}{2} + m_{eff}, \end{aligned} \tag{4.6}$$

Note that the masses m_C and m_D are not halved in (4.6), such that these represent the weight of only one the parallel beams.

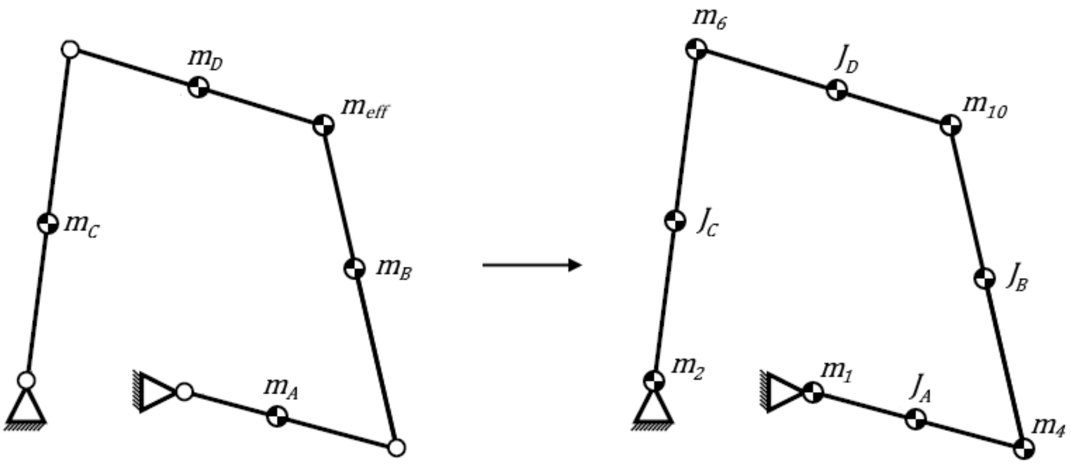


Figure 4.4: 2DOF manipulator in lumped mass representation

The flexure hinges will be represented in the model as ideal spring elements located at each of the hinges. Furthermore, ideal damper elements will be included in hinges 1 and 2 representing the friction induced by cables connected to the actuators and encoders. These are illustrated in figure 4.5, where k_i represents a compliance connected to hinge i, and d_i represents damping connected to hinge i. Note that the compliances connected to beams C and D are doubled, such that those k_i represent the stiffness of the flexure hinge in one of the parallel arms.

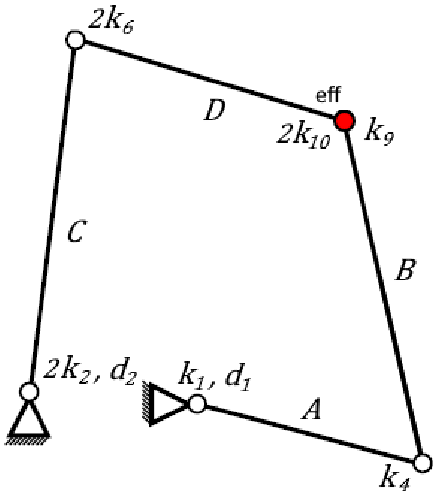


Figure 4.5: 2DOF manipulator with modelled springs and dampers

4.2.3 Kinematic analysis

To derive a dynamic model of the system, a kinematic analysis is required. This process was performed by Aarts [11]. The system is illustrated with relevant parameters in figure 4.6. In this figure, L_i represents the length of beam i, θ_i represents the angular deflection with respect to the x-axis of beam i, r_{enc} represents the distance at which the encoders are attached to the beams, and x_{enc} and y_{enc} represent the displacement of the encoders along that radius. Note that all angles are defined positive in counter-clockwise direction.

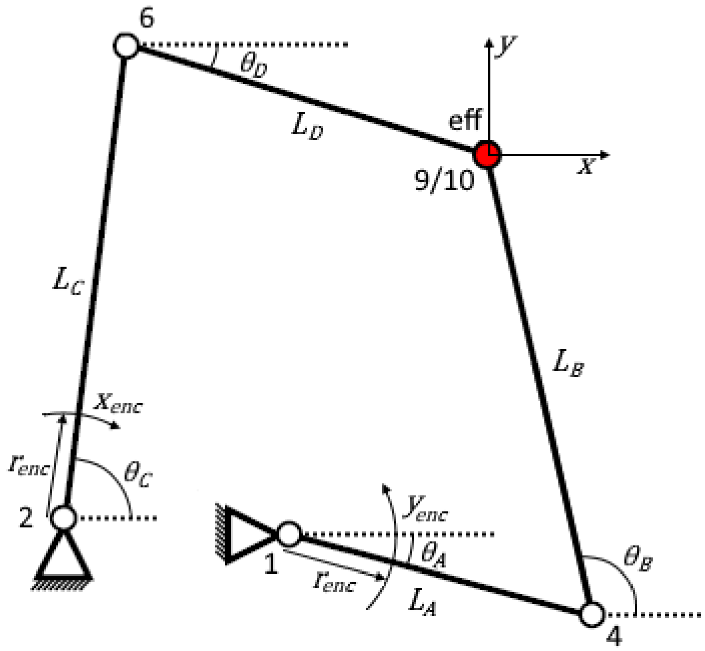


Figure 4.6: System with kinematic parameters

Parameter	Value (unit)
L_A	0.2648 (m)
L_B	0.2847 (m)
L_C	0.346 (m)
L_D	0.25 (m)
r_{enc}	0.251 (m)
r_{act}	0.235 (m)
$\theta_{A,nom}$	-0.0793 (rad)
$\theta_{B,nom}$	1.7959 (rad)
$\theta_{C,nom}$	1.4567 (rad)
$\theta_{D,nom}$	-0.3108 (rad)

Table 4.3: Parameters of the 2DOF manipulator

The generalized coordinates $\boldsymbol{x}$ of the system correspond to the displacements of each the (moving) inertias in the lumped mass representation:

$$\boldsymbol{x} = \begin{bmatrix}\theta_A & \theta_B & \theta_C & \theta_D & x_4 & y_4 & x_6 & y_6 & x_{10} & y_{10}\end{bmatrix}^T . \tag{4.7}$$

The independent coordinates $\boldsymbol{q}$ are chosen as the encoder displacements:

$$\boldsymbol{q} = \begin{bmatrix}x_{enc}\\ y_{enc}\end{bmatrix} . \tag{4.8}$$

Choosing the independent coordinates this way is convenient, as this will allow the use of the encoder readings without the need for a transformation later on.

Based on the nominal positions of the hinges in table 4.2, the length of the beams L_i and their nominal angles $\theta_{i,nom}$ can be calculated. The resulting values are shown in table 4.3. With these parameters, all kinematic relations required to define the geometric transfer function $\mathcal{F}(\boldsymbol{q})$ can be determined. To start, the angular deflections of beams x_1 and y_1 can be found as,

$$\begin{aligned}\theta_A &= \theta_{y1,nom} + \frac{y_{enc}}{r_{enc}},\\ \theta_C &= \theta_{x1,nom} - \frac{x_{enc}}{r_{enc}}.\end{aligned} \tag{4.9}$$

Expressions for the positions of hinges 4 and 6 can subsequently be found as,

$$\begin{aligned}x_4 &= x_1 + L_A \cos\left(\theta_A\right),\\ y_4 &= y_1 + L_A \sin\left(\theta_A\right),\\ x_6 &= x_2 + L_C \cos\left(\theta_C\right),\\ y_6 &= y_2 + L_C \sin\left(\theta_C\right),\end{aligned} \tag{4.10}$$

where x_1, x_2, y_1, and y_2 correspond to the constant position of these hinges shown in table 4.2.

Expressions for the angles θ_B and θ_D are found by introducing the triangle shown in figure 4.7. The length L_s and angle θ_s of the side of the triangle between nodes 4 and 6 can be found as,

$$\begin{aligned} L_s &= \sqrt{[x_4 - x_6]^2 + [y_4 - y_6]^2} \\ \theta_s &= \tan^{-1}\left(\frac{y_4 - y_6}{x_4 - x_6}\right) \end{aligned} \tag{4.11}$$

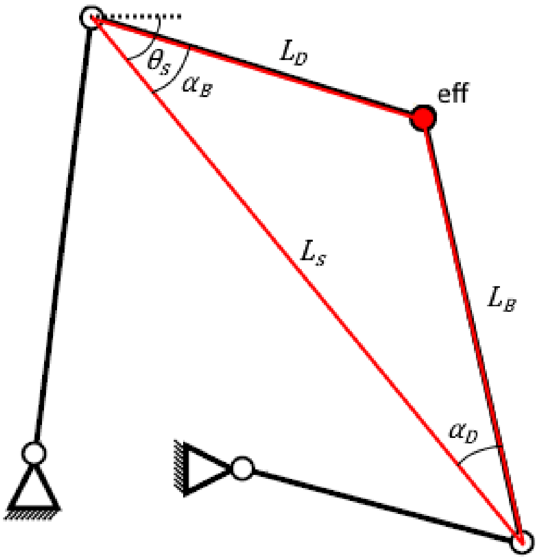


Figure 4.7: Triangle defined to find θ_B and θ_D

Using the cosine rule and (4.11), the inner angles of the triangle α_B and α_D can be found as,

$$\begin{aligned} \alpha_B &= \cos^{-1}\left(\frac{L_D^2 + L_s^2 - L_B^2}{2L_D L_s}\right), \\ \alpha_D &= \cos^{-1}\left(\frac{L_B^2 + L_s^2 - L_D^2}{2L_B L_s}\right). \end{aligned} \tag{4.12}$$

In turn, these can be used to find expressions for the angles θ_B and θ_D:

$$\begin{aligned} \theta_B &= \pi + \theta_s - \alpha_D, \\ \theta_D &= \theta_s + \alpha_B. \end{aligned} \tag{4.13}$$

Lastly, expressions for the positions of hinge 10 can be found by combining (4.10) and (4.13):

$$\begin{aligned} x_{10} &= x_6 + L_D \cos(\theta_D)\,, \\ y_{10} &= y_6 + L_D \sin(\theta_D)\,. \end{aligned} \tag{4.14}$$

Then, an expression for all generalized coordinates $\boldsymbol{x}$ is found as a function of the independent coordinates $\boldsymbol{q}$. Together, these relations represent the geometric transfer function $\boldsymbol{\mathcal{F}}(\boldsymbol{q})$, which is found by substituting the kinematic relations found into

$$\boldsymbol{x} = \boldsymbol{\mathcal{F}}(\boldsymbol{q}) = \begin{bmatrix} \theta_A & \theta_B & \theta_C & \theta_D & x_4 & y_4 & x_6 & y_6 & x_{10} & y_{10} \end{bmatrix}^T . \tag{4.15}$$

However, the generalized coordinates $\boldsymbol{x}$ only include the positions and orientations of the bodies. In order to include springs and dampers in the dynamic model, the deformation coordinates $\boldsymbol{\varepsilon}$ are defined as,

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_4 \\ \varepsilon_6 \\ \varepsilon_9 \\ \varepsilon_{10} \end{bmatrix}, \tag{4.16}$$

where ε_i represents the angular deflection of hinge i from its nominal angle.

Similar to before, a geometric transfer function $\boldsymbol{\mathcal{E}}(\boldsymbol{q})$ can be defined to express these deformation coordinates in terms of the independent coordinates. The deflection of each hinge can be found from (4.9) and (4.13) as:

$$\boldsymbol{\varepsilon} = \boldsymbol{\mathcal{E}}(\boldsymbol{q}) = \begin{bmatrix} \theta_A - \theta_{A,nom} \\ \theta_C - \theta_{C,nom} \\ \theta_B - \theta_{B,nom} - \theta_A + \theta_{A,nom} \\ \theta_D - \theta_{D,nom} - \theta_C + \theta_{C,nom} \\ -\theta_B + \theta_{B,nom} \\ -\theta_D + \theta_{D,nom} \end{bmatrix} . \tag{4.17}$$

Finally, the time derivatives of the geometric transfer functions $\boldsymbol{\mathcal{F}}(\boldsymbol{q})$ and $\boldsymbol{\mathcal{E}}(\boldsymbol{q})$ as defined in (4.2) should be found as well. Unfortunately the geometric transfer functions consist of complicated expressions, and their time derivatives are significantly more complicated. Finding expressions for the time derivatives analytically would be a time consuming and error prone process. Instead, expressions for the time derivatives are found using Matlab's symbolic toolbox. This is achieved by defining a symbolic variable in Matlab for each generalized coordinate and independent coordinate, and defining the geometric transfer functions as shown in this section. The time derivatives can then be calculated in Matlab as shown in (4.2) to (4.4) using the commands 'jacobian' and 'hessian'.

4.2.4 Equations of motion

The equations of motion of the simplified system described in section 4.2.2 will be found using the generalized coordinate approach as in section 4.2.1. This process was performed by Aarts [11]. First, the inertia, damping, and spring forces will be defined with respect to the individual bodies and hinge-points. These will then be transformed such that the forces reflected on the independent coordinates $\boldsymbol{q}$ are found. These can be combined to find the equations of motion as reflected on the independent coordinates. As a final step, the equations of motion are reflected to the end-effector position. This way, the external forces acting on the end-effector do not need to be transformed later on.

The inertia forces reflected on the individual bodies $\boldsymbol{F}_{m,x}$ can be expressed as

$$\boldsymbol{F}_{m,x} = \boldsymbol{M}\ddot{\boldsymbol{x}}, \tag{4.18}$$

with $\ddot{x}$ the acceleration of the generalized coordinates (4.7)

$$\ddot{\boldsymbol{x}} = \begin{bmatrix}\ddot{\theta}_A & \ddot{\theta}_B & \ddot{\theta}_C & \ddot{\theta}_D & \ddot{x}_4 & \ddot{y}_4 & \ddot{x}_6 & \ddot{y}_6 & \ddot{x}_{10} & \ddot{y}_{10}\end{bmatrix}^T, \tag{4.19}$$

and $\boldsymbol{M}$ the inertia matrix corresponding to these accelerations

$$\boldsymbol{M} = \text{diag}(J_A, J_B, J_C, J_D, m_4, m_4, m_6, m_6, m_{10}, m_{10})\,. \tag{4.20}$$

Substituting expressions for the accelerations $\ddot{x}$ given in (4.2) and reflecting the forces to the independent coordinates using (4.5) gives

$$\boldsymbol{F}_{m,q} = \boldsymbol{\mathcal{F}}_{,q}^T \boldsymbol{M} \boldsymbol{\mathcal{F}}_{,q} \ddot{\boldsymbol{q}} + \boldsymbol{\mathcal{F}}_{,q}^T \boldsymbol{M} \left(\boldsymbol{\mathcal{F}}_{,q} \dot{\boldsymbol{q}}\right) \dot{\boldsymbol{q}}, \tag{4.21}$$

with $\boldsymbol{F}_{m,q}$ the inertia forces of the bodies reflected on the independent coordinates $\boldsymbol{q}$.

The spring and damper forces can be expressed with respect to the individual hinge rotations ε. The spring forces reflected on the individual hinges $\boldsymbol{F}_{k,\varepsilon}$ can be expressed as

$$\boldsymbol{F}_{k,\varepsilon} = \boldsymbol{K}\boldsymbol{\varepsilon}, \tag{4.22}$$

with the hinge rotations ε and the stiffness matrix $\boldsymbol{K}$ defined as

$$\boldsymbol{\varepsilon} = \begin{bmatrix}\varepsilon_1 & \varepsilon_2 & \varepsilon_4 & \varepsilon_6 & \varepsilon_9 & \varepsilon_{10}\end{bmatrix}^T, \quad \boldsymbol{K} = \begin{bmatrix} k_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2k_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2k_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2k_{10} \end{bmatrix}. \tag{4.23}$$

Substituting the geometric transfer function $\boldsymbol{\mathcal{E}}(\boldsymbol{q})$ given in (4.17) and reflecting the forces to the independent coordinates using (4.5) gives

$$\boldsymbol{F}_{k,q} = \boldsymbol{\mathcal{E}}_{,q}^T \boldsymbol{K} \boldsymbol{\mathcal{E}}(\boldsymbol{q}), \tag{4.24}$$

with $\boldsymbol{F}_{k,q}$ the spring forces of the hinges reflected on the independent coordinates $\boldsymbol{q}$, and $\boldsymbol{\mathcal{E}}_{,q}$ the Jacobian of $\boldsymbol{\mathcal{E}}(\boldsymbol{q})$ as defined in (4.3).

The damping forces reflected on the individual hinges $\boldsymbol{F}_{d,\varepsilon}$ can be expressed as

$$\boldsymbol{F}_{d,\varepsilon} = \boldsymbol{D}\dot{\boldsymbol{\varepsilon}}, \tag{4.25}$$

with $\dot{\varepsilon}$ the velocities of the deflection coordinates (4.16)

$$\dot{\boldsymbol{\varepsilon}} = \begin{bmatrix}\dot{\varepsilon}_1 & \dot{\varepsilon}_2 & \dot{\varepsilon}_4 & \dot{\varepsilon}_6 & \dot{\varepsilon}_9 & \dot{\varepsilon}_{10}\end{bmatrix}^T, \tag{4.26}$$

and $\boldsymbol{D}$ the damping matrix corresponding to these velocities

$$\boldsymbol{D} = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.27}$$

Substituting (4.2) and reflecting the forces to the independent coordinates using (4.5) gives

$$\boldsymbol{F}_{d,q} = \boldsymbol{\mathcal{E}}_{,q}^T \boldsymbol{D} \boldsymbol{\mathcal{E}}_{,q} \dot{\boldsymbol{q}}. \tag{4.28}$$

The equations of motion expressed from the perspective of the independent coordinates $\boldsymbol{q}$ can then be found by taking the sum of all forces:

$$\begin{aligned} \boldsymbol{F}_{in,q} + \boldsymbol{F}_{ext,q} &= \boldsymbol{F}_{m,q} + \boldsymbol{F}_{d,q} + \boldsymbol{F}_{k,q} \\ &= \boldsymbol{\mathcal{F}}_{,q}^T \boldsymbol{M} \boldsymbol{\mathcal{F}}_{,q} \ddot{\boldsymbol{q}} + \boldsymbol{\mathcal{F}}_{,q}^T \boldsymbol{M} \left(\boldsymbol{\mathcal{F}}_{,q} \dot{\boldsymbol{q}} \right) \dot{\boldsymbol{q}} + \boldsymbol{\mathcal{E}}_{,q}^T \boldsymbol{D} \boldsymbol{\mathcal{E}}_{,q} \dot{\boldsymbol{q}} + \boldsymbol{\mathcal{E}}_{,q}^T \boldsymbol{K} \boldsymbol{\mathcal{E}}, \end{aligned} \tag{4.29}$$

with $\boldsymbol{F}_{in,q}$ the actuator forces reflected on the independent coordinates, and $\boldsymbol{F}_{ext,q}$ the external forces reflected on the independent coordinates.

The actuator forces in (4.29) are those reflected on the independent coordinates, while these will be available as reflected on the actuator positions. So, these need to be transformed before they can be used. This is a simple transformation, since the independent coordinates (encoder positions) and actuator positions are connected to the same angular deflection at different radii. The actuator torques can be reflected to the independent coordinates by applying

$$\boldsymbol{F}_{in,q} = \frac{r_{act}}{r_{enc}} \boldsymbol{F}_{in,act}. \tag{4.30}$$

with $\boldsymbol{F}_{in,act}$ the actuator forces reflected on the actuator positions, and $\boldsymbol{F}_{in,q}$ the same actuator forces reflected on the independent coordinates.

To use the equations of motion (4.29) found by Aarts [11], they should be adjusted. The external forces incorporated in the equation of motion represent those reflected on the independent coordinates. However, the external forces reflected on the end-effector position are desired to be estimated in the end. Therefore, the equations of motion should be transformed such that they are expressed with respect to the end-effector position. This can be achieved by transforming all forces using (4.5). For this transformation, the Jacobian expressing the end-effector velocities in terms of the velocities of the independent coordinates is required. A relation between these velocities is already incorporated in the Jacobian of (4.15):

$$\dot{\boldsymbol{x}} = \boldsymbol{\mathcal{F}}_{,q}(\boldsymbol{q})\, \dot{\boldsymbol{q}}, \tag{4.31}$$

for which an expression was determined using Matlab's symbolic toolbox. The submatrix of $\boldsymbol{\mathcal{F}}_{,q}(\boldsymbol{q})$ relevant to this transformation is defined as

$$\dot{\boldsymbol{x}}_{eff} = \boldsymbol{\mathcal{F}}_{eff,q}(\boldsymbol{q})\, \dot{\boldsymbol{q}}, \tag{4.32}$$

with $\dot{\boldsymbol{x}}_{eff}$ the end-effector velocities

$$\dot{\boldsymbol{x}}_{eff} = \begin{bmatrix} \dot{x}_{10} \\ \dot{y}_{10} \end{bmatrix}. \tag{4.33}$$

The Jacobian required for the transformation can be found from (4.32) by rewriting as

$$\dot{\boldsymbol{q}} = \boldsymbol{\mathcal{F}}^{-1}_{eff,q}(\boldsymbol{q})\, \dot{\boldsymbol{x}}_{eff}. \tag{4.34}$$

The forces in the equations of motion can then be reflected to the end-effector position by applying $\boldsymbol{\mathcal{F}}^{-1}_{eff,q}(\boldsymbol{q})$ to (4.5) as

$$\boldsymbol{F}_{eff} = \boldsymbol{\mathcal{F}}^{-T}_{eff,q}(\boldsymbol{q})\boldsymbol{F}_q, \tag{4.35}$$

with $\boldsymbol{F}_q$ a force reflected on the independent coordinates, and $\boldsymbol{F}_{eff}$ that same force reflected on the end-effector position.

Applying this transformation to the equation of motion (4.29) gives

$$\boldsymbol{F}_{ext,eff} = \boldsymbol{\mathcal{F}}^{-T}_{eff,q} \left[\boldsymbol{\mathcal{F}}^{T}_{,q} \boldsymbol{M} \boldsymbol{\mathcal{F}}_{,q} \ddot{\boldsymbol{q}} + \boldsymbol{\mathcal{F}}^{T}_{,q} \boldsymbol{M} \left(\boldsymbol{\mathcal{F}}_{,q} \dot{\boldsymbol{q}} \right) \dot{\boldsymbol{q}} + \boldsymbol{\mathcal{E}}^{T}_{,q} \boldsymbol{D} \boldsymbol{\mathcal{E}}_{,q} \dot{\boldsymbol{q}} + \boldsymbol{\mathcal{E}}^{T}_{,q} \boldsymbol{K} \boldsymbol{\mathcal{E}} - \frac{r_{act}}{r_{enc}} \boldsymbol{F}_{in,act} \right] \tag{4.36}$$

where $\boldsymbol{F}_{ext,eff}$ represents the external forces reflected on the end-effector position.

During simulations, it was found that calculating the inverse $\boldsymbol{\mathcal{F}}^{-T}_{eff,q}$ was difficult for Matlab's symbolic toolbox, requiring significant time to solve. An alternative approach to calculate this transformation matrix is given in appendix B, which circumvents the need to calculate a matrix inverse.

4.2.5 Identification

The 2DOF manipulator system parameters were identified by Aarts [11]. This was achieved by applying linear least squares regression based on the model derived in previous sections and data collected by Heerze [14]. The values found for the parameters are listed in table 4.4.

Parameter	Identified value (unit)
m_A	0.497547523949 (kg)
m_B	0.581336378956 (kg)
m_C	0.144416464957 (kg)
m_D	0.572295408988 (kg)
m_{eff}	-0.503277438841 (kg)
k_1	3.963719549131 $({}^{Nm}/_{rad})$
k_2	3.492090967745 $({}^{Nm}/_{rad})$
k_4	1.278892975362 $({}^{Nm}/_{rad})$
k_6	1.506895042708 $({}^{Nm}/_{rad})$
k_9	-1.909866894848 $({}^{Nm}/_{rad})$
k_{10}	-0.824908122655 $({}^{Nm}/_{rad})$
d_1	0.130875094160 $({}^{Nm\cdot s}/_{rad})$
d_2	0.072239449199 $({}^{Nm\cdot s}/_{rad})$

Table 4.4: Identified parameters of 2DOF manipulator

4.2.6 Linearization

To be able to write the system in state-space form, the equation of motion (4.36) needs to be rewritten such that it becomes an expression for $\ddot{\boldsymbol{q}}$. This would require the inverse of the matrix $\boldsymbol{\mathcal{F}}_{eff,q}^{-T}\boldsymbol{\mathcal{F}}_{,q}^{T}\boldsymbol{M}\boldsymbol{\mathcal{F}}_{,q}$ to be calculated. Unfortunately, Matlab's symbolic toolbox has difficulties calculating this inverse, which is caused by this matrix being such a large and complicated expression.

However, the equation of motion will also need to be linearized in order to apply it to the LKF. To prevent Matlab from having to compute the inverse of the mentioned matrix in symbolic form, the equation of motion can first be linearized, and rewritten such that $\ddot{q}$ is on the left hand side of the equation thereafter. Thereby, Matlab would only need to calculate the inverse of a non-symbolic 2×2 matrix.

The equations of motion will be linearized by applying first-order Taylor expansion as discussed in section 2.3. For readability, the equation of motion (4.36) will be denoted in this section as

$$\boldsymbol{F}_{ext,eff} = \boldsymbol{F}\left(\boldsymbol{Q}\right), \tag{4.37}$$

with $\boldsymbol{Q}$ defined as

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}} \\ \boldsymbol{F}_{in,act} \end{bmatrix}. \tag{4.38}$$

Applying first-order Taylor approximation at an operating point $\boldsymbol{Q}_0$ gives

$$\boldsymbol{F}_{lin}\left(\boldsymbol{Q}\right) = \boldsymbol{F}|_{\boldsymbol{Q}=\boldsymbol{Q}_0} + \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}}\right|_{\boldsymbol{Q}=\boldsymbol{Q}_0}\left(\boldsymbol{Q} - \boldsymbol{Q}_0\right). \tag{4.39}$$

Although the equation of motion $\boldsymbol{F}\left(\boldsymbol{Q}\right)$ consists of complicated symbolic expressions, Matlab's symbolic toolbox is capable of calculating the partial derivatives in (4.39) without problems.

Next, the linearized equation of motion is rewritten such that it can be applied in a state-space model. (4.39) can be rewritten to group the constant terms as

$$\boldsymbol{F}_{lin}\left(\boldsymbol{Q}\right) = \left[\boldsymbol{F}|_{\boldsymbol{Q}=\boldsymbol{Q}_0} - \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}}\right|_{\boldsymbol{Q}=\boldsymbol{Q}_0}\boldsymbol{Q}_0\right] + \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}}\right|_{\boldsymbol{Q}=\boldsymbol{Q}_0}\boldsymbol{Q}. \tag{4.40}$$

Substituting (4.38) back gives for the non-linear term

$$\left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}}\right|_{\boldsymbol{Q}=\boldsymbol{Q}_0}\boldsymbol{Q} = \left.\frac{\partial \boldsymbol{F}}{\partial \ddot{\boldsymbol{q}}}\right|_{\boldsymbol{Q}=\boldsymbol{Q}_0}\ddot{\boldsymbol{q}} + \left.\frac{\partial \boldsymbol{F}}{\partial \dot{\boldsymbol{q}}}\right|_{\boldsymbol{Q}=\boldsymbol{Q}_0}\dot{\boldsymbol{q}} + \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{q}}\right|_{\boldsymbol{Q}=\boldsymbol{Q}_0}\boldsymbol{q} + \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{F}_{in,act}}\right|_{\boldsymbol{Q}=\boldsymbol{Q}_0}\boldsymbol{F}_{in,act}. \tag{4.41}$$

Given their relation to the accelerations $\ddot{\boldsymbol{q}}$, velocities $\dot{\boldsymbol{q}}$, and displacements $\boldsymbol{q}$ of the system, these partial derivatives can be interpreted as the linearized inertia, damping, and stifness matrices:

$$\begin{aligned} \boldsymbol{M}_{lin} &= \left.\frac{\partial \boldsymbol{F}}{\partial \ddot{\boldsymbol{q}}}\right|_{\boldsymbol{Q}=\boldsymbol{Q}_0}, \\ \boldsymbol{D}_{lin} &= \left.\frac{\partial \boldsymbol{F}}{\partial \dot{\boldsymbol{q}}}\right|_{\boldsymbol{Q}=\boldsymbol{Q}_0}, \\ \boldsymbol{K}_{lin} &= \left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{q}}\right|_{\boldsymbol{Q}=\boldsymbol{Q}_0}. \end{aligned} \tag{4.42}$$

Furthermore, the remaining terms of the linearized equation of motion (4.40) are defined as

$$\begin{aligned} \boldsymbol{F}_0 &= \left[\boldsymbol{F}|_{\boldsymbol{Q}=\boldsymbol{Q}_0} - \left. \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}} \right|_{\boldsymbol{Q}=\boldsymbol{Q}_0} \boldsymbol{Q}_0 \right], \\ \boldsymbol{U}_{lin} &= \left. \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{F}_{in,act}} \right|_{\boldsymbol{Q}=\boldsymbol{Q}_0}. \end{aligned} \tag{4.43}$$

Substituting the definitions (4.42) and (4.43) into (4.40) gives the linearized equation of motion

$$\boldsymbol{F}_{ext,eff}\left(\ddot{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \boldsymbol{q}\right) = \boldsymbol{F}_0 + \boldsymbol{M}_{lin}\ddot{\boldsymbol{q}} + \boldsymbol{D}_{lin}\dot{\boldsymbol{q}} + \boldsymbol{K}_{lin}\boldsymbol{q} + \boldsymbol{U}_{lin}\boldsymbol{F}_{in,act}. \tag{4.44}$$

Finally, the linearized equation of motion can be rewritten such that $\ddot{q}$ is on the left hand side:

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}_{lin}^{-1}\boldsymbol{F}_{ext,eff} - \boldsymbol{M}_{lin}^{-1}\boldsymbol{F}_0 - \boldsymbol{M}_{lin}^{-1}\boldsymbol{D}_{lin}\dot{\boldsymbol{q}} - \boldsymbol{M}_{lin}^{-1}\boldsymbol{K}_{lin}\boldsymbol{q} - \boldsymbol{M}_{lin}^{-1}\boldsymbol{U}_{lin}\boldsymbol{F}_{in,act}. \tag{4.45}$$

Since $\boldsymbol{M}_{lin}$ was linearized, Matlab can easily calculate its inverse.

The operating point $\boldsymbol{Q}_0$ at which the 2DOF manipulator will be linearized is set as

$$\boldsymbol{Q}_0 = \boldsymbol{0}. \tag{4.46}$$

This represents the 2DOF manipulator when it is in its nominal position, with zero velocity, acceleration, and input force. This is the optimal choice, since the expected range of each of these variables is centered around zero. This has the additional benefit that the constant term $\boldsymbol{F}_0$ becomes zero, which simplifies representing the equation of motion as a state-space system. The final linearized equation of motion then becomes

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}_{lin}^{-1}\boldsymbol{F}_{ext,eff} - \boldsymbol{M}_{lin}^{-1}\boldsymbol{D}_{lin}\dot{\boldsymbol{q}} - \boldsymbol{M}_{lin}^{-1}\boldsymbol{K}_{lin}\boldsymbol{q} - \boldsymbol{M}_{lin}^{-1}\boldsymbol{U}_{lin}\boldsymbol{F}_{in,act}. \tag{4.47}$$

4.2.7 Discrete-time augmented state-space model

Similar to section 3.2.2, the linearized equation of motion (4.47) can be rewritten to a state-space model

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x}. \end{aligned} \tag{4.48}$$

The output vector will consist of the encoder displacements $\boldsymbol{q}$, and the input vector will consist of the actuator forces $\boldsymbol{F}_{in,act}$. Furthermore, since the LKF will be used to find an estimate the state x of the system, the state vector should be augmented to include the external forces. The resulting state-, output-, and input-vector of the augmented state-space model become

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \\ \boldsymbol{F}_{ext,eff} \end{bmatrix} \quad , \quad \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}} \\ \dot{\boldsymbol{F}}_{ext,eff} \end{bmatrix} \quad , \quad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{q} \end{bmatrix} \quad , \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{F}_{in,act} \end{bmatrix}. \tag{4.49}$$

Based on the linearized equation of motion, the augmented state-space matrices can then be written as

$$
\begin{aligned}
\boldsymbol{A} &= \begin{bmatrix} \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ -\boldsymbol{M}_{lin}^{-1}\boldsymbol{K}_{lin} & -\boldsymbol{M}_{lin}^{-1}\boldsymbol{D}_{lin} & \boldsymbol{M}_{lin}^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \\
\boldsymbol{B} &= \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{M}_{lin}^{-1}\boldsymbol{U}_{lin} \\ \mathbf{0} \end{bmatrix}, \\
\boldsymbol{C} &= \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \mathbf{0} \end{bmatrix}.
\end{aligned} \tag{4.50}
$$

In order to utilize the state-space matrices in the LKF, they should be converted to discrete time. This can be achieved by applying (3.28), which gives:

$$
\begin{aligned}
\boldsymbol{A}_d &= \begin{bmatrix} \boldsymbol{I} & T_s \cdot \boldsymbol{I} & \mathbf{0} \\ -T_s \cdot \boldsymbol{M}_{lin}^{-1}\boldsymbol{K}_{lin} & \boldsymbol{I} - T_s \cdot \boldsymbol{M}_{lin}^{-1}\boldsymbol{D}_{lin} & T_s \cdot \boldsymbol{M}_{lin}^{-1} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} \end{bmatrix}, \\
\boldsymbol{B}_d &= \begin{bmatrix} \mathbf{0} \\ -T_s \cdot \boldsymbol{M}_{lin}^{-1}\boldsymbol{U}_{lin} \\ \mathbf{0} \end{bmatrix}, \\
\boldsymbol{C}_d &= \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \mathbf{0} \end{bmatrix}.
\end{aligned} \tag{4.51}
$$

These state-space matrices can be applied to the LKF. The methods used to derive the state-space model in this chapter are validated using the double pendulum model in appendix C.

4.3 Setting LKF parameters

With a model of the 2DOF manipulator available, the LKF can be set up for use. The methods used to set the LKF parameters for the double pendulum in section 3.3 will be applied to the 2DOF manipulator where possible as well. This process will be discussed next.

The noise covariance matrix $\boldsymbol{R}_d$ will include the measurement noise and quantization noise similarly to section 3.3. The encoders have a resolution of $5 \cdot 10^{-8} m$. Then, the variance of the quantization noise $\sigma^2_{quant,vi}$ can be approximated as

$$
\sigma^2_{quant,vi} = \frac{\left[5 \cdot 10^{-8}\right]^2}{12}. \tag{4.52}
$$

The variance of the measurement noise can be found by analyzing data measured while the system is completely at rest. The variance of the measurement noise $\sigma^2_{rest,vi}$ of each encoder is found based on the data shown in figure 4.8. The measurement noise covariance matrix $\boldsymbol{R}_d$ then becomes

$$
\begin{aligned}
\boldsymbol{R}_d &= \begin{bmatrix} \sigma_{quant,v1} + \sigma^2_{meas,v1} & 0 \\ 0 & \sigma_{quant,v2} + \sigma^2_{meas,v2} \end{bmatrix} \\
&= 10^{-10} \cdot \begin{bmatrix} 0.0162 & 0 \\ 0 & 0.2621 \end{bmatrix}.
\end{aligned} \tag{4.53}
$$

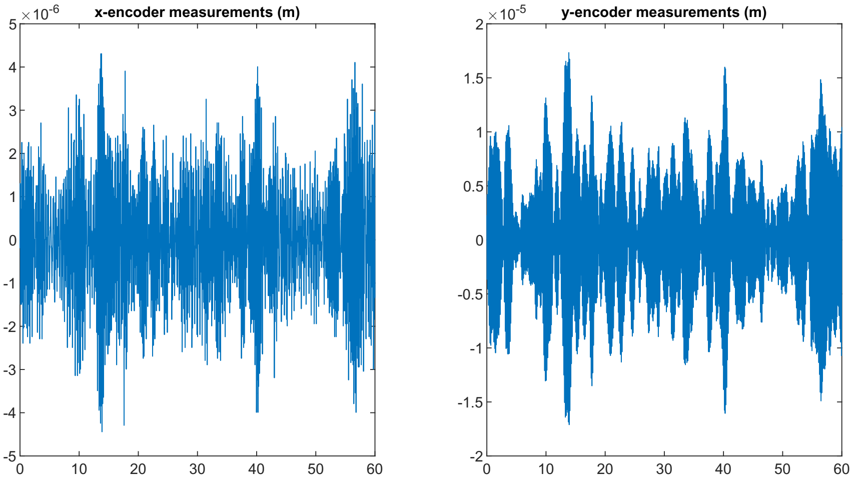


Figure 4.8: Measurement noise

The process covariance $\boldsymbol{Q}_{d,par}$ caused by errors in parameters will be found similarly to section 3.3 as well. It will be approximated by running multiple simulations with the system shown in figure 3.2. The 'real' system will be the state-space model represented by the state matrices in (4.51). The 'wrong' system will be generated in each simulation by randomizing the parameters in table 4.4 within a range of 10%, and applying these parameters to the state matrices in (4.51) as well.

Similarly, the state- and input-vectors will be generated from random values within a range. The range of each state and input is found by investigating the data gathered from experiments. For all data sets, the maximum value of each of the states and inputs is noted. The largest values of all data sets will be used as the ranges. Since there are no velocity sensors, the maximum values of the velocities will be approximated from the encoder data as

$$\begin{aligned} v_x &\approx \frac{x_{k+1} - x_k}{T_s} \\ v_y &\approx \frac{y_{k+1} - y_k}{T_s} \end{aligned} \tag{4.54}$$

Using (4.54) may give a distorted approximation of the velocities due to the presence of measurement and quantization noise, however. To reduce these effects, the x_k and y_k in (4.54) are replaced by the mean of a small range ($0.01s$) of the encoder data.

State/input	$+-$**Range**
x_1	0.05
x_2	0.05
x_3	0.12
x_4	0.12
x_5	13
x_6	13
u_1	10
u_2	10

Table 4.5: Assumed range of the states and inputs

The ranges used for each of the states and inputs are given in table 4.5. The resulting values for $\boldsymbol{Q}_{d,par}$ after running 100 simulations as described above become

$$\boldsymbol{Q}_{d,par} = 10^{-7} \cdot \begin{bmatrix} 0.0414 & 0 \\ 0 & 0.1998 \end{bmatrix}. \tag{4.55}$$

The process covariance $\boldsymbol{Q}_{d,F}$ caused by change in external torques is found in the same way as in section 3.3, based on data measured with the force sensors in experiments. To ensure that the LKF will be able to respond quickly enough to impulses, data will be used where the force measured changes quickly. The data that will be used is shown in figure 4.9.

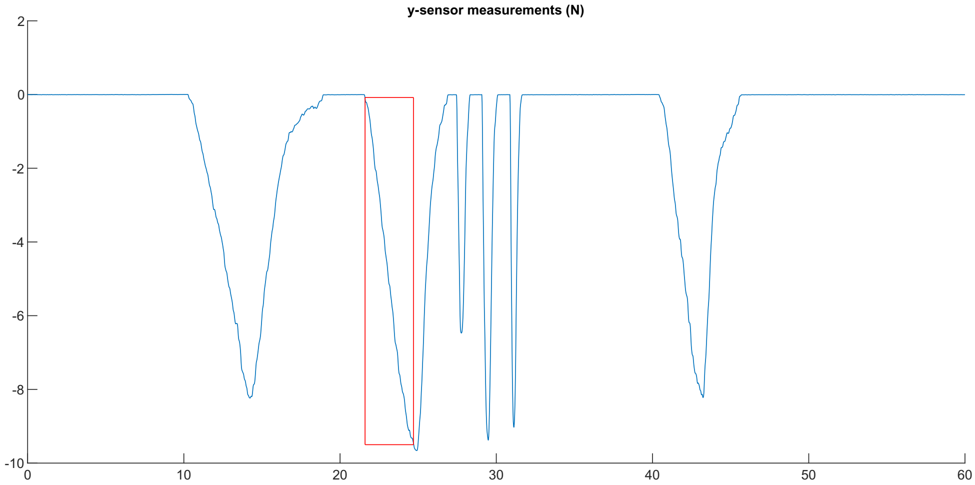


Figure 4.9: Force sensor data range used to find $\boldsymbol{Q}_F$

The resulting covariance matrix $\boldsymbol{Q}_{d,F}$ becomes

$$\boldsymbol{Q}_{d,F} = 10^{-5} \cdot \begin{bmatrix} 2.23 & 0 \\ 0 & 2.23 \end{bmatrix}. \tag{4.56}$$

In section 3.5, it was found that the elements of $\boldsymbol{Q}_{d,F}$ should be significantly greater than those of $\boldsymbol{Q}_{d,par}$. Conveniently, this is already the case.

Finally, the initial estimated states $\boldsymbol{x}_0$ and the initial estimate covariance matrix $\boldsymbol{P}_0$ need to be set. The initial estimated states are simply set to the nominal position of the 2DOF manipulator, with external forces of 0. The initial estimate's covariance matrix $\boldsymbol{P}_0$ is arbitrarily set very large, to indicate the uncertainty of the initial estimate. The exact values applied are

$$\boldsymbol{x}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \boldsymbol{P}_0 = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}. \tag{4.57}$$

4.4 External force estimation results

In this section, the performance of the LKF is tested based on data collected from multiple experiments. In these experiments, the end-effector was excited either by the actuators or by pushing the force sensors against it by hand. The external forces estimated by the LKF were then compared to the forces measured by the force sensors in the setup. The results of each experiment are shown in one figure, which show the displacements measured by the encoders, the forces exerted by the actuators, and the estimated force compared to the measured force in each direction.

4.4.1 Results: end-effector held in place by actuator

To start, the end-effector was held in place by the actuators in different positions, while the force sensors were pushed against it by hand. In doing so, the dynamic behavior of the 2DOF manipulator plays almost no role in the estimated external forces. This data can be used to test the performance of the LKF based on the actuator forces alone. The results are shown in figures 4.10 to 4.18.

The estimated external forces in y-direction seem to be estimated very accurately for all data sets. Unfortunately, the estimated x-forces seem to be coupled to the estimated y-forces. While the estimated x-forces should be zero at all times, they have a similar shape as the estimated y-forces. This may indicate that the actuator forces are not included correctly in the model. This can cause the LKF perceive an external force that is in reality not acting on the system. In each of the results, the erroneously estimated x-forces are about 10% of the magnitude of the y-forces. Alternatively, this could be explained by friction between the end-effector and the surface of the y-force sensor. This force would not be observed by the x-force sensor, but would still influence the estimated external forces. Despite that only a force in the y-direction was applied, a friction force may still have been induced because the displacements x_{eff} and y_{eff} of the system are coupled.

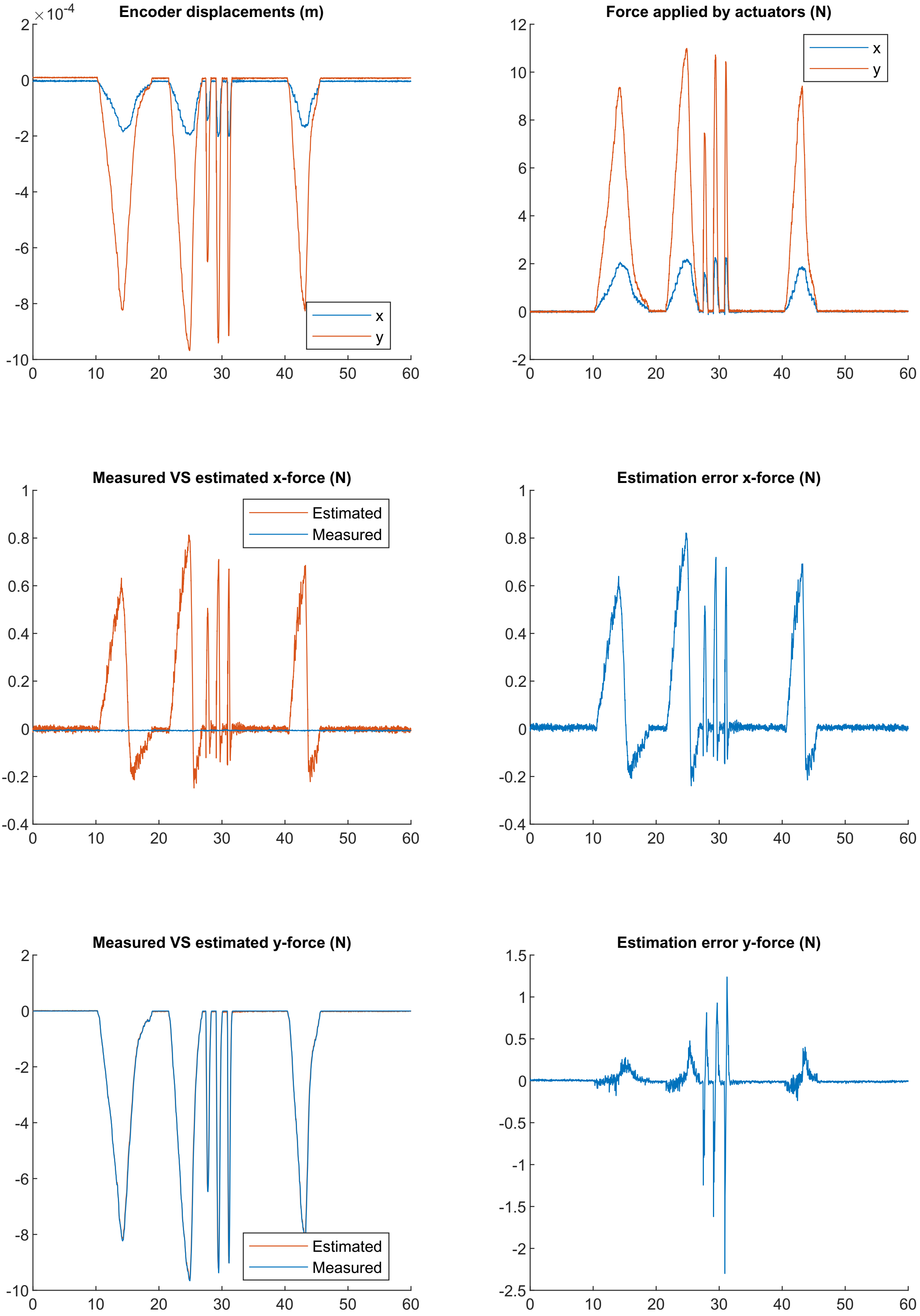


Figure 4.10: End-effector held in place in nominal position

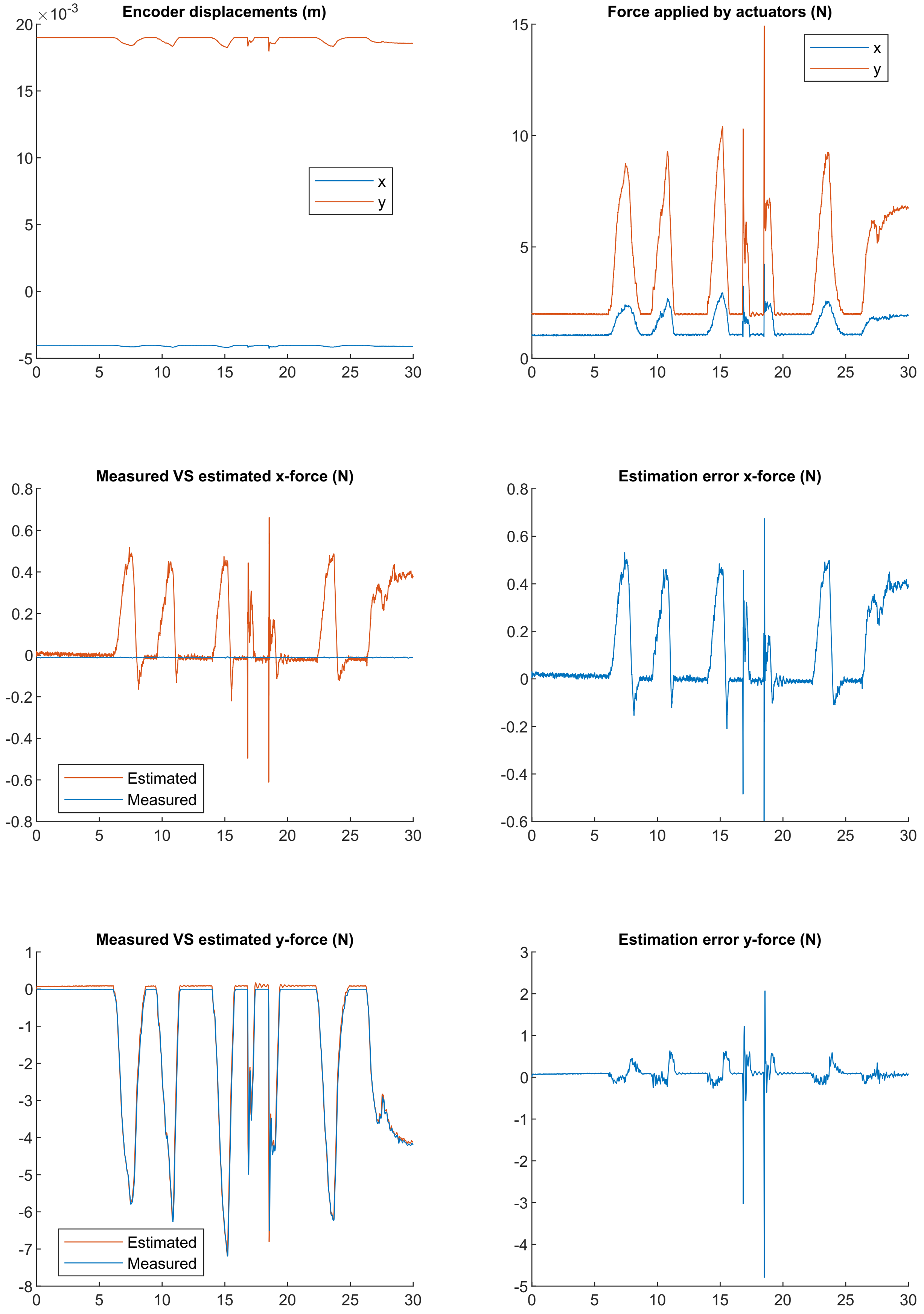


Figure 4.11: End-effector held in place at $x_{eff} = 0mm$, $y_{eff} = 20mm$

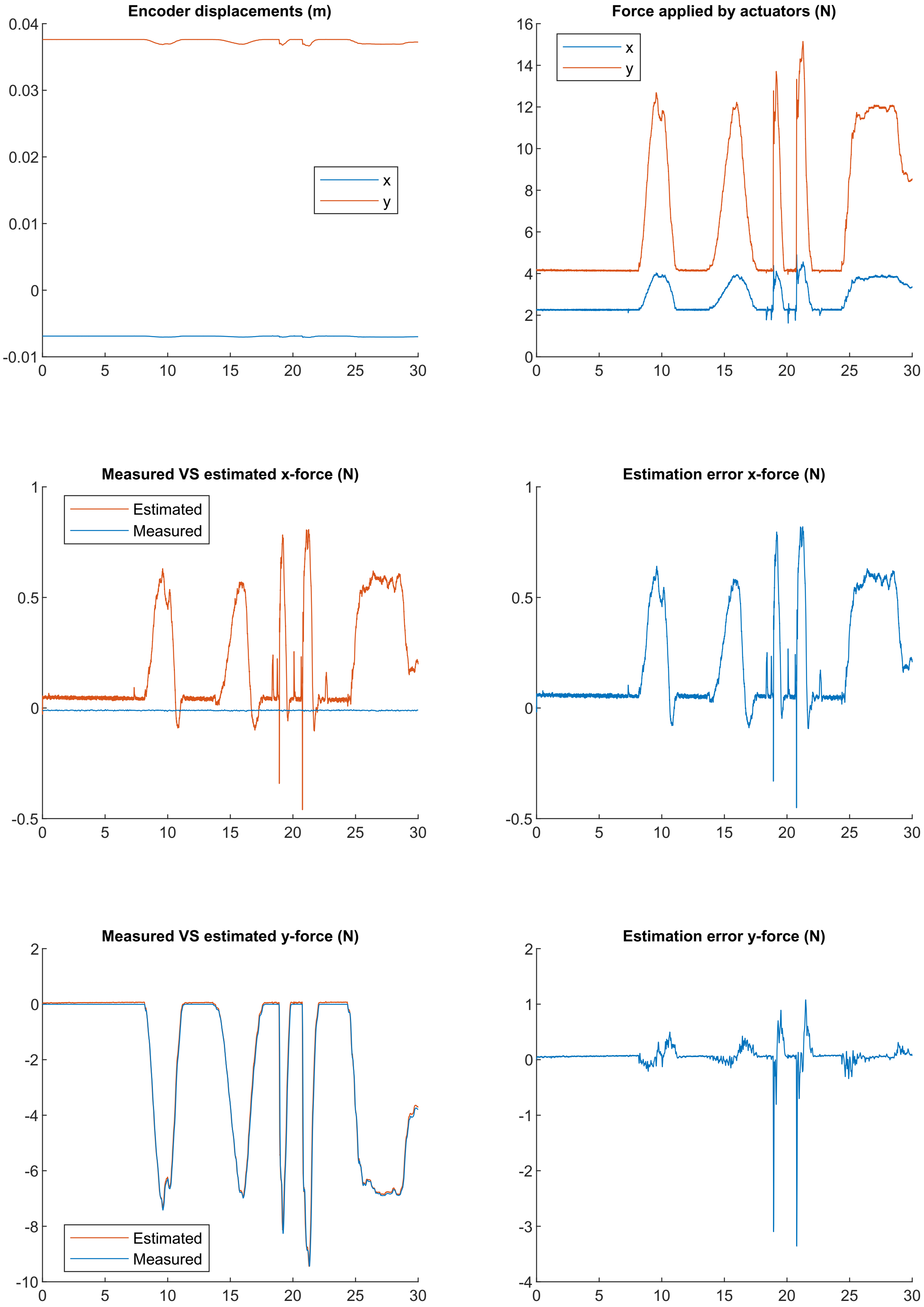


Figure 4.12: End-effector held in place at $x_{eff} = 0mm$, $y_{eff} = 40mm$

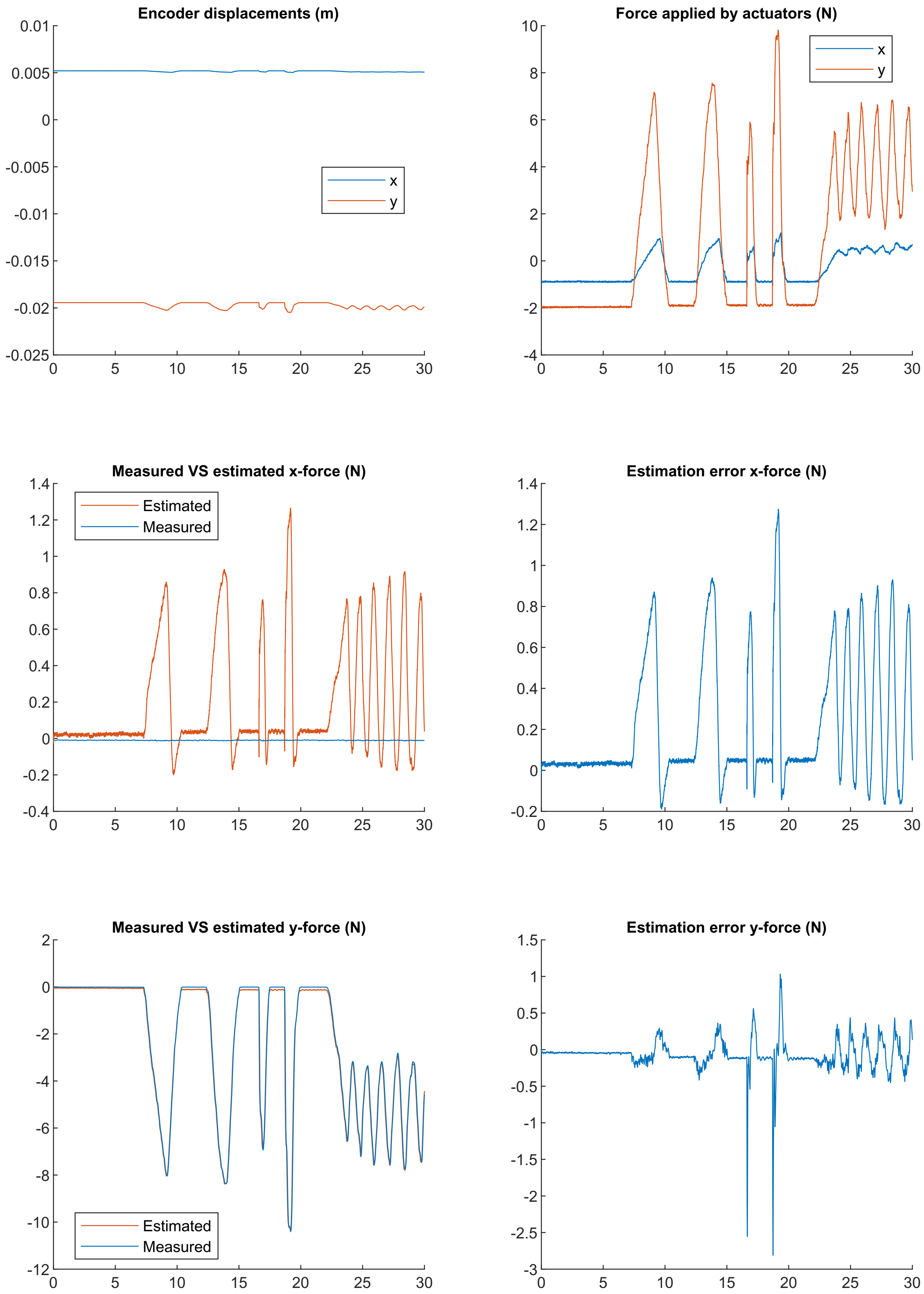


Figure 4.13: End-effector held in place at $x_{eff} = 0mm$, $y_{eff} = -20mm$

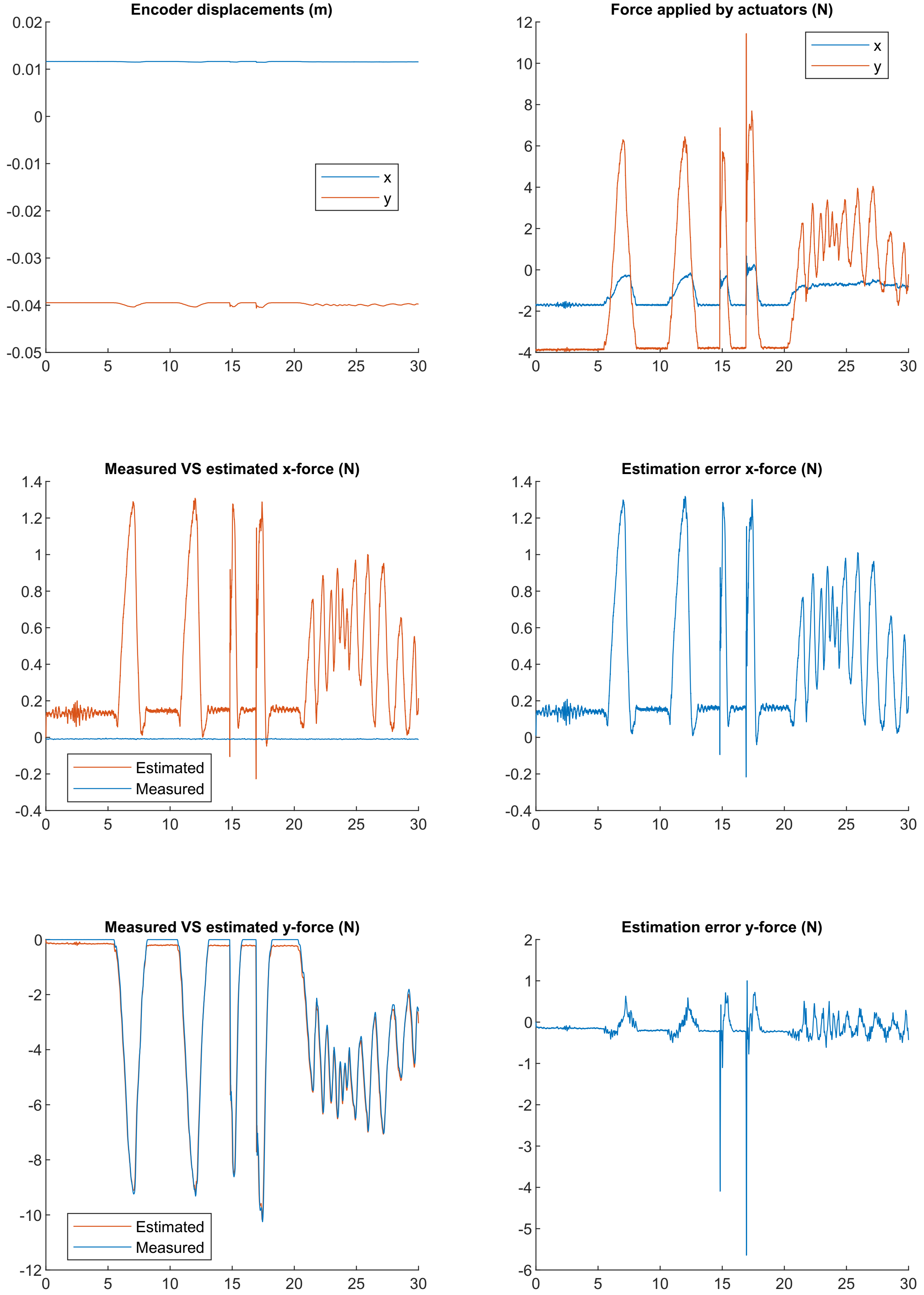


Figure 4.14: End-effector held in place at $x_{eff} = 0mm$, $y_{eff} = -40mm$

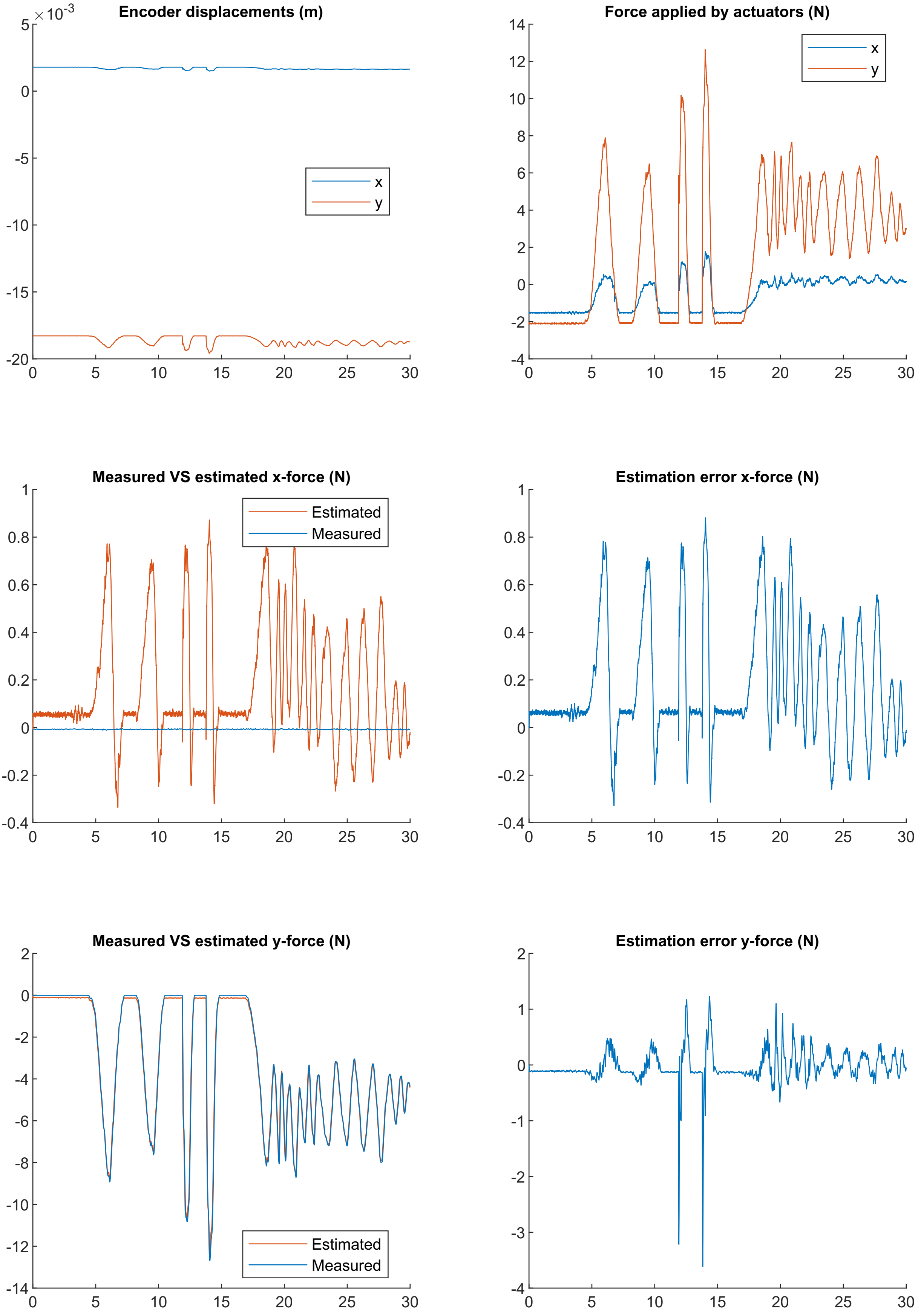


Figure 4.15: End-effector held in place at $x_{eff} = -5mm$, $y_{eff} = -20mm$

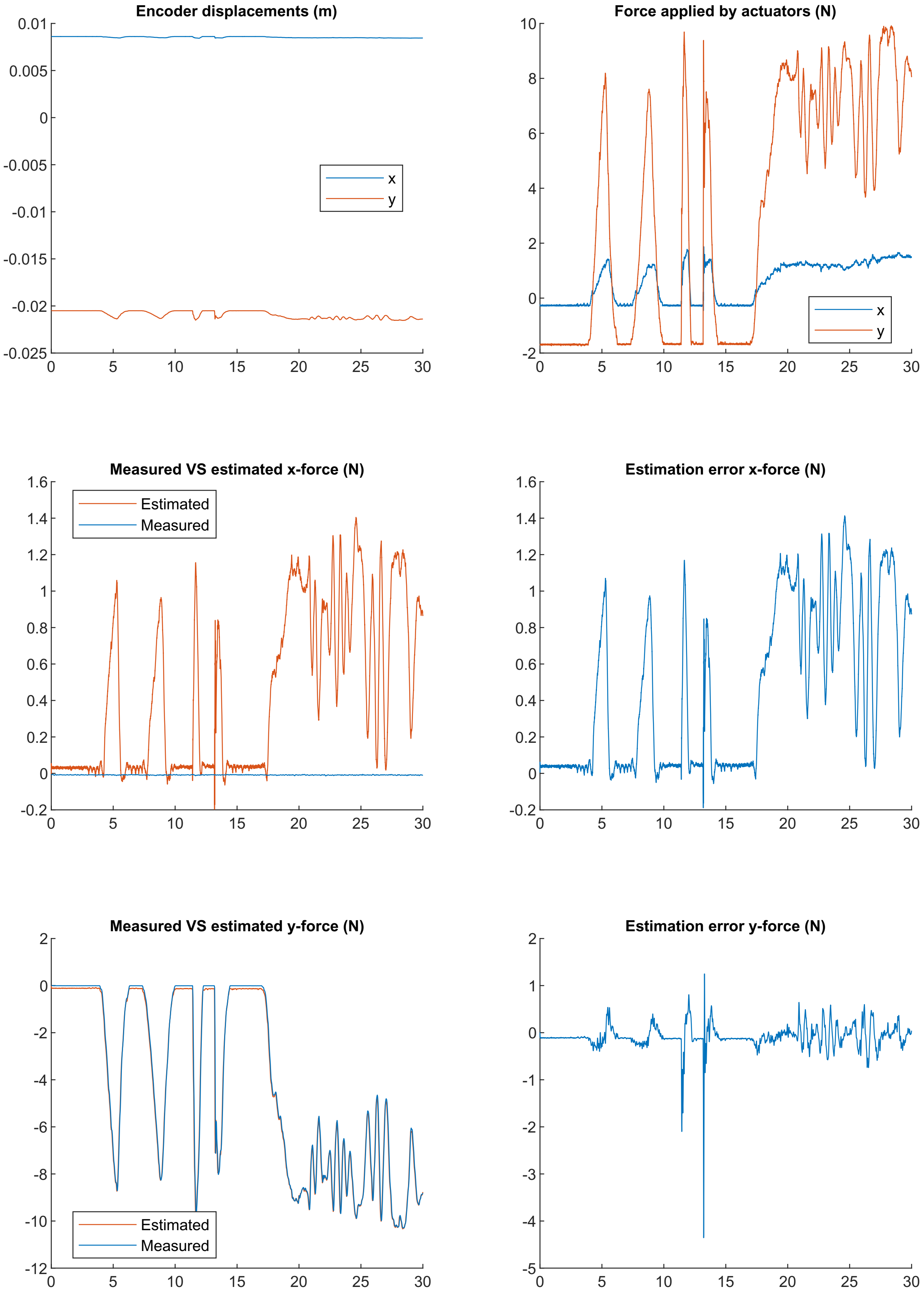


Figure 4.16: End-effector held in place at $x_{eff} = 5mm$, $y_{eff} = -20mm$

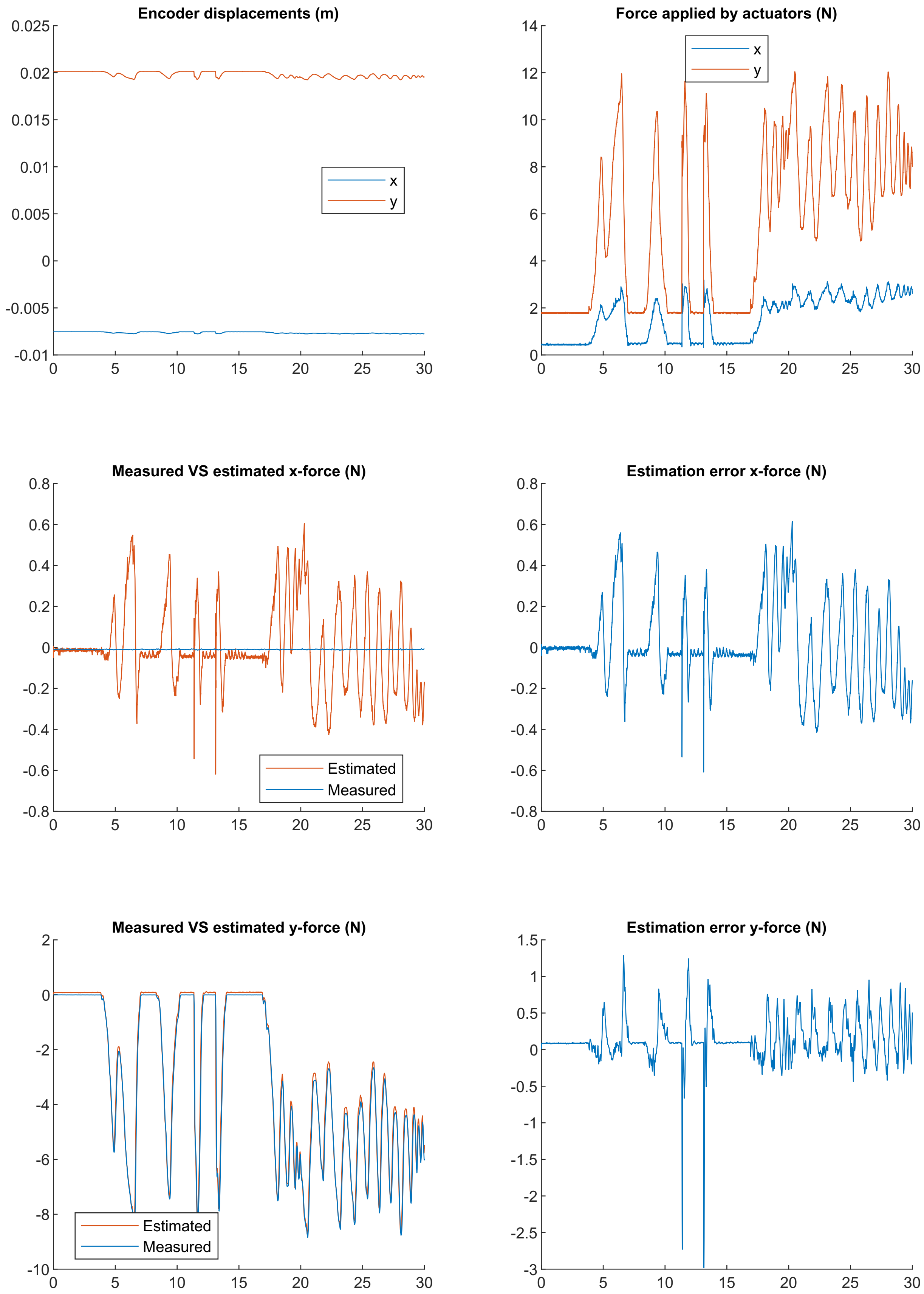


Figure 4.17: End-effector held in place at $x_{eff} = -5mm$, $y_{eff} = 20mm$

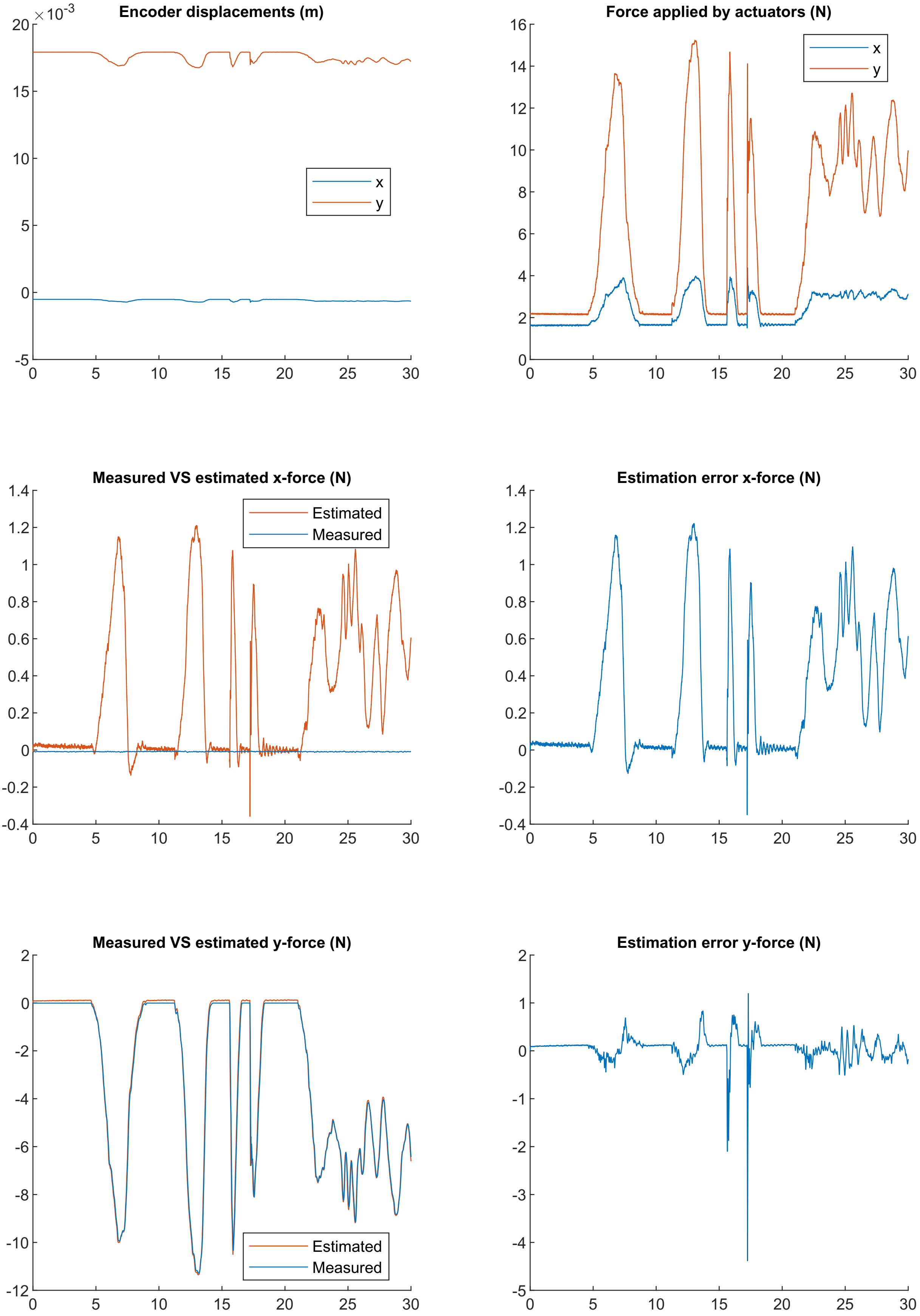


Figure 4.18: End-effector held in place at $x_{eff} = 5mm$, $y_{eff} = 20mm$

4.4.2 Results: displacement by external force without actuators

In a second round of tests, the actuators were turned off completely while the sensors were pushed against the end-effector by hand. This data can be used to test the performance of the LKF based on the dynamic behavior of the 2DOF manipulator without the influence of the actuators. The results are illustrated in figures 4.19 to 4.22.

In figures 4.19 and 4.20, the y-force sensor was pushed against the end-effector while the x-force sensor was not in contact with the end-effector. The estimated values for the y-forces seem very accurate. However, the estimated x-forces are not zero as they should be. Again, this could be explained by friction induced by the force sensor. In an attempt to reduce the influence of this friction, these experiments were repeated where it was ensured that the end-effector remained in contact with the x-sensor as well. The results for these experiments are shown in figures 4.21 and 4.22. Although it cannot be guaranteed that the effects of friction were removed, the estimates of the x-forces seem to have improved slightly.

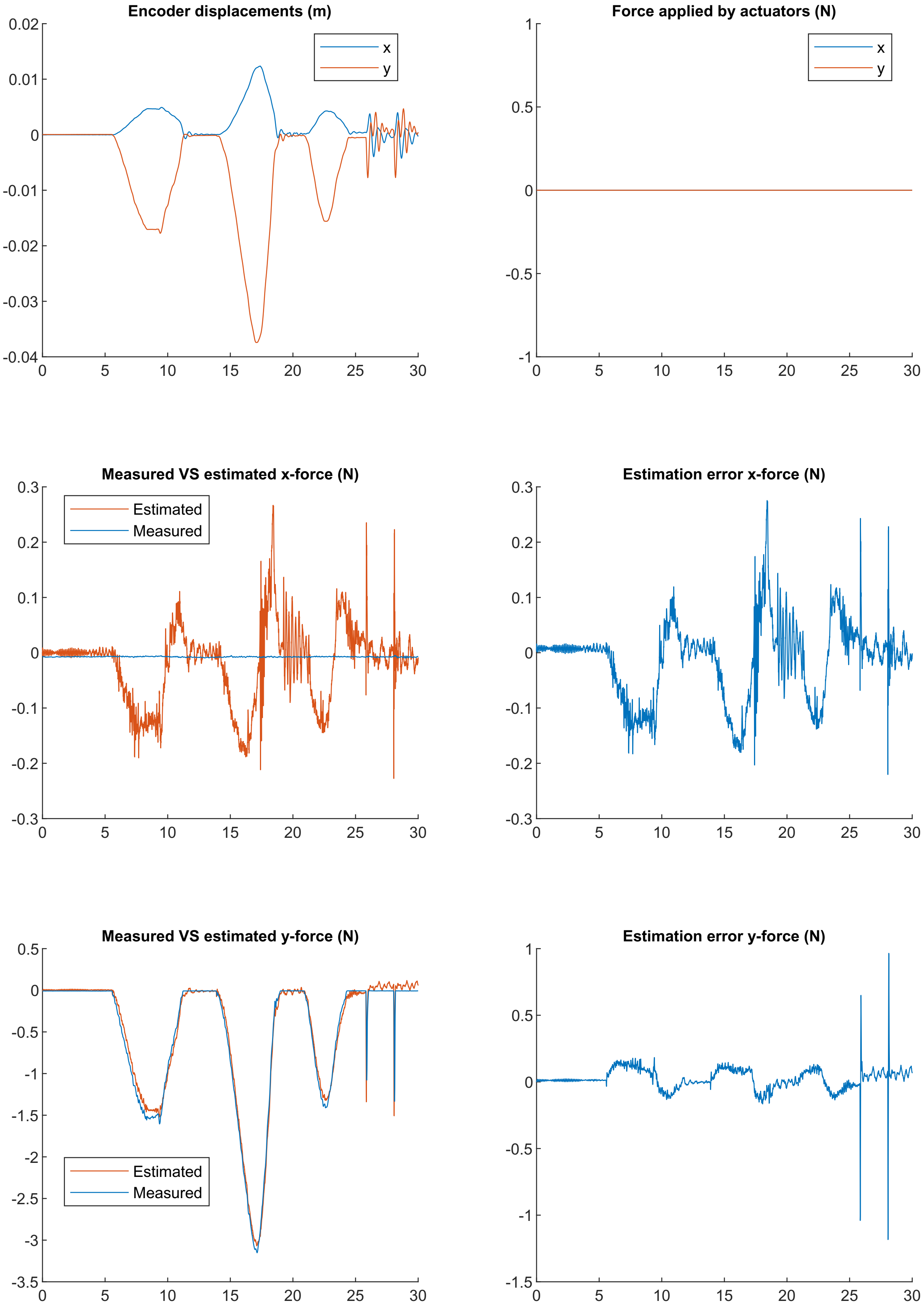


Figure 4.19: End-effector pushed in y-direction without actuator data set 1

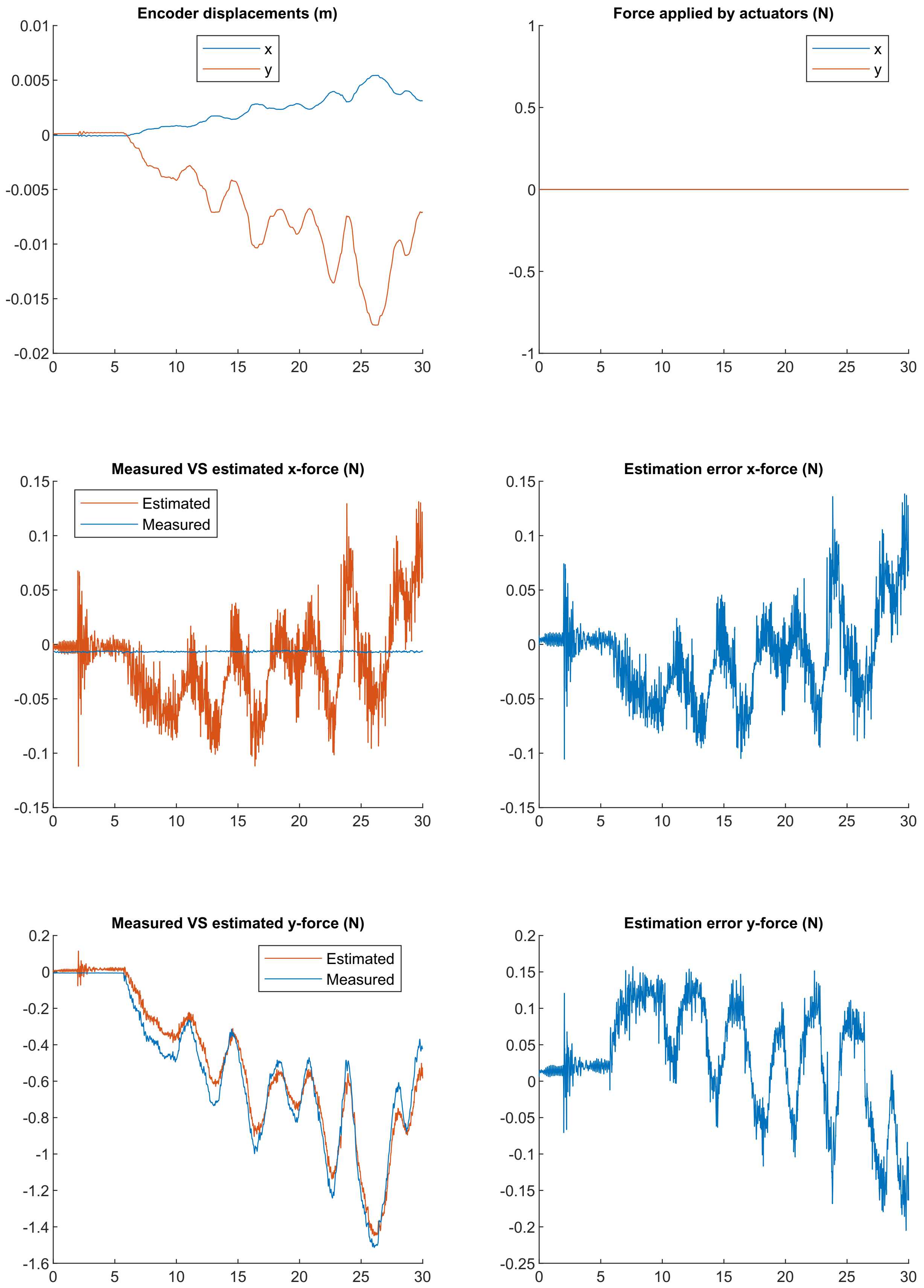


Figure 4.20: End-effector pushed in y-direction without actuator data set 2

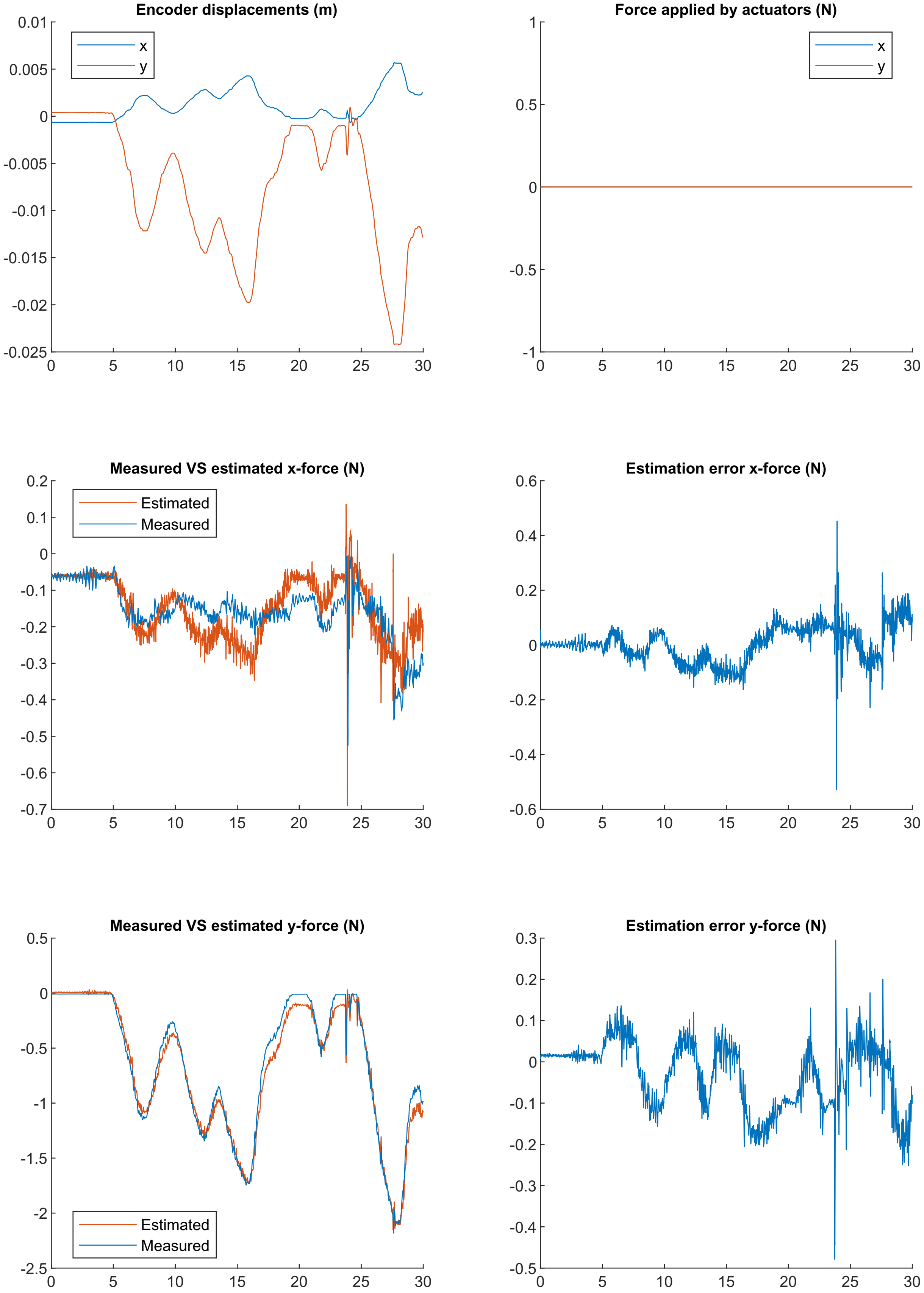


Figure 4.21: End-effector pushed in y-direction without actuator and contact with both sensors data 1

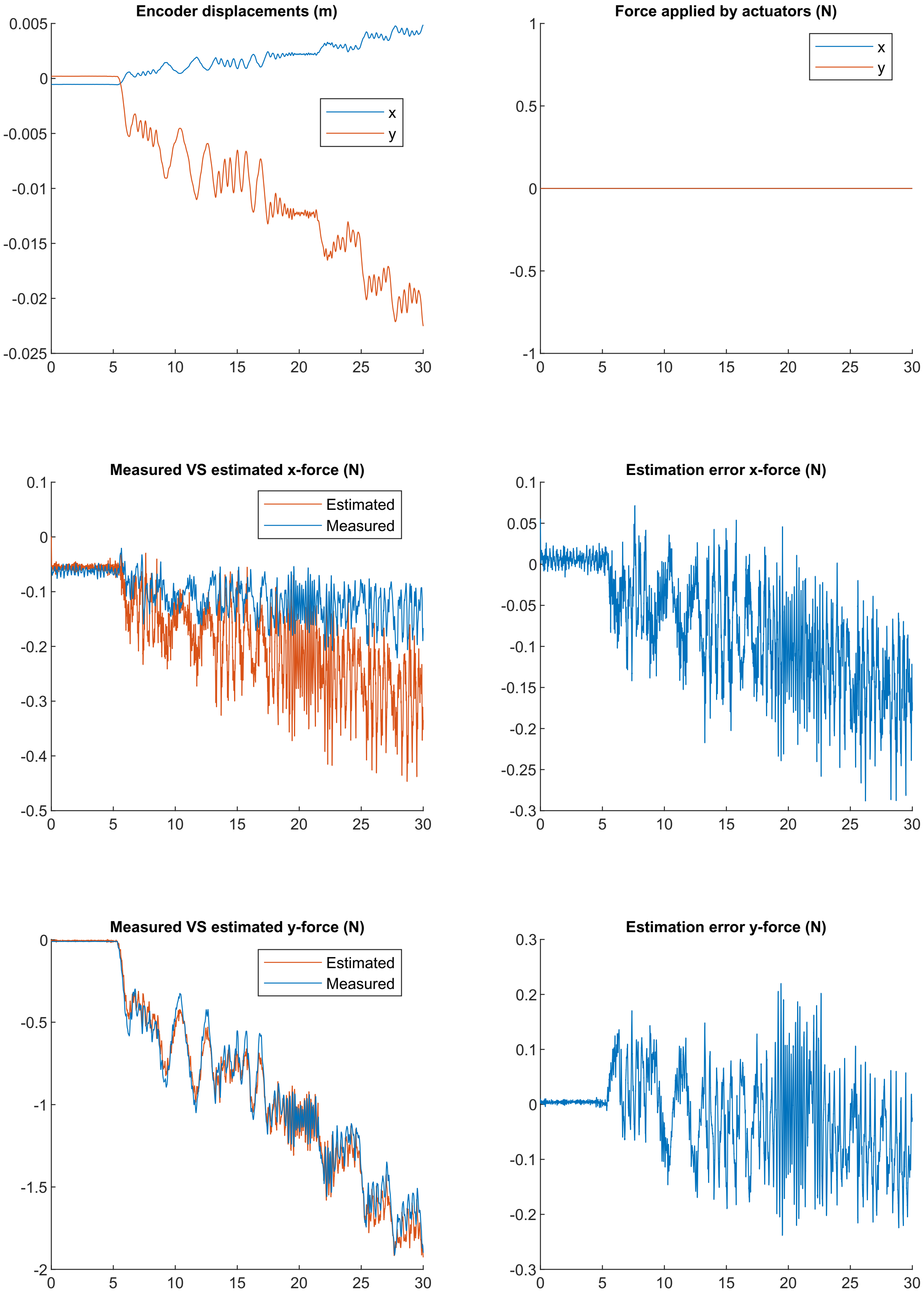


Figure 4.22: End-effector pushed in y-direction without actuator and contact with both sensors data 2

4.4.3 Results: end-effector following a reference path without external forces

In a third round of tests, the end-effector was set to travel along a reference path without any external forces acting on the system. This data tests the performance of the EKF when both the actuators and the dynamic behavior of the system play a role, but without the influence of external forces. The results are shown in figures 4.23 and 4.24. Figure 4.23 shows the results where the reference path was set to move in a plus-shape in the first ten seconds. Afterward, the end-effector was instructed to move to and hold multiple different positions for a short time. Figure 4.24 shows the results where the reference path was set to move in spirals with varying radius at first. Then, after a brief delay, the end-effector was instructed to move in a circular path.

The results of these tests are quite satisfactory. The estimation errors mostly lie within a range of $0.1N$. The estimated forces in figure 4.24 have a small offset, which can be recognized at the times when the system is in its resting position. This is probably caused by a small error in the calibration of the encoders, since the y-actuator is also inducing a force in its resting position.

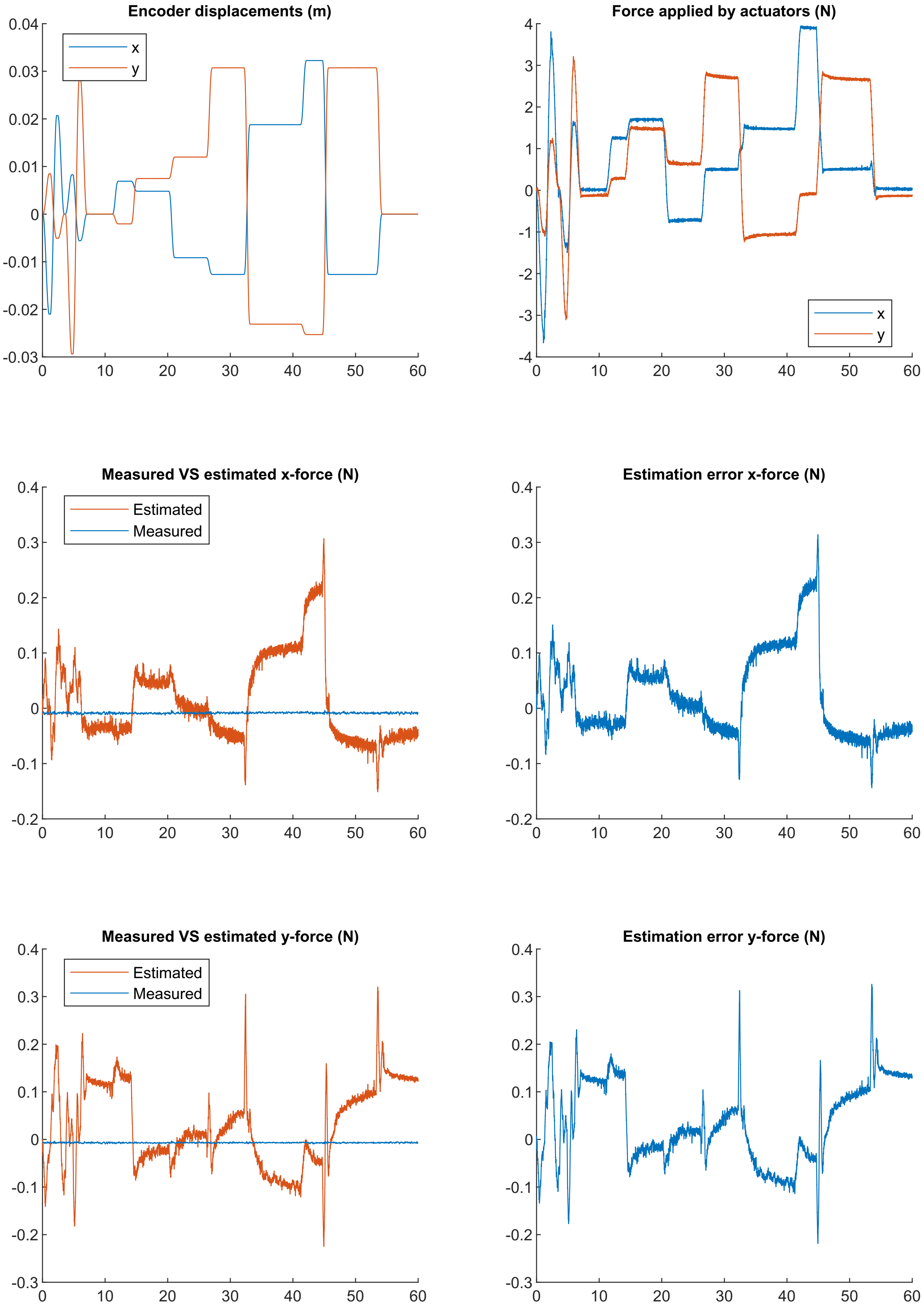
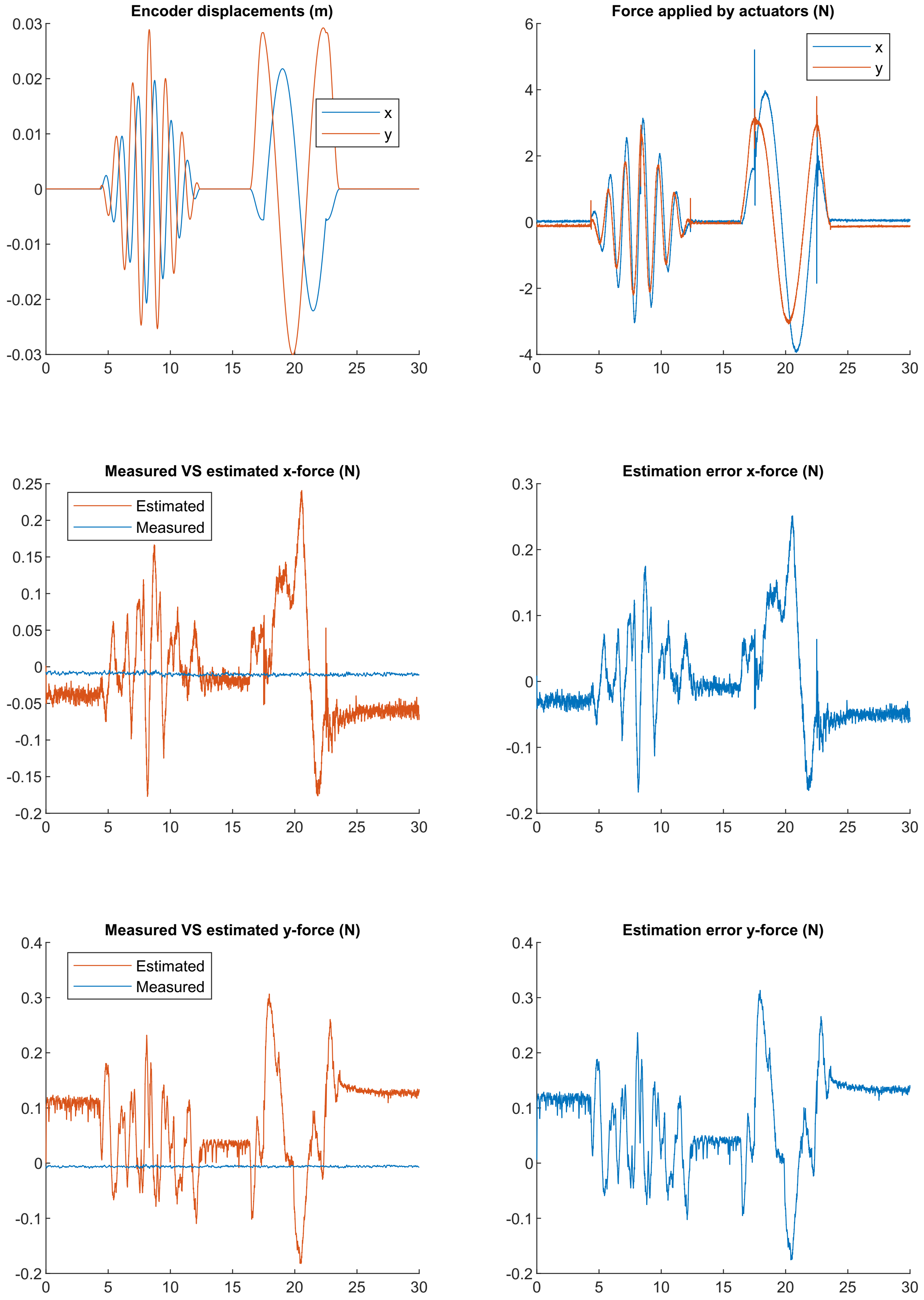


Figure 4.23: End-effector reference-path cross, followed by different positions


Encoder displacements (m)

Force applied by actuators (N)

Measured VS estimated x-force (N)

Estimation error x-force (N)

Measured VS estimated y-force (N)

Estimation error y-force (N)

Figure 4.24: End-effector reference-path spirals, followed by circle

4.4.4 Results: end-effector colliding with sensor on its reference path

In the last round of tests, the end-effector was set to travel along a reference path where it would collide with the force sensors. The results are shown in figures 4.25 and 4.26.

In figure 4.25, the end-effector started in a deflected position, and was set to collide with the x-force sensor after a delay. These results show that the forces are also accurately estimated in x-direction, and similar errors as found before show up in the y-force estimates.

Figure 4.26 shows the results where the end-effector path was set to collide with the y-force sensor. In this experiment, a small mass was placed on the rail to increase the external force exerted. The end-effector collides multiple times with the sensor during the measurements, taking a small step back between each collision. The sensors are able to move in y-direction, so the end-effector pushes the sensors along the rails during the experiment. The results are not perfect, but agreeable. Again, similar estimation errors coupled to the y-forces are observed as before in the x-forces.

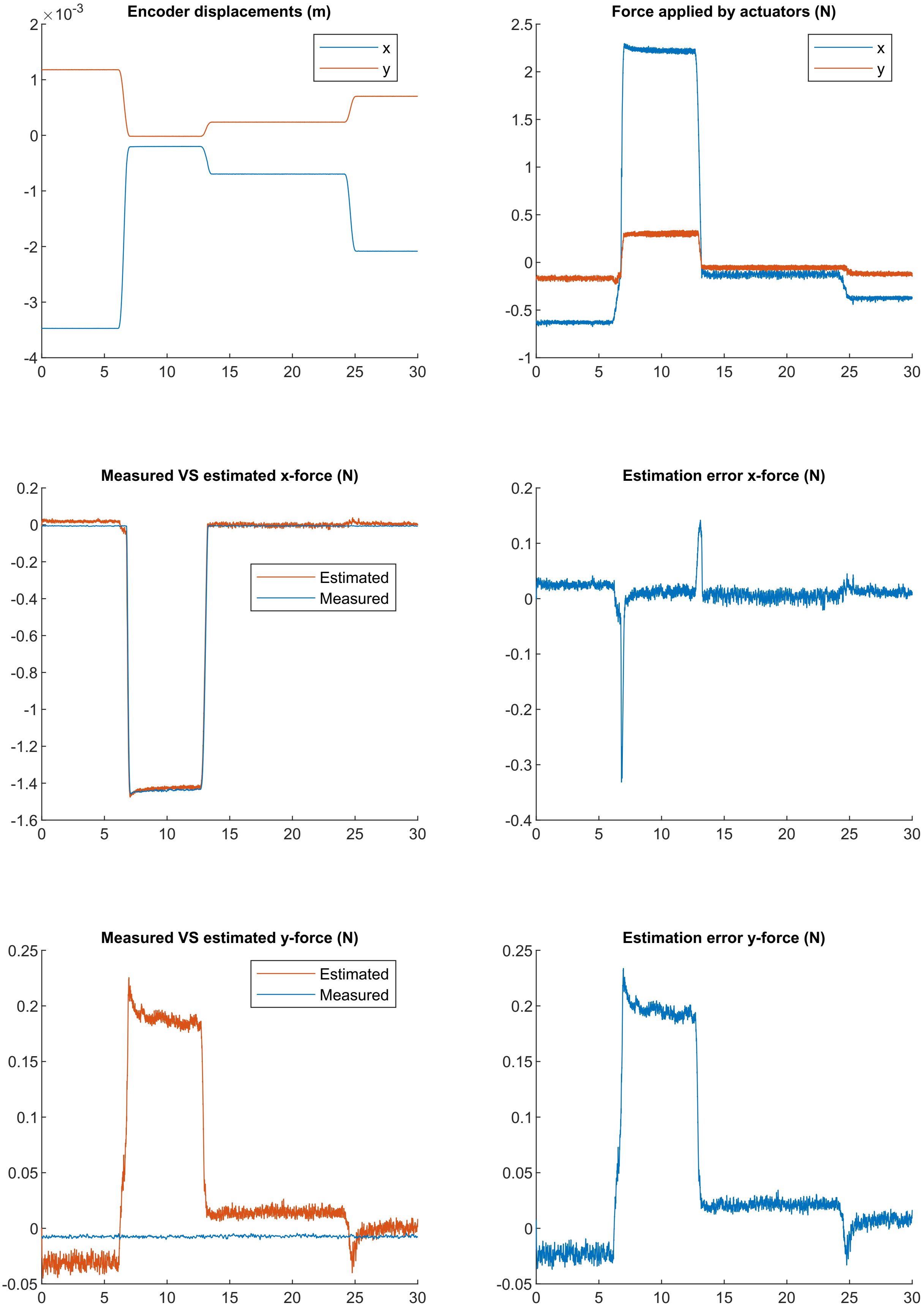


Figure 4.25: End-effector collides with x-force sensor

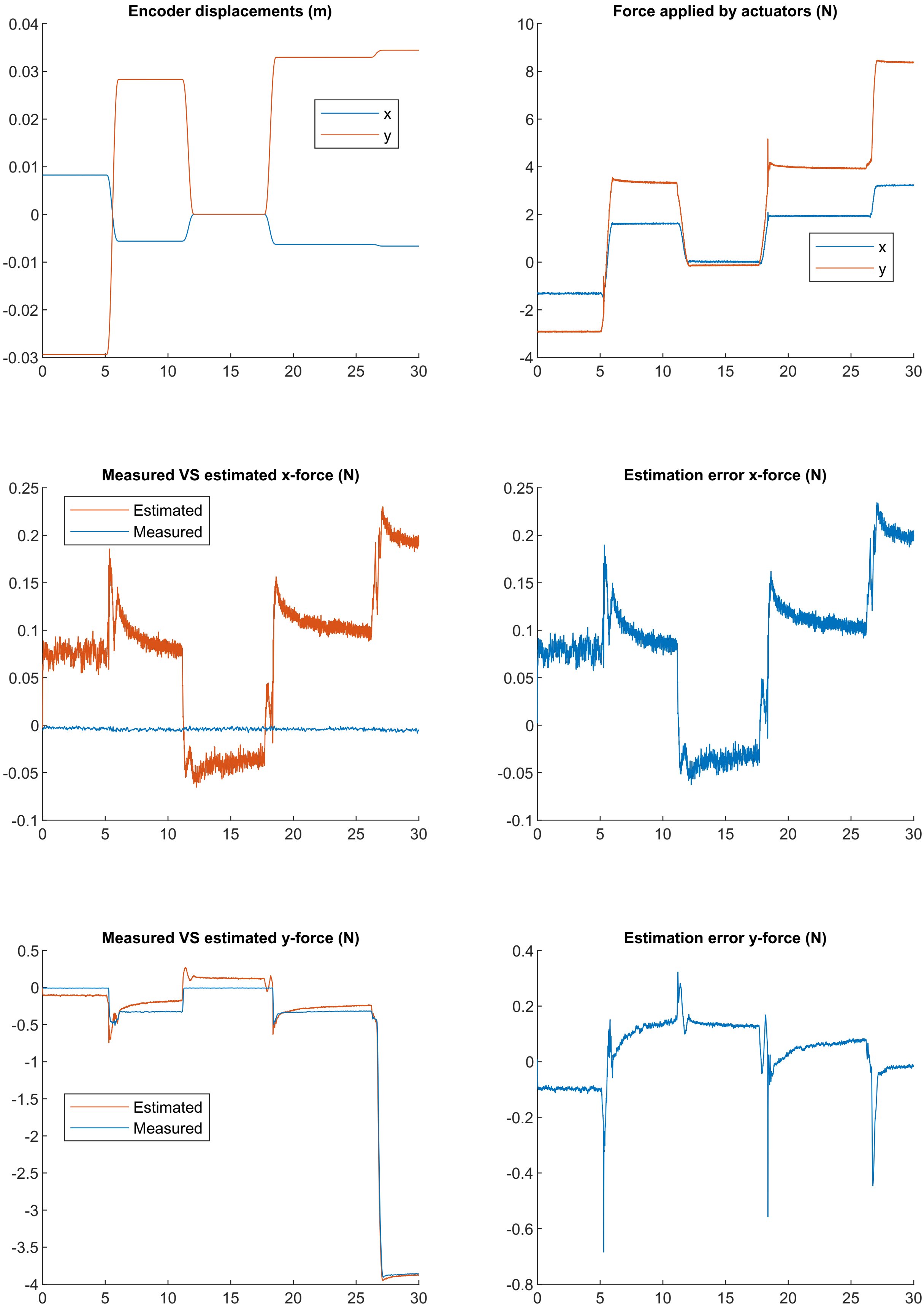


Figure 4.26: End-effector collides with y-force sensor

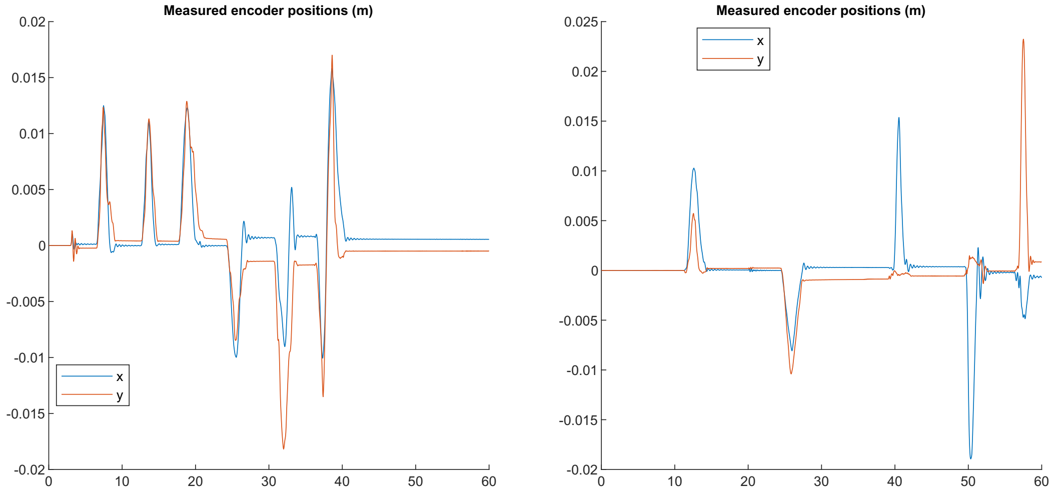
4.5 Discussion: 2DOF manipulator experiments

There were some limitations and complications in the testing phase, which will be discussed next. The force sensors were mounted on a rail, which limited the range of positions in which the system could be tested while external forces were acting on the system. The rail only allows the sensors to move in y-direction, so data for the external forces could only be collected when the end-effector was positioned on the y-axis.

Furthermore, because the force sensors were fixed in a construction on the rails in vertical position, they could not be calibrated. Comparing the estimated forces to the force data collected, it was notable that the estimates were exactly a scalar multiple of the force sensor data. It was assumed that this was caused by incorrect calibration of the force sensors. To fix this, the data was corrected by scaling it by the inverse of that factor. This seemed plausible, as the estimates significantly improved for all data sets gathered. Nevertheless, this step does not improve the reliability of this data. Additionally, it was found that the force sensors had an offset in their measurements. This was observed in data where there were no forces acting on the sensors. These offsets have been removed from the measured data as well.

During data collections, it was observed that the force sensors also applied friction forces to the end-effector tangential to the direction in which they measure. After a force was applied and removed, the end-effector would 'shoot' back in direction perpendicular to the force that was being applied. These friction forces were not be measured by the other sensor, hence they have a detrimental effect on the validation data.

Another observation that was made is that when the end-effector was nudged slightly from its nominal position, the system did not revert back to its original position. Two data sets from which this can be seen are shown in figure 4.27. This occurred despite that there were no actuator or external forces (intentionally) acting on the system. It is suspected that these offsets are caused by hysteresis induced by the cables of the sensors and actuators. Although these external forces may 'correctly' be estimated by the LKF, they were not included in the validation data measured by the force sensors.



(a) Data set 1

(b) Data set 2

Figure 4.27: 2DOF manipulator rest positions after multiple nudges

Finally, the encoders were calibrated multiple times to set the nominal position of the 2DOF manipulator to zero. This was not done perfectly, however. As a result, some of the encoder measurements have a slight offset. Where this has occurred, a small error will be introduced in the external forces estimated by the LKF.

It is possible that the estimated external forces are more accurate than they appear. Any of the mentioned uncertainties may have negatively influenced the validation data. For example, the friction induced by the force sensors is not present in the validation data, but will be estimated by the LKF. This was illustrated in section 4.4.2, where the estimated external forces were improved slightly by ensuring contact with both force sensors. As shown in figure 4.27, it is clear that the system is influenced by hysteresis. Forces caused by hysteresis will also be included in the estimated external forces, but these are not measured either.

The estimation errors could also have been caused by modelling errors. As was found from the tests on the double pendulum, differences in the dynamic behavior between the real system and the model used by the LKF will result in fictitious forces to be estimated. Based on the shape of the estimation errors, it is suspected that this has occurred in the 2DOF manipulator tests as well. It is striking that the estimation errors generally are of the same shape as the external forces and actuator forces applied in perpendicular direction. The dynamic behavior of the 2DOF manipulator in x- and y-directions is coupled, which can be recognized from figure 4.26 for example. In this experiment, a movement purely in y-direction was accomplished by applying a force with both actuators. It is possible that the coupling between these forces is not correctly incorporated in the model. As a result, the 'expected displacement' caused by these actuator forces based on the model would not comply with the measured displacement. The LKF will attribute this difference to a fictitious external force acting on the system, which will lead to an estimation error. However, it is difficult to determine to what extent the estimation errors are caused by modelling errors or uncertainties such as friction.

In conclusion, it is difficult to judge the performance of the LKF exactly based on the data gathered with the current setup. The validation data does not include all external forces acting on the system, which makes it possible that the estimation errors are in actuality smaller than they appear. The estimation errors could also be caused by modelling errors, but the effects of modelling errors and uncertainties such as friction cannot be differentiated from the results. Despite all complications, the estimated external forces seem reasonably accurate. In general, the magnitude of the estimation errors depend on the magnitudes of the applied actuator and external forces. The magnitude of the estimation errors were in general within a range of 10% of the magnitude of these forces.

5 CONCLUSION

In this report, external force estimation was applied based on linearized Kalman filters (LKF). A method was derived to set the LKF parameters, and this method was tested based on a double pendulum model in simulations. The method gave promising results based on tests on a double pendulum in simulations. In ideal circumstances, the external forces could be estimated very accurately. It was found that the quality of the results heavily depend on the accuracy of the model, however. Any difference in the dynamic behavior of the real system and the model used by the LKF will result in fictitious forces observed in the estimates.

The LKF was subsequently applied to the 2DOF manipulator using the developed method. Although uncertainties were minimized in the design of the 2DOF manipulator, it was found that these were not absent in the experiments. The force sensors used to collect validation data as well as the cables connected to the actuators and encoders exerted (unmeasured) friction forces. Although these forces will be estimated by the LKF, these were not included in the validation data. The estimation errors could also have been caused by errors in the model used by the LKF, but it is difficult to determine to what extent the estimation errors were caused by undeterministic behavior or modelling errors.

Although the results were influenced by friction and hysteresis, their contributions were small enough such that the estimation errors were reasonably small. The magnitude of the estimation errors were dependent on the magnitude of the external and actuator forces applied. In general, the estimation errors did not exceed a magnitude of 10% of the external and actuator forces. Since it is likely that the validation data does not include all external forces acting on the 2DOF manipulator, the performance of the LKF might be better than shown by these experiments.

5.1 Recommendation

From the results in this report, it was difficult to determine to what extent estimation errors were caused by modelling errors or errors in the validation data. The performance of the LKF could be judged better if the validation data represents the external forces on the end-effector more accurately. Main contributions to the errors in the validation data were (unmeasured) friction forces. Some friction was caused by the cables conntected to the actuators and encoders, although these effects were already minimized in the design process of the 2DOF manipulator.

A more promising point of improvement is the force sensor setup. The force sensors were set up such that these could apply a friction force tangential to its surface. This has a detrimental effect on the validation data, since this force would not be measured. Furthermore, the current setup limited the extent to which the LKF could be tested. The sensors could move on a rail in only one direction. As a result, the LKF could only be tested in a limited range of positions of the 2DOF manipulator.

REFERENCES

[1] K.G.P. Folkersma. Design and evaluation of a 2-dof large stroke stage with elastic elements. Master's thesis, University of Twente, 2011.

[2] K. Hoitzing. Kinematic and dynamic modeling for the conceptual design of a 2-dofs large stroke elastic positioning mechanism. Master's thesis, University of Twente, 2010.

[3] Shih-Hsuan Chien, Jyun-Hsiang Wang, and Ming-Yang Cheng. Performance comparisons of different observer-based force-sensorless approaches for impedance control of collaborative robot manipulators. In *2020 IEEE conference on industrial cyberphysical systems (ICPS)*, volume 1, pages 326–331. IEEE, 2020.

[4] Aaron Radke and Zhiqiang Gao. A survey of state and disturbance observers for practitioners. In *2006 American Control Conference*, pages 6–pp. IEEE, 2006.

[5] Sang-Duck Lee, Min-Cheol Kim, and Jae-Bok Song. Sensorless collision detection for safe human-robot collaboration. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2392–2397. IEEE, 2015.

[6] Gijo Sebastian, Zeyu Li, Vincent Crocher, Demy Kremers, Ying Tan, and Denny Oetomo. Interaction force estimation using extended state observers: An application to impedance-based assistive and rehabilitation robotics. *IEEE Robotics and Automation Letters*, 4(2):1156–1161, 2019.

[7] Chowarit Mitsantisuk, Kiyoshi Ohishi, Shiro Urushihara, and Seiichiro Katsura. Kalman filter-based disturbance observer and its applications to sensorless force control. *Advanced Robotics*, 25(3-4):335–353, 2011.

[8] Ramsey Faragher. Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]. *IEEE Signal processing magazine*, 29(5):128–132, 2012.

[9] John L. Crassidis and John L. Junkins. *Optimal estimation of dynamic systems*. Chapman and Hall/CRC, 2012.

[10] Loris Roveda and Dario Piga. Sensorless environment stiffness and interaction force estimation for impedance control tuning in robotized interaction tasks. *Autonomous Robots*, 45(3):371–388, 2021.

[11] R.G.K.M. Aarts. *private communication*. March 2022.

[12] J.B. Jonker, R.G.K.M. Aarts, and J.P. Meijaard. *Flexible multibody dynamics for design purposes — A Finite Element Approach*. University of Twente, 2020.

[13] Janusz S Przemieniecki. *Theory of matrix structural analysis*. Courier Corporation, 1985.

[14] E. Heerze. Data-driven feedforward control of a multi degree of freedom manipulator with flexure joints using machine learning. Master's thesis, University of Twente, 2021.

A PARTIAL DERIVATIVES OF THE DOUBLE PENDULUM EQUATIONS OF MOTION

This appendix shows expressions for the partial derivatives of the functions f_3 and f_4 in (3.21) with respect to each state x_i and input u_i. In the derivation of these expressions, the partial derivative of the denominator of the summation terms in the functions f_3 and f_4 with respect to x_2 was found as,

$$\frac{\partial}{\partial x_2}\left[\frac{1}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos\left(x_2\right)}\right] = -\frac{\sin\left(x_2\right)}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos\left(x_2\right)\right]^2}. \tag{A.1}$$

$$\frac{\partial f_3}{\partial x_1} = -\frac{m_2 L_2^2 k_1}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos\left(x_2\right)} \tag{A.2}$$

$$
\begin{aligned}
\frac{\partial f_3}{\partial x_2} = {} & \frac{m_2L_2^2k_1x_1\sin(x_2)}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}+\frac{\left[m_2L_2^2+m_2L_1L_2\cos(x_2)\right]k_2}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}\\
&-\frac{m_2L_1L_2k_2\sin(x_2)\,x_2}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}-\frac{\left[m_2L_2^2+m_2L_1L_2\cos(x_2)\right]k_2\sin(x_2)\,x_2}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}\\
&+\frac{m_2L_2^2d_1\sin(x_2)\,x_3}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}+\frac{m_2^2L_1L_2^3\cos(x_2)\,x_3^2}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}\\
&-\frac{m_2^2L_1L_2^3\sin^2(x_2)\,x_3^2}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}+\frac{m_2^2L_1^2L_2^2\cos(2x_2)\,x_3^2}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}\\
&-\frac{\frac{1}{2}m_2^2L_1^2L_2^2\sin(x_2)\sin(2x_2)\,x_3^2}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}+\frac{2m_2^2L_1L_2^3\cos(x_2)\,x_3x_4}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}\\
&-\frac{2m_2^2L_1L_2^3\sin^2(x_2)\,x_3x_4}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}-\frac{m_2L_2^2d_2\sin(x_2)\,x_4}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}\\
&-\frac{\frac{1}{2}m_2L_1L_2d_2\sin(2x_2)\,x_4}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}-\frac{m_2L_1L_2d_2\sin(x_2)\,x_4}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}\\
&+\frac{m_2^2L_1L_2^3\cos(x_2)\,x_4^2}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}-\frac{m_2^2L_1L_2^3\sin^2(x_2)\,x_4^2}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}\\
&-\frac{m_2L_2^2\sin(x_2)\,x_5}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}+\frac{m_2L_1L_2\sin(x_2)\,x_6}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}\\
&+\frac{\left[m_2L_2^2+m_2L_1L_2\cos(x_2)\right]\sin(x_2)\,x_6}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}-\frac{m_2L_2^2\sin(x_2)\,u_1}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}\\
&+\frac{m_2L_1L_2\sin(x_2)\,u_2}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}+\frac{\left[m_2L_2^2+m_2L_1L_2\cos(x_2)\right]\sin(x_2)\,u_2}{\left[\left[m_1+m_2\right]L_1L_2-m_2L_1L_2\cos(x_2)\right]^2}
\end{aligned}
\tag{A.3}
$$

$$
\begin{aligned}
\frac{\partial f_3}{\partial x_3} = {} & -\frac{m_2L_2^2d_1}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}+\frac{2m_2^2L_1L_2^3\sin(x_2)+m_2^2L_1^2L_2^2\sin(2x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}x_3\\
&+\frac{2m_2^2L_1L_2^3\sin(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}x_4
\end{aligned}
\tag{A.4}
$$

$$
\begin{aligned}
\frac{\partial f_3}{\partial x_4} = {} & \frac{m_2L_2^2d_2+m_2L_1L_2d_2\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}+\frac{2m_2^2L_1L_2^3\sin(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}x_3\\
&+\frac{2m_2^2L_1L_2^3\sin(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}x_4
\end{aligned}
\tag{A.5}
$$

$$
\frac{\partial f_3}{\partial x_5} = \frac{m_2L_2^2}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}
\tag{A.6}
$$

$$
\frac{\partial f_3}{\partial x_6} = -\frac{m_2L_2^2+m_2L_1L_2\cos(x_2)}{\left[m_1m_2+m_2^2\right]L_1^2L_2^2-m_2^2L_1^2L_2^2\cos(x_2)}
\tag{A.7}
$$

$$\frac{\partial f_3}{\partial u_1} = \frac{m_2 L_2^2}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \text{cos}\left(x_2\right)} \tag{A.8}$$

$$\frac{\partial f_3}{\partial u_2} = -\frac{m_2 L_2^2 + m_2 L_1 L_2 \text{cos}\left(x_2\right)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \text{cos}\left(x_2\right)} \tag{A.9}$$

$$\frac{\partial f_4}{\partial x_1} = \frac{m_2 L_2^2 + m_2 L_1 L_2 \text{cos}\left(x_2\right)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \text{cos}\left(x_2\right)} k_1 \tag{A.10}$$

$$
\begin{aligned}
\frac{\partial f_4}{\partial x_2} = & -\frac{m_2 L_1 L_2 k_1 x_1 \sin(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} - \frac{\left[m_2 L_2^2 + m_2 L_1 L_2 \cos(x_2)\right] k_1 x_1 \sin(x_2)}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} \\
& - \frac{\left[\left[m_1 + m_2\right] L_1^2 + m_2 L_2^2 + 2 m_2 L_1 L_2 \cos(x_2)\right] k_2}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} + \frac{2 m_2 L_1 L_2 k_2 \sin(x_2) x_2}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} \\
& + \frac{\left[\left[m_1 + m_2\right] L_1^2 + m_2 L_2^2 + 2 m_2 L_1 L_2 \cos(x_2)\right] k_2 \sin(x_2) x_2}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} - \frac{m_2 L_2^2 d_1 \sin(x_2) x_3}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} \\
& - \frac{\frac{1}{2} m_2 L_1 L_2 d_1 \sin(2x_2) x_3}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} - \frac{m_2 L_1 L_2 d_1 \sin(x_2) x_3}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} \\
& - \frac{2 m_2^2 L_1 L_2^3 \cos(x_2) x_3 x_4}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} + \frac{2 m_2^2 L_1 L_2^3 \sin^2(x_2) x_3 x_4}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} \\
& - \frac{2 m_2^2 L_1^2 L_2^2 \cos(2x_2) x_3 x_4}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} + \frac{m_2^2 L_1^2 L_2^2 \sin(x_2) \sin(2x_2) x_3 x_4}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} \\
& - \frac{\left[m_1 m_2 + m_2^2\right] L_1^3 L_2 \cos(x_2) x_3^2}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} + \frac{\left[m_1 m_2 + m_2^2\right] L_1^3 L_2 \sin^2(x_2) x_3^2}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} \\
& - \frac{m_2^2 L_1 L_2^3 \cos(x_2) x_3^2}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} + \frac{m_2^2 L_1 L_2^3 \sin^2(x_2) x_3^2}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} \\
& - \frac{2 m_2^2 L_1^2 L_2^2 \cos(2x_2) x_3^2}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} + \frac{m_2^2 L_1^2 L_2^2 \sin(x_2) \sin(2x_2) x_3^2}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} \\
& + \frac{\left[\left[m_1 + m_2\right] L_1^2 d_2 + m_2 L_2^2 d_2\right] \sin(x_2) x_4}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} + \frac{m_2 L_1 L_2 d_2 \sin(2x_2) x_4}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} \\
& + \frac{2 m_2 L_1 L_2 d_2 \sin(x_2) x_4}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} - \frac{m_2^2 L_1 L_2^3 \cos(x_2) x_4^2}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} \\
& + \frac{m_2^2 L_1 L_2^3 \sin^2(x_2) x_4^2}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} - \frac{m_2^2 L_1^2 L_2^2 \cos(2x_2) x_4^2}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} \\
& + \frac{\frac{1}{2} m_2^2 L_1^2 L_2^2 \sin(x_2) \sin(2x_2) x_4^2}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} + \frac{m_2 L_1 L_2 \sin(x_2) x_5}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} \\
& + \frac{\left[m_2 L_2^2 + m_2 L_1 L_2 \cos(x_2)\right] \sin(x_2) x_5}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} - \frac{2 m_2 L_1 L_2 \sin(x_2) x_6}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} \\
& - \frac{\left[\left[m_1 + m_2\right] L_1^2 + m_2 L_2^2 + 2 m_2 L_1 L_2 \cos(x_2)\right] \sin(x_2) x_6}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} \\
& + \frac{m_2 L_1 L_2 \sin(x_2) u_1}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} + \frac{\left[m_2 L_2^2 + m_2 L_1 L_2 \cos(x_2)\right] \sin(x_2) u_1}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2} \\
& - \frac{2 m_2 L_1 L_2 \sin(x_2) u_2}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} - \frac{\left[\left[m_1 + m_2\right] L_1^2 + m_2 L_2^2 + 2 m_2 L_1 L_2 \cos(x_2)\right] \sin(x_2) u_2}{\left[\left[m_1 + m_2\right] L_1 L_2 - m_2 L_1 L_2 \cos(x_2)\right]^2}
\end{aligned} \tag{A.11}
$$

$$
\begin{aligned}
\frac{\partial f_4}{\partial x_3} = & \frac{m_2 L_2^2 d_1 + m_2 L_1 L_2 d_1 \cos(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} - \frac{2 m_2^2 L_1 L_2^3 \sin(x_2) + m_2^2 L_1^2 L_2^2 \sin(2x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} x_4 \\
& - \frac{2 \left[m_1 m_2 + m_2^2\right] L_1^3 L_2 \sin(x_2) + 2 m_2^2 L_1 L_2^3 \sin(x_2) + 2 m_2^2 L_1^2 L_2^2 \sin(2x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} x_3
\end{aligned} \tag{A.12}
$$

$$\frac{\partial f_4}{\partial x_4} = -\frac{[m_1+m_2]\,L_1^2 d_2 + m_2 L_2^2 d_2 + 2m_2 L_1 L_2 d_2 \cos(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} - \frac{2m_2^2 L_1 L_2^3 \sin(x_2) + m_2^2 L_1^2 L_2^2 \sin(2x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} x_3 - \frac{2m_2^2 L_1 L_2^3 \sin(x_2) + m_2^2 L_1^2 L_2^2 \sin(2x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} x_4 \tag{A.13}$$

$$\frac{\partial f_4}{\partial x_5} = -\frac{m_2 L_2^2 + m_2 L_1 L_2 \cos(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} \tag{A.14}$$

$$\frac{\partial f_4}{\partial x_6} = \frac{[m_1+m_2]\,L_1^2 + m_2 L_2^2 + 2m_2 L_1 L_2 \cos(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} \tag{A.15}$$

$$\frac{\partial f_4}{\partial u_1} = -\frac{m_2 L_2^2 + m_2 L_1 L_2 \cos(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} \tag{A.16}$$

$$\frac{\partial f_4}{\partial u_2} = \frac{[m_1+m_2]\,L_1^2 + m_2 L_2^2 + 2m_2 L_1 L_2 \cos(x_2)}{\left[m_1 m_2 + m_2^2\right] L_1^2 L_2^2 - m_2^2 L_1^2 L_2^2 \cos(x_2)} \tag{A.17}$$

B ALTERNATIVE DERIVATION OF TRANSFORMATION MATRIX

In section 4.2.4, the equations of motion were transformed from the perspective of the independent coordinates to the end-effector positions. This was achieved by using the inverse of a sub-matrix. This inverse needs to be calculated using Matlab's symbolic toolbox. However, due to the complexity of the symbolic expressions, Matlab's symbolic toolbox has difficulties calculating the inverse. This appendix offers an alternate approach to find the transformation matrix, where an inverse needs not be calculated.

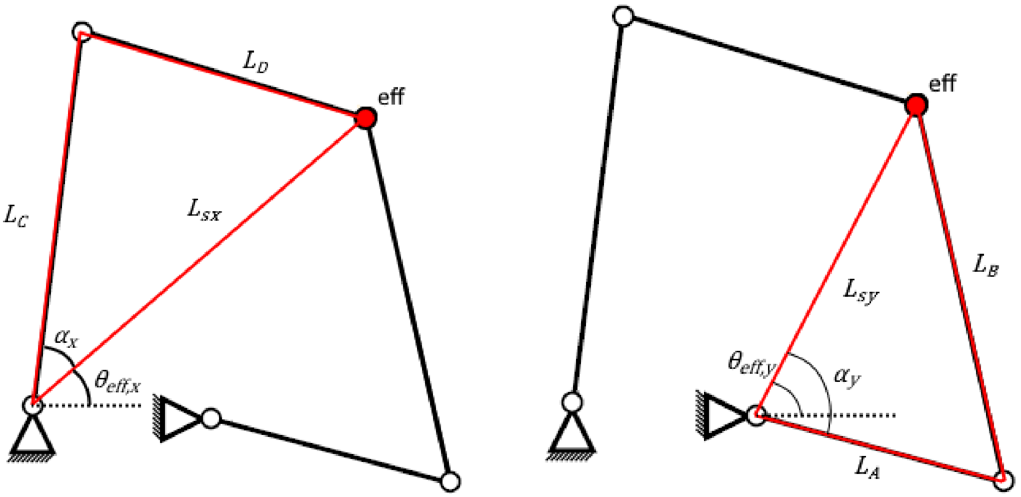
The goal is to find the transformation matrix that reflects the forces on the independent coordinates $\boldsymbol{q}$ to the end-effector position $\boldsymbol{x}_{eff}$. This can be achieved by defining the kinematic relation of the independent coordinates expressed in terms of the positions of the end-effector as

$$\begin{aligned} \boldsymbol{q} &= \boldsymbol{\mathcal{G}}(\boldsymbol{x}_{eff})\,, \\ \dot{\boldsymbol{q}} &= \boldsymbol{\mathcal{G}}_{,x_{eff}}\dot{\boldsymbol{x}}_{eff}. \end{aligned} \tag{B.1}$$

The forces can be transformed by applying the Jacobian as

$$\boldsymbol{F}_{eff} = \boldsymbol{\mathcal{G}}^{T}_{,x_{eff}}\boldsymbol{F}_{q}. \tag{B.2}$$

The geometric transfer function $\boldsymbol{\mathcal{G}}(\boldsymbol{x}_{eff})$ can be found by defining the triangles shown in figure B.1.



(a) Triangle x-arm

(b) Triangle y-arm

Figure B.1: Triangles defined to determine the geometric transfer function

To find and expression for the geometric transfer function $\boldsymbol{\mathcal{G}}(\boldsymbol{x}_{eff})$, the angles α_i and $\theta_{eff,i}$ are required. The angles $\theta_{eff,i}$ can be found from the end-effector position and the arctangent as

$$
\begin{aligned}
\theta_{eff,x} &= \tan^{-1}\left(\frac{y_{10}-y_2}{x_{10}-x_2}\right), \\
\theta_{eff,y} &= \tan^{-1}\left(\frac{y_{10}-y_1}{x_{10}-x_1}\right).
\end{aligned} \tag{B.3}
$$

The angles α_i can be found using the cosine rule as

$$
\begin{aligned}
\alpha_x &= \cos^{-1}\left(\frac{L_C^2+L_{sx}^2-L_D^2}{2L_C L_{sx}}\right), \\
\alpha_y &= \cos^{-1}\left(\frac{L_A^2+L_{sy}^2-L_B^2}{2L_A L_{sy}}\right).
\end{aligned} \tag{B.4}
$$

where the lengths L_{sx} and L_{sy} can be found as

$$
\begin{aligned}
L_{sx} &= \sqrt{\left[x_{10}-x_2\right]^2+\left[y_{10}-y_2\right]^2}, \\
L_{sy} &= \sqrt{\left[x_{10}-x_1\right]^2+\left[y_{10}-y_1\right]^2}.
\end{aligned} \tag{B.5}
$$

An expression for the angles of the lower arms θ_A and θ_C as a function of the end-effector position then be found as

$$
\begin{aligned}
\theta_A &= \theta_{eff,y}-\alpha_y, \\
\theta_C &= \theta_{eff,x}+\alpha_x.
\end{aligned} \tag{B.6}
$$

Finally, an expression for the encoder positions is derived by rewriting equation 4.9 as

$$
\begin{aligned}
x_{enc} &= r_{enc}\left[\theta_{C,nom}-\theta_C\right], \\
y_{enc} &= r_{enc}\left[\theta_A-\theta_{A,nom}\right],
\end{aligned} \tag{B.7}
$$

which gives an expression for the geometric transfer function $\boldsymbol{\mathcal{G}}(\boldsymbol{x}_{eff})$. The transformation matrix $\boldsymbol{\mathcal{G}}_{,x_{eff}}$ can be found using Matlab's symbolic toolbox.

C VALIDATION OF MODELLING METHOD APPLIED ON 2DOF MANIPULATOR

The process of deriving a model for the 2DOF manipulator using Matlab's symbolic toolbox can be validated using the double pendulum model. State-space matrices for the double pendulum derived analytically are already available. To test the modelling process used on the 2DOF manipulator in chapter 4, the same process can be repeated on the double pendulum and compared to the analytically derived state-space matrices. Applying the parameter values described in section 3.5 to the analytically derived state-space matrices (3.24) gives

$$\boldsymbol{A}_{lin} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -399 & 1287 & -0.4 & 0.8 & 3920 & -9147 \\ 931 & 4611 & 0.9 & -3 & -9147 & 32761 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B}_{lin} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 3920 & -9147 \\ -9147 & 32761 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{C.1}$$

Then, an alternative set of state-space matrices can be derived by applying the process described in chapter 4. The equations of motion of the double pendulum (3.7) are defined in Matlab using its symbolic toolbox. These symbolic expressions are then linearized as in section 4.2.6, and converted to a (continuous-time) state-space system as in section 4.2.7. The resulting state-space matrices can be compared to the analytically derived state-space matrices (C.1) by subtracting as

$$\begin{aligned} \Delta\boldsymbol{A} &= \boldsymbol{A}_{lin} - \boldsymbol{A}_{alt}, \\ \Delta\boldsymbol{B} &= \boldsymbol{B}_{lin} - \boldsymbol{B}_{alt}. \end{aligned} \tag{C.2}$$

with $\Delta\boldsymbol{A}$ and $\Delta\boldsymbol{B}$ the difference between the state-space matrices, and $\boldsymbol{A}_{alt}$ and $\boldsymbol{B}_{alt}$ the state-space matrices derived using Matlab's symbolic toolbox.

The differences between these state-space matrices become

$$\Delta\boldsymbol{A} = 10^{-12} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0001 & 0 & 0 & 0 \\ 0.1137 & 0 & -0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \Delta\boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{C.3}$$

Given the order of magnitude of the elements shown in (C.1), these differences are insignificant. This indicates that the methods used in chapter 4 to derive a model for the 2DOF manipulator are valid.