Impedance Control for a 2-DOF Compliant Manipulator using Kalman Filter-Based State Estimation

Jesse J. Peters
Bachelor Advanced Technology, University of Twente, Bachelor Thesis
July 25, 2024

***Abstract*—This study proposes an enhanced impedance control scheme for a 2-DOF compliant manipulator, building on an existing scheme by introducing an accelerometer and a linear Kalman filter for improved state estimation. The manipulator features flexure hinges, reducing friction and providing high predictability of the system. System identification is achieved by analysing the position response to a Pseudo Random Binary Sequence (PRBS) force input, which provided the mass, damping and stiffness matrices of the system. These are used according to the equation of motion, together with the Kalman filter position and velocity estimates and the direct acceleration measurements, to cancel the admittance of the system. Virtual mass, damping and stiffness matrices are used according to the equation of motion, using the same feedback parameters to follow a reference while mimicking a desired impedance behaviour. Several simulation studies are done, where the working of the Kalman filter and the controller as a whole are tested. Validation of the controller was conducted in a simulation environment, where the natural frequency and damping ratio are obtained for several virtual systems and their corresponding desired systems. This is done by conducting three tests, where the virtual mass, stiffness and damping ratio parameters are varied, respectively. The results are beneath expectations, mainly due to deviation and uncertainty in the position response of the system, caused by inaccurate velocity estimation.**

Keywords: impedance control, compliant manipulator, 2-DOF motion, accelerometer, encoder, sensor fusion, linear Kalman filter, state estimation

INTRODUCTION

Interaction of robots with their surroundings is becoming an increasingly more important subject in robotics with the current advancements in this field[1]. Besides operation of robots in structured environments, like in a factory, an increasing number of projects arise that bring robots in a more unstructured environment, such as everyday life. With robots being implemented in our daily lives rather than in a controlled environment, the ability to move safely through a certain space has a more significant role in the design choices made. This is due to interaction of robots with not only surrounding objects, but also with humans.

Impedance control is a useful concept which helps in making robots safe and functional in such environments. With impedance control, the mechanical impedance of the robot, that is, the mapping of velocities to forces, is altered[2]. For this, the impedance of the system is described as a mass-spring-damper system. Applying impedance control makes a robot mimic a certain desired virtual impedance, described by virtual mass, damping and stiffness matrices. This way, the behaviour can be adjusted, which is highly important in interaction with humans or objects. For example, the mass, damping and stiffness parameters can be set such that hard collisions can be avoided and such that the robotic mechanism will comply more, showing 'softer' behaviour. This can avoid harm, where using only position control might result in hard collisions and unwanted behaviour of the robot, in the case of a change in the environment.

Impedance control uses position, velocity and acceleration feedback from the plant to compute a suiting actuator force to mimic a desired impedance behaviour[2][3]. In this paper, a method for impedance control is proposed to improve upon an already existing impedance control scheme for a 2-DOF complaint manipulator[3], which uses two position encoders for the position, velocity and acceleration determination. A schematic illustration of the corresponding setup is shown in Figure 1. In this study, research is conducted to the addition of an accelerometer to the setup, to fuse the corresponding acceleration signals with the encoder position signals using a Kalman filter. This achieves an improved state estimation (position and velocity) and a direct acceleration measurement. The velocity estimation is especially important, as the velocity is not directly measured but has a significant influence on the performance of the impedance controller.

By introducing an accelerometer to the setup, multiple changes can be made to the controller to include the accelerometer influence in the actuator force determination process. The implementation, optimization and challenges that arise with this are discussed. Simulation studies are employed to validate the controller performance, also by comparing to the previous real-time implementation. The corresponding research question is: *How can an accelerometer and a Kalman filter be used to achieve more accurate impedance control?*

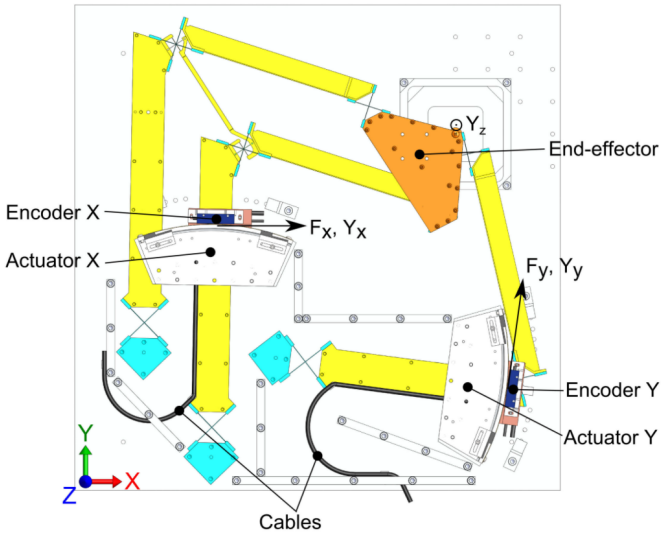


Fig. 1: 2-DOF compliant manipulator setup[4]

I. Previous Research

The 2-DOF compliant manipulator[4] used in this study has been utilized before in research, mainly to force estimation and force and impedance control. Lübbermann proposed a control scheme[3] for impedance control on the system, which showed promising results. The control scheme was tested using free motion to mimic the impedance of a virtual system as closely as possible, for a certain range of virtual parameters. Uitbeijerse continued the research to impedance control on the same setup[5], by using the same control scheme. The accuracy of the system identification was improved and the impedance control was tested for two degrees of freedom. Also, the range over which the virtual parameters could be changed was improved in this study. The same impedance controller used in these two previous studies will be used and improved upon in this study.

Also, the manipulator setup has been used in research to force control. Here, the idea of implementing an accelerometer, for a direct acceleration measurement and feedback signal, was already introduced by Hakvoort[6] and Kluin[7]. This resulted in reasonably well force tracking, however, only for low frequencies. For frequencies above 1.5 Hz, the velocities and accelerations are too high for the controller to keep up with the reference. In this study, the same accelerometer will be used, and a linear Kalman filter will be applied to fuse the sensor data of the two encoders and the accelerometer that are used on the setup, in order to improve the impedance control.

II. Impedance Control

The impedance control implementation can be explained by considering the underlying theory and the corresponding control scheme illustrations. The controller could be showcased, including details about the implementation of the equations from the theory.

A. Control scheme

1) Theory: The impedance control scheme that was proposed by Lübbermann[3] is built up from three main components, which are the plant, the inverse plant and the control impedance parts. In both the inverse plant and the control impedance parts, force computation is conducted through two similar equations, based on the equation of motion. These forces are summed to get the desired input force, which is fed to the actuators in the plant.

The 'system force' $\vec{F}_s$ is computed in the inverse plant according to the equation of motion of the system, described as

$$\vec{F}_s = M\ddot{\vec{x}} + D\dot{\vec{x}} + K\vec{x} \tag{1}$$

Here, the identified mass, damping and stiffness matrices of the system (M, D and K, respectively) are used in combination with the measured position and the corresponding derivatives. The resulting force is used to cancel the admittance of the system.

The 'virtual force' $\vec{F}_v$ is computed in the control impedance part according to

$$\vec{F}_v = M_v\vec{e}_a + D_v\vec{e}_v + K_v\vec{e}_p \tag{2}$$

Here, the virtual mass, damping and stiffness matrices (M_v, D_v and K_v, respectively) are used in combination with the error between the reference and the measured position, and the derivatives of this error. This force is used to obtain the desired impedance behaviour while following a certain reference.

The M, D and K system matrices are obtained through system identification, while the M_v, D_v and K_v virtual system matrices are achieved according to

$$M_v = I \cdot m_v \tag{3}$$

$$D_v = I \cdot 2\zeta_v\sqrt{m_v k_v} \tag{4}$$

$$K_v = I \cdot k_v \tag{5}$$

Here, m_v and k_v are the virtual mass and stiffness parameters, ζ_v is the virtual damping ratio and I represents the 2×2 identity matrix. The virtual damping matrix is determined using the virtual damping coefficient c_v for a harmonic oscillator, defined as

$$c_v = 2\zeta_v\sqrt{m_v k_v} \tag{6}$$

In the final controller, these virtual mass, damping and stiffness parameters can be altered to obtain different desired impedance behaviours through Eqs. (3) - (5).

2) Current implementation: This study uses the three main components in the control scheme, according to the equations from the theory. However, the implementation of the scheme will be improved using an accelerometer and a Kalman filter. This provides the system with a Kalman filter state estimate, which contains position and velocity, and a direct acceleration signal. These enhanced position, velocity and acceleration signals can be used in the inverse plant and control impedance parts, according to Eqs. (1) and (2). For the inverse plant, the enhanced signals are directly used in the force computation. For the control impedance, the enhanced signals are used in combination with the first and second derivatives of the reference to compute the direct errors. The corresponding improved scheme is illustrated in Figure 2.

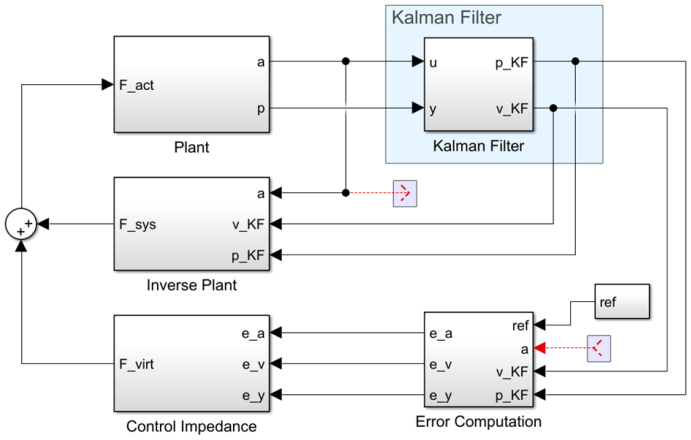


Fig. 2: Impedance control scheme

B. Controller

The impedance controller is based on the control scheme in Figure 2. A simplified version of the controller is shown in Figure 3, in which the most important parts are illustrated. The compensator part contains the force computation based on the system and virtual system, which is illustrated in Figure 4. Shown is how the position, velocity and acceleration error could be computed directly using the reference, the estimated position and velocity from the Kalman filter and the acceleration measurement. This ensures that the accelerometer and Kalman filter influences are not only taken into account in the system part of the compensator, but also in the virtual part. The accelerometer signal can not be fed directly into the compensator, because of the sensor noise. Therefore, a low-pass filter is designed carefully to remove the most significant part of this noise.

C. Kinematics

One of the outputs of the plant block in Figure 3 is the end-effector position. The setup uses two position encoders to measure the position of the X and Y upper arms. These encoders are placed against the actuators as shown in Figure 1. To achieve the end-effector position from the encoder measurements, a forward kinematics model was used, based on the position and orientation of the hinges and trigonometry. This model is used within the plant block in the controller.

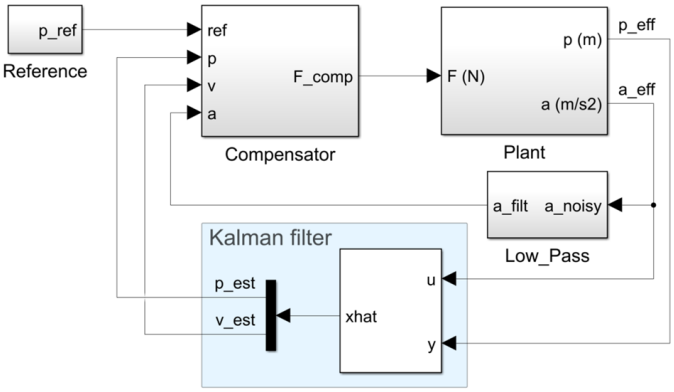


Fig. 3: Improved impedance controller

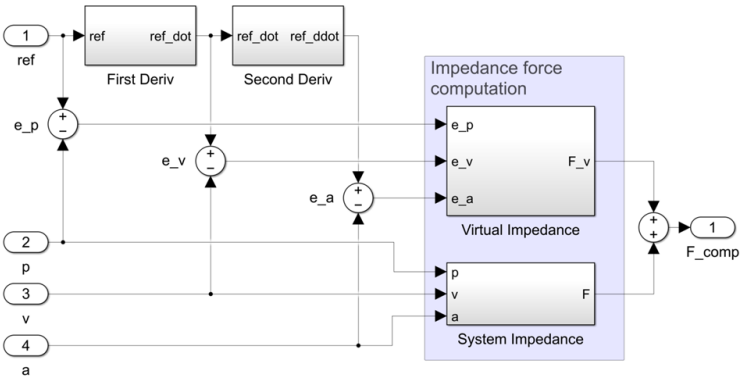


Fig. 4: Compensator, impedance force computation

III. Setup

The 2-DOF compliant manipulator setup[4] used in this study was designed and built by K.G.P. Folkersma for studying the implementation of a high precision mechanism in a vacuum. A schematic representation of the setup is shown in Figure 1. The flexure-based design of the system ensures high predictability with little uncertainties. Also, this prevents contamination of the vacuum due to the increased wear and evaporation of lubricants that might occur when using regular joints. With a maximum deflection of the flexure hinges of 20° and a workspace of 100×100 mm for the end-effector, the footprint of the system is minimized. The maximum dimensions of the system as a whole are $540 \times 585 \times 87$ mm. The setup consists of two encoder position sensors near the actuators and an accelerometer at the end-effector.

A. Encoders

The setup has only one type of sensor installed by default, namely the position encoders on both the X and Y arms. These encoders are used as main source for feedback from the system and are the main method to allow closed loop control. The

encoders are located on the upper arm parts of one of the X arms and the Y arm, against the actuators, as shown in figure 1. The setup uses Heidenhain LIDA 479 encoders of the quad A/B type, which are selected by Folkersma[4]. The encoders have a resolution of $0.05\,\mu\mathrm{m}$, which makes for an accurate position reading.

B. Accelerometer

For this study the dual-axis ADXL203EB accelerometer is mounted on the end-effector. This same accelerometer has been used in previous work on the manipulator, where Hakvoort[6] has conducted research to the implementation of an accelerometer in a force control scheme to obtain improved force control. In this paper, the accelerometer is used to obtain a better real-time estimate of the state, consisting of position and velocity, through a Kalman filter in the controller. Also, the direct acceleration measurement can be used in the impedance controller instead of differentiating the encoder position twice. These enhanced position, velocity and acceleration data are used in the compensator part of the controller as shown in Figure 4.

The used accelerometer has a range of approximately $\pm 1.7g$ and has an adjustable bandwidth according to

$$f_{-3dB} = \frac{5\mu F}{C_{(X,Y)}} \tag{7}$$

By changing the value of $C_{(X,Y)}$, which corresponds to the combined capacitance of the capacitors C_1 and C_2, bandwidths can be achieved in the range of 0.5 Hz to 2.5 kHz. The capacitors were changed by Hakvoort to obtain $C_{X,Y} = 5$ nF, which achieves a bandwidth of 1 kHz, which was found to be sufficiently large for the application. The x and y output signals of the accelerometer are connected to two analog input ports on a National Instruments BNC-2110 board, of which the corresponding block can be found in the Simulink library to fetch the data and use it in the controller. An external 5 V power supply is used to power the accelerometer.

An important part of this study is the calibration of the accelerometer, as this has a large influence on the success of the state estimation and the controller as a whole. The developers of the device recommend linear calibration, which could be done using the known gravitational acceleration g. The calibration was based on the sensitivity and the zero-g bias of the accelerometer, according to

$$a = \frac{V - V_{zero,g}}{S} \cdot 9.81 + a_{bias} \tag{8}$$

where V is the measured voltage, $V_{zero,g}$ is the zero-g bias voltage, S is the sensitivity of the accelerometer and a_{bias} is used to correct for any acceleration offset in the calibration. The zero-g bias voltage and sensitivity could be found in the data sheet of the accelerometer and were slightly adjusted during the calibration process. The latter was done by implementing Eq. (8) in a Simulink model, and holding the accelerometer upright in all four directions and in a level, stationary position. The voltage and acceleration measurements could be used to make adjustments to the parameters for both axes. The a_{bias} parameter was used to make sure that the acceleration signal was zero when in a stationary position. This was important regarding the controller, where an offset in acceleration causes an offset in the force input of the system. Therefore, this was considered a priority during the calibration. Due to slight nonlinearities in the behaviour of the accelerometer, this has resulted in a slight offset at accelerations around $-g$ and g, which does not have too much influence on the behaviour of the Kalman filter and the controller as a whole. The Curve Fitting Toolbox in Matlab was used to create residual plots of both the x- and y-direction, which are shown in Figure 15a and 15b, respectively, in Appendix A. Here, the slight nonlinearities are shown to be small enough to use this linear calibration method.

IV. System Identification

A. Identification procedure

In order to implement impedance control in real-time and in simulation, system identification had to be performed. This provided insight in the system parameters related to mass, damping and stiffness. The choice for linear and nonlinear identification had to be made carefully, since the system is nonlinear in nature. Also, the choice for SIMO or MIMO identification is relevant for the resulting system matrices.

1) Stiffness matrix: For the stiffness matrix determination, a basic position controller was used in Simulink to move the end-effector of the manipulator to different predetermined locations. The corresponding end-effector position measurement and actuator force input are retrieved from the setup. This data was used to create a force lookup table corresponding to different x and y position values, which is used in the impedance controller to directly fetch the stiffness force contribution, based on the current position of the end-effector. This was done in order to capture the nonlinearities in the stiffness behaviour that occur when the end-effector moves away from the equilibrium position for such an amount that it might not be accurate to consider the system as linear anymore.

2) Mass and damping matrices: For the mass and damping matrix determination, linear identification has been performed, because the deviations from the equilibrium position during this study are expected to be low enough to consider linear identification to be accurate enough. The system was excited using a PRBS force signal, such that the end-effector deviated around the equilibrium position. The resulting position response and corresponding actuator force input were retrieved from the setup. This excitation technique is a suiting method for the identification of such a 2-DOF manipulator, as it provides insight in the mass and damping behaviour of the system over a large range of frequencies. Having the actuators apply this force to the system in multiple ways resulted in

various identification possibilities regarding the inputs and outputs of the system. Both MIMO and SIMO identification were performed for identification of the system, of which the results were compared in simulation. MIMO identification was regarded as most accurate, because this captures all dynamics of the system as accurately as possible. However, for the mass matrix determination, SIMO identification provided the best results. For the damping matrix determination, MIMO identification provided the best results. The System Identification Toolbox in Matlab could be used for the analysis of the data and the creation of the state space models, described as

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{9}$$

Here, the state x is defined to contain the position and velocity, the output y is defined to contain the position and the input u is defined to contain the force as in

$$x = \begin{pmatrix} \vec{x} \\ \dot{\vec{x}} \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \\ v_x \\ v_y \end{pmatrix} \tag{10}$$

$$y = \vec{x} = \begin{pmatrix} p_x \\ p_y \end{pmatrix}, \qquad u = \vec{F} = \begin{pmatrix} F_x \\ F_y \end{pmatrix} \tag{11}$$

Note that this respresentation is for the MIMO system and that the SIMO identification uses two systems with $u = F_x$ and $u = F_y$, respectively. The state space matrices take form according to a fourth order system implementation and the corresponding MIMO or SIMO method, where D is a zero matrix. The corresponding dimensions are shown in Table I.

Matrix	MIMO	SIMO
A	4×4	4×4
B	4×2	4×1
C	2×4	2×4
D	2×2	2×1

TABLE I: Dimensions system matrices MIMO and SIMO identification

For the experimental setup, a basic open-loop Simulink controller was used, which applied a PRBS force signal with an amplitude of ± 2 N to both actuators simultaneously or separately. A sample time of 0.0001 s and a clock time of 20 were used as options within the block. This resulted in the PRBS signal changing every 0.0020 s, providing excitation of the system up to 500 Hz.

B. Data processing

1) Stiffness matrix: The input force and resulting displacement data could be plotted for the whole end-effector workspace, which is shown in Figure 5. The stiffness matrix can be determined by fitting the force and displacement data using Matlab, according to

$$K = \begin{bmatrix} \frac{\delta F_x}{\delta x} & \frac{\delta F_x}{\delta y} \\ \frac{\delta F_y}{\delta x} & \frac{\delta F_y}{\delta y} \end{bmatrix} = \begin{bmatrix} 121.8 & 55.1 \\ 42.4 & 95.9 \end{bmatrix} \frac{N}{m} \tag{12}$$

For the implementation in the impedance controller, the whole stiffness lookup table was used.

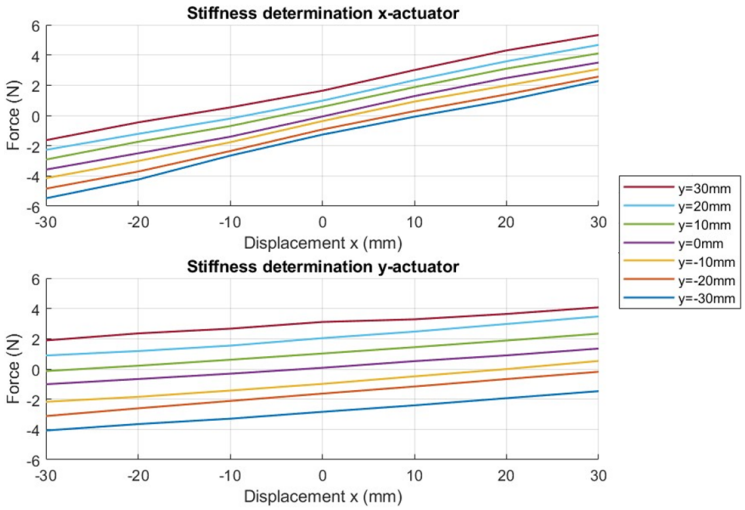


Fig. 5: Force-displacement plots of both actuators, used for the determination of the stiffness matrix

2) Mass matrix: The input forces and measured position response from the PRBS excitation can be used in the System Identification Toolbox to perform MIMO identification. The mean of the position signals was removed and the resulting data was used to create a fourth order state space model, which was provided as the most suiting fit based on the number of singular values. The frequency responses of the different I/O combinations were used in the determination of the mass matrix. The corresponding bode plots are shown together in Figure 6.

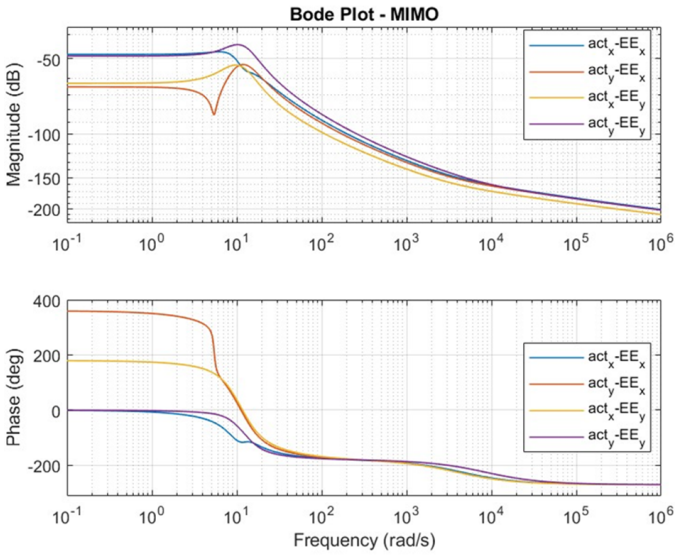


Fig. 6: Bode plots of all I/O combinations; MIMO identification

The corresponding MIMO equation that describes the transfer

functions in Figure 6 can be defined in the frequency domain as

$$\vec{x} = [-\omega^2 M + i\omega D + K]^{-1}\vec{F} \tag{13}$$

where ω is the frequency in rad/s. The mass dominant behaviour occurs at relatively high frequencies, which means frequencies higher than the natural frequency. In the frequency range where the magnitude bode plots are approximately parallel to one another, referred to as the 'mass line', the masses for the different transfer functions could be determined. For this, Eq. (13) can be rewritten to

$$\vec{x} = [-\omega^2 M]^{-1}\vec{F} \tag{14}$$

where only the mass behaviour is relevant. Determining the mass matrix M, was done by considering the SISO equation to determine the mass m of a single transfer function, as in

$$m = \frac{1}{\omega^2 \cdot 10^{(G/20)}} \tag{15}$$

where $G = \frac{x}{F}$ is the gain in dB. Using this equation, the mass m for all transfer functions could be determined. In the frequency range of 70 - 1000 rad/s, where the transfer functions are approximately parallel, the mass behaviour can be fetched from the plot data of Figure 6. The slopes of the transfer functions are shown to not be constant when the frequency increases. Also, although the transfer function graphs are approximately parallel when regarded at small intervals, their respective distances change slightly with the frequency. These things can be caused by the fourth order model characteristics, where the influence of poles and zeros, and the difference in resonant peak frequencies can cause variations in the slope of the bode plots. This made it hard to determine a specific frequency that is best for the mass matrix determination. Therefore, a sweep was done over the frequencies in this domain, using Eq. (15), with which a set of masses for all individual transfer functions was determined for several frequencies. Since the masses within a set only slightly varied, the mean could be taken of every set of masses to achieve the masses as in

$$\bar{m} = \begin{bmatrix} \bar{m}_{xx} & \bar{m}_{yx} \\ \bar{m}_{xy} & \bar{m}_{yy} \end{bmatrix} = \begin{bmatrix} 2.54 & -3.34 \\ -8.48 & 1.50 \end{bmatrix} kg \tag{16}$$

Since a 180° phase difference is expected between the diagonal and off-diagonal elements, like in the study of Lübbermann[3], the off-diagonal entries in (16) are assumed negative. Although this behaviour is only partially shown in Figure 6, this assumption was made, such that the mass matrix M itself will contain positive values, which is desired. The mass matrix, which represents the MIMO system as a whole, could be achieved by using all determined masses for the respective transfer functions as in

$$M = \begin{bmatrix} \frac{1}{\bar{m}_{xx}} & \frac{1}{\bar{m}_{yx}} \\ \frac{1}{\bar{m}_{xy}} & \frac{1}{\bar{m}_{yy}} \end{bmatrix}^{-1} = \begin{bmatrix} 2.93 & 1.32 \\ 0.519 & 1.73 \end{bmatrix} kg \tag{17}$$

Also, SIMO identification was performed using the same method as for MIMO identification. The corresponding bode plots are shown in Figure 7.

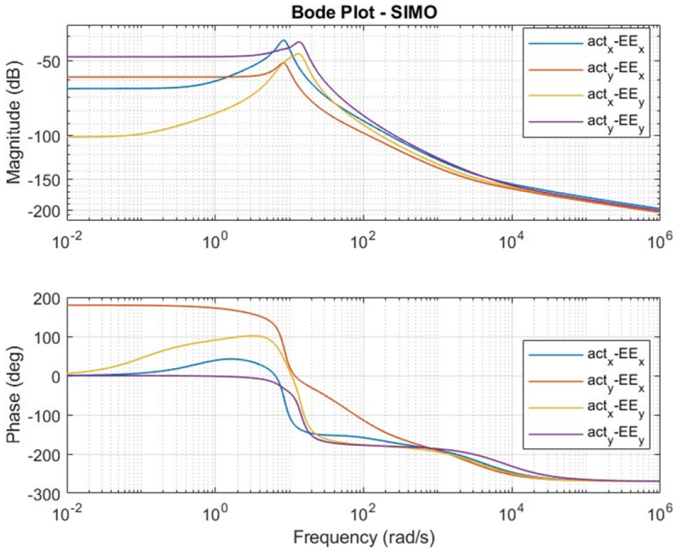


Fig. 7: Bode plots of all I/O combinations; SIMO identification

Again, a sweep was done over the frequencies in the 'mass line' domain, and the mean of every set of masses was taken to achieve the masses shown in

$$\bar{m} = \begin{bmatrix} \bar{m}_{xx} & \bar{m}_{yx} \\ \bar{m}_{xy} & \bar{m}_{yy} \end{bmatrix} = \begin{bmatrix} 1.93 & -6.19 \\ -3.56 & 1.51 \end{bmatrix} kg \tag{18}$$

where the off-diagonal elements are again negative, because of the same reason as described for the MIMO identification. The mass matrix M, which represents the SIMO system as a whole, could be achieved as in

$$M = \begin{bmatrix} \frac{1}{\bar{m}_{xx}} & \frac{1}{\bar{m}_{yx}} \\ \frac{1}{\bar{m}_{xy}} & \frac{1}{\bar{m}_{yy}} \end{bmatrix}^{-1} = \begin{bmatrix} 2.22 & 0.542 \\ 0.942 & 1.74 \end{bmatrix} kg \tag{19}$$

Both the MIMO and SIMO determined mass matrices were tested in simulation. Since the SIMO mass identification gave the most promising results, this mass matrix was implemented in the impedance controller.

3) Damping matrix: The damping matrix D can be determined using the same MIMO and SIMO identification data as for the mass matrix determination. The damping coefficients could be calculated for every transfer function from the respective state space model, using the -3 dB method. The so-called quality factor Q_f could be achieved for all transfer functions, by regarding the width of the peak at -3 dB from the top, according to

$$Q_f = \frac{f_0}{f_2 - f_1} \tag{20}$$

where f_0 is the frequency of the resonant peak and f_1 and f_2 are the frequencies at a magnitude of -3 dB on the lower and higher frequency sides of the peak, respectively. From the quality factor, the damping ratio ζ could be determined, according to

$$\zeta = \frac{1}{2Q_f} \tag{21}$$

where ζ is the damping ratio. The obtained damping ratios for MIMO identification are as in

$$\zeta = \begin{bmatrix} \zeta_{xx} & \zeta_{yx} \\ \zeta_{xy} & \zeta_{yy} \end{bmatrix} = \begin{bmatrix} 0.477 & 0.276 \\ 0.217 & 0.411 \end{bmatrix} \tag{22}$$

The damping coefficient c for all transfer functions was achieved by combining the corresponding damping ratio ζ and critical damping coefficient c_c, according to

$$c = \zeta \cdot c_c = \zeta \cdot 2\sqrt{mk} \tag{23}$$

where m and k are the mass and stiffness of the corresponding transfer function. This resulted in the damping coefficients as in

$$c = \begin{bmatrix} c_{xx} & c_{yx} \\ c_{xy} & c_{yy} \end{bmatrix} = \begin{bmatrix} 18.0 & 4.70 \\ 2.04 & 10.6 \end{bmatrix} \frac{Ns}{m} \tag{24}$$

The achieved damping coefficients can then be used in the damping matrix as in

$$D = \begin{bmatrix} \frac{1}{c_{xx}} & \frac{1}{c_{yx}} \\ \frac{1}{c_{xy}} & \frac{1}{c_{yy}} \end{bmatrix}^{-1} = \begin{bmatrix} -0.949 & 2.14 \\ 4.94 & -0.558 \end{bmatrix} \frac{Ns}{m} \tag{25}$$

for which the same method is used as for the mass matrix determination. The diagonal entries of this damping matrix contain negative values. Physically, this would mean that the damping forces amplify the motion instead of reducing it. This behaviour might arise due to modelling errors, although this is not comprehended fully. However, since similar results have proven to be successful in the study of Lübbermann[3], this result is accepted for now. Also, the off-diagonal damping elements are found to be relatively high, compared to the diagonal elements, which means there is relatively much cross-axis damping compared to direct damping.

To obtain a possibly better result, SIMO identification was performed as well, using the same data as for the SIMO mass identification, according to Figure 7. The same method was used as for the MIMO damping identification, which resulted in the damping ratios as in

$$\zeta = \begin{bmatrix} \zeta_{xx} & \zeta_{yx} \\ \zeta_{xy} & \zeta_{yy} \end{bmatrix} = \begin{bmatrix} 0.161 & 0.155 \\ 0.207 & 0.291 \end{bmatrix} \tag{26}$$

The damping coefficient c for all transfer functions was achieved in the same way as before, according to Eq. (23). This resulted in the damping coefficients as in

$$c = \begin{bmatrix} c_{xx} & c_{yx} \\ c_{xy} & c_{yy} \end{bmatrix} = \begin{bmatrix} 5.30 & 1.70 \\ 2.62 & 7.52 \end{bmatrix} \frac{Ns}{m} \tag{27}$$

The achieved damping coefficients can then be used in the damping matrix as in

$$D = \begin{bmatrix} \frac{1}{c_{xx}} & \frac{1}{c_{yx}} \\ \frac{1}{c_{xy}} & \frac{1}{c_{yy}} \end{bmatrix}^{-1} = \begin{bmatrix} -0.665 & 2.95 \\ 1.91 & -0.943 \end{bmatrix} \frac{Ns}{m} \tag{28}$$

which uses the same method as before. As in the MIMO case, the diagonal entries are found to be negative. However, the off-diagonal damping is not as high as before, due to the significantly lower value of the element corresponding to d_{xy}. Because the manipulator setup is expected to have little damping, the SIMO identified damping matrix was thought to be the best implementation in the controller. However, testing showed more promising results using the MIMO damping identification, which is why the corresponding matrix was used in the controller.

V. Kalman Filter

Sensor fusion is a key part of this research, as both an encoder and an accelerometer are used as sensors on the setup. The accelerometer is introduced to avoid differentiating the position signal twice, which causes noise and error. In the setup for this study, position and acceleration are already known, since these variables are fetched directly from the system. The velocity remains unknown from only the direct measurements and needs to be constructed from the available data in order to implement impedance control. Through sensor fusion techniques, the encoder and accelerometer data can be used to estimate this velocity. The sensor fusion method chosen for this study is a linear Kalman filter[8], which will be used in the controller to make an estimate of the state of the system (position and velocity), using the encoder and accelerometer measurements as input. The noise in the measurements, especially in the accelerometer data, is taken into account in the Kalman filter through the process noise covariance matrix Q and measurement noise covariance matrix R, which are provided to the Kalman filter as well. Since the Kalman filter will be implemented in a discrete environment, the choice was made to design a discrete Kalman filter, using the Kalman filter block from the 'Control System Toolbox' in Simulink.

A. Theoretical Kalman filter

In order to design a discrete linear Kalman filter, a state-space format can be used for the state transition and measurement equations. The transition of the state x_{k-1} to x_k is given by

$$x_k = Fx_{k-1} + Bu_k \tag{29}$$

where F is the state transition matrix, B is the control matrix and u_k is the control input, which corresponds to the acceleration in this study. The state vector x is defined as $x = [p_x \ p_y \ v_x \ v_y]^T$, where p and v correspond to the end-effector position and velocity, respectively. To compute the current state x_k from the previous state x_{k-1}, equations are used for the discrete position and velocity determination, which are

$$p_k = p_{k-1} + Ts \cdot v_{k-1} + \frac{Ts^2}{2} \cdot a_k \tag{30}$$

$$v_k = v_{k-1} + Ts \cdot a_k \tag{31}$$

where Ts is the time step used in the controller and Kalman filter. In these equations, the previous position and velocity are used together with the current acceleration to compute the current position and velocity, which is the current state of the

system. This is done by integrating the velocity and acceleration using the time step Ts. Given that the control input u_k is the acceleration measurement a_k, the state transition matrix F and control matrix B can be defined, based on Eqs. (30) and (31), as in

$$F = \begin{bmatrix} 1 & 0 & Ts & 0 \\ 0 & 1 & 0 & Ts \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{32}$$

$$B = \begin{bmatrix} \frac{Ts^2}{2} & 0 \\ 0 & \frac{Ts^2}{2} \\ Ts & 0 \\ 0 & Ts \end{bmatrix} \tag{33}$$

The state transition equation can be extended using the process noise to give

$$x_k = Fx_{k-1} + Bu_k + w_k \tag{34}$$

where w_k is the process noise, which will be taken into account using the process noise covariance matrix Q of the Kalman filter, which will be elaborated upon.

The measurement equation is given by

$$y_k = Hx_k \tag{35}$$

where H is the measurement matrix. Because the measurement vector y corresponds to the encoder position measurements, it is defined as $y = [p_x \ p_y]^T$. The measurement matrix H can therefore be defined as in

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{36}$$

The measurement equation can be extended using the measurement noise to give

$$y_k = Hx_k + v_k \tag{37}$$

where v_k is the measurement noise, which will be taken into account using the measurement noise covariance matrix R of the Kalman filter, which will be elaborated upon.

1) Prediction and update stages: The Kalman filter is divided into two main parts, which are the prediction stage and the update stage. In these stages the state estimate at time k is computed using the control input vector u and the measurement vector y, which correspond to the acceleration and position data, respectively.

The first stage is the prediction stage, in which a prediction $\hat{x}_k^-$ is made for the state based on the previous state $\hat{x}_{k-1}^+$ and the acceleration data u_k. Also, a prediction P_k^- of the error covariance matrix is computed, which is a measure of the uncertainty in the state estimate, computed using the F and Q matrices and the previous error covariance matrix P_{k-1}^+. The corresponding equations are given by Eqs. (38) and (39). The second stage is the update stage, in which the Kalman gain K is computed based on the error covariance matrix P. With that, the Kalman gain is basically determined through the state transition and measurement matrices F and H and on the process and measurement noise covariance matrices Q and R. The computation of the Kalman gain is shown in Eq. (40). In Eqs. (41) and (42), it is shown how the state x and the error covariance matrix P are updated using the Kalman gain. The initial values used for these parameters are a zero vector and the identity matrix, respectively, as shown in Table II. With this choice for the error covariance matrix, a relatively high uncertainty for the initial state is assumed. This is done because there is little information about what the initial state actually is. The matrix can be adjusted later, during simulation studies.

Prediction stage:

$$\hat{x}_k^- = F\hat{x}_{k-1}^+ + Bu_k \tag{38}$$

$$P_k^- = FP_{k-1}^+F^T + Q \tag{39}$$

Update stage:

$$K_k = P_k^-H^T[HP_k^-H^T + R]^{-1} \tag{40}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k[y_k - H\hat{x}_k^-] \tag{41}$$

$$P_k^+ = [I - K_kH]P_k^- \tag{42}$$

As shown in these equations, a time-invariant Kalman filter approach is used, as the F, B and H matrices do not change with time. Also, the Q and R matrices do not change with time.

2) Kalman gain: The Kalman gain K, which is determined in the update stage of the Kalman filter, is the key part of the Kalman filter, since this serves the purpose of weighting the measurements. This determines how much the measurements are used in updating the state and therefore also how much the predicted state can be trusted. Since the implementation of the Kalman filter is time-invariant, the error covariance matrix P and with that the Kalman gain K will converge over time. Important components that determine the Kalman gain matrix are the noise covariance matrices Q and R.

3) Noise covariance matrices: The process and measurement noise covariance matrices Q and R[9] are of high importance for the behaviour of the Kalman filter, because these matrices influence the Kalman gain K, as shown in the previously provided prediction and update equations, specifically in Eqs. (39), (40) and (42). The Q matrix represents the uncertainty in the transition of the previous to the current state. This includes uncertainty in the system, for example in the parameters, but more dominant is the uncertainty in the accelerometer measurements. The R matrix represents the uncertainty in updating the state using the measurements. This means uncertainty in the encoder measurements. Therefore, the contents of the Q and R matrices (mainly) depend upon

the variance in the measurements of the accelerometer and encoder, respectively. For the implementation of both matrices, the covariances of the noise (off-diagonal elements) are assumed to be zero, which gives two diagonal matrices with the variances on the diagonal. For the initial computation of the Q matrix, it was chosen to make the matrix contain the variances of the accelerometer data in x and y direction. This gives the Q matrix in

$$Q = \begin{bmatrix} \sigma_{a,x}^2 & 0 & 0 & 0 \\ 0 & \sigma_{a,y}^2 & 0 & 0 \\ 0 & 0 & \sigma_{a,x}^2 & 0 \\ 0 & 0 & 0 & \sigma_{a,y}^2 \end{bmatrix} \tag{43}$$

where $\sigma_{a,x}$ and $\sigma_{a,y}$ are the standard deviations in a 'noise measurement' of the accelerometer in x- and y-direction, respectively. This noise measurement was conducted using an activated position controller to keep the system in place, to get the variance in a slightly more realistic process environment. Although this measure has been taken to include more of the process noise in these variances, the real process noise will probably be slightly higher, due to the unmeasured uncertainties, for example in the parameters of the system. Therefore, the Q matrix might need some adjusting through trial-and-error in simulation and experiments to get optimal behaviour of the Kalman filter.

For the computation of the R matrix, the encoder specifications[4] are used in combination with sensor data. The uncertainty in the measurements can be caused by multiple sources, of which quantization error and sensor measurement error are the most dominant. The quantization noise power[10] can be computed according to

$$\sigma_q^2 = \frac{\Delta}{12} \tag{44}$$

where Δ is the resolution of the encoder, equal to $\Delta = 5 \cdot 10^{-8}$ m. The total measurement error could be determined by combining the quantization noise power with the sensor measurement error, described by the variances of the encoder data in x and y direction. The same noise measurement was conducted as for the accelerometer, using the activated controller to have a more realistic measurement environment. This gives the R matrix in

$$R = \begin{bmatrix} R_{e,x} & 0 \\ 0 & R_{e,y} \end{bmatrix} \tag{45}$$

where $R_{e,x} = \sigma_q^2 + \sigma_{p,x}^2$ and $R_{e,y} = \sigma_q^2 + \sigma_{p,y}^2$, where $\sigma_{p,x}$ and $\sigma_{p,y}$ are the standard deviations from the noise measurement of the encoder in x- and y-direction, respectively. The real measurement noise might slightly differ from the computed value. Therefore, the R matrix might need some adjusting through trial-and-error in simulation and experiments to get optimal behaviour of the Kalman filter.

The corresponding numerical values for the parameters mentioned in this section are given in Table II.

Variable	Value	Unit
$\sigma_{a,x}^2$	6.74×10^{-2}	$(\mathrm{m/s^2})^2$
$\sigma_{a,y}^2$	1.015×10^{-1}	$(\mathrm{m/s^2})^2$
$\sigma_{p,x}^2$	1.5827×10^{-12}	m^2
$\sigma_{p,y}^2$	4.395×10^{-13}	m^2
σ_q^2	4.167×10^{-9}	m^2
x_0	$[0\ 0\ 0\ 0]^T$	$\mathrm{m} - \mathrm{m/s}$
P_0	I_4	$\mathrm{m}^2 - (\mathrm{m/s})^2$

TABLE II: Numerical values of mentioned parameters

VI. Simulation Kalman Filter

A. Simulation procedure

Before trying to implement the Kalman filter in the controller in simulation and real-time, simulation studies had to be conducted to test the behaviour of the Kalman filter and to make adjustments and optimizations where necessary. Two methods are proposed for which different inputs are used, namely generated sinusoidal signals and real data from the setup. Several tests were done for both methods, where variations in the frequency of the input signal, the entries of Q and R and the noise characteristics were tested to see the effect on the behaviour of the Kalman filter. The results are discussed, and for the optimal configuration the working of the Kalman filter is presented.

1) Generated sinusoidal signals: Generated sinusoidal signals were used to obtain the position and corresponding acceleration inputs, to which noise was added according to the respective, real world noise power of the represented sensors. Also, the position and the corresponding velocity (without noise addition) could be used together as 'true' state, which is the desired position and velocity that the Kalman filter should obtain. Testing was done by comparing the estimated state to this 'true' state. Several tests are conducted, where various sinusoidal waveforms are added to the signals together with noise, in order to obtain multiple realistic conditions for the filter to test its behaviour. A representative example of these used multi-sine inputs is shown in Figures 16 and 17 in Appendix B.

2) Real data input: As a final test, real position and acceleration data from the setup are collected and used to see the influence of the optimized Kalman filter on representative data, as in real-time implementation. The downside of using real data is the absence of a desired or 'true' state to compare the state estimation to. However, the state resulting from the measurements of the system is compared to the estimated state from the Kalman filter. This has provided understanding about the influence of different configurations regarding the Kalman filter settings and the simulation model parameters on the estimate of the state. Especially important is the velocity estimation, as the velocity is not measured directly using sensors. Additional adjustments were made to the Kalman filter and the model to enhance the state estimation further.

B. Results - generated sinusoidal signals

For the sinusoidal input method, the true and estimated position signals showed no significant difference, indicating a good position estimate. Especially important is the effect of several tests on the observed error between the true and estimated velocities, which is explained below. The optimized velocity error is shown in Figure 8. In Figures 18 and 19 in Appendix B, the corresponding true and estimated velocity plots are shown for a multi-sine and a single sine input. Both cases result in the same velocity error plot.

1) Q and R matrices: It was observed that changing the Q and R matrices did not make a visible difference, unless the Q matrix was chosen significantly lower than the R matrix, starting at an order of 10^2 lower. This made the position and velocity estimates significantly worse, because the noisy accelerometer measurements were taken into account too much in the estimation process. Changing the R matrix slightly also did not yield better results, and because R was determined quite accurately, it was kept constant for the rest of the study.

Even when the Q and R matrices do not seem to make a difference, it was observed that the Kalman gain K and the error covariance matrix P do indeed converge to different values depending on Q and R, like expected from the theory. In Figures 21 - 23 in Appendix B, the Kalman gain matrix convergence over time is showcased for different combinations of Q and R matrices. This showcases the expected behaviour of the Kalman gain quite well. However, in this study, the Q matrix is way higher than the R matrix (with a factor of about 10^7), which means a slight change in the respective difference does not yield a significant change in the Kalman gain.

2) Input frequency: Changing the input frequency does not have effect on the velocity estimation. The velocity error is consistent, even for larger frequencies up to and including 100 Hz, which is considered to be very high for a position signal in this study. Note that this does not mean that the Kalman filter passes through noise up to this frequency; the Kalman filter removes white noise while at the same time being able to handle high frequent sinusoidal signals.

3) Low-pass filter: To reduce the noise and deviation in the resulting velocity error further, a second order low-pass filter was added to the Kalman filter acceleration input. However, the filter does not yield better results when using sinusoidal input signals, as the acceleration signal itself gets slightly attenuated as well, which results in a larger velocity error. Also, since the Kalman filter is known to handle white noise very well, the filter should not benefit the result and is not an improvement when using generated sinusoidal input signals with added noise.

4) White noise: When the generated noise in the acceleration signal is increased, the velocity error increases as well, although the shape of the graph does not change. However, within the expected error margin of the accelerometer, the velocity error does not significantly change. To illustrate, increasing the accelerometer noise power with a factor of 10 results in an increase in the velocity error with a factor of 3. However, the accelerometer noise is determined quite accurately and is dominant in the process noise. Therefore, the possibly unmodeled uncertainty, which will not cause such an increase in process noise, is not considered to cause a problem for the implementation of the Kalman filter.

C. Results - real data input

When having real data as input, the downside is that there is no true state to compare to. However, the derivative of the encoder position was used to compare with. This is not ideal for validation, as the Kalman filter velocity estimation is desired to improve upon the position derivative in order to come closer to the true state. However, for this simulation test, it is found to be a decent alternative to the true state. The benefit of having real data as input, is that the Kalman filter is subjected to representative data; measurements that it needs to handle in the real-time controller as well. The measured position and estimated position signals show no significant difference, namely an error in the order of about 10^{-3}. The observed error between the position derivative and estimated velocities was quite substantial, which was not expected. In Figure 9, this velocity error is shown. To obtain this readable plot, the velocity estimation was filtered to some degree to remove most of the noise, as the raw velocity data from the Kalman filter contained a substantial amount of noise. An illustration of this noise is shown in Figure 20 in Appendix B.

So, this error had to be reduced, which was done by adding a second-order low-pass filter to the Kalman filter acceleration input. This is done because too much noise in the acceleration signal is found to cause bad results for the Kalman filter, even though it is known for being able to handle white noise. The latter was also found when using generated sinusoidal input signals. Where a low-pass filter did not show promising results when using these generated input signals, it does with real data as input. Using a cut-off frequency of 6 Hz, the noise in the acceleration was reduced significantly, which made it possible to decrease the entries in the Q matrix by a factor of about 10^{-4}. Based on the measured variances in this accelerometer signal, this exact resulting Q_{filt} matrix was constructed and is shown in

$$Q_{filt} = 10^{-4} \cdot \begin{bmatrix} 0.357 & 0 & 0 & 0 \\ 0 & 0.0596 & 0 & 0 \\ 0 & 0 & 0.357 & 0 \\ 0 & 0 & 0 & 0.0596 \end{bmatrix} \tag{46}$$

The low-pass filter together with the updated Q matrix resulted in an improved velocity estimation. In Figure 10, this opti-

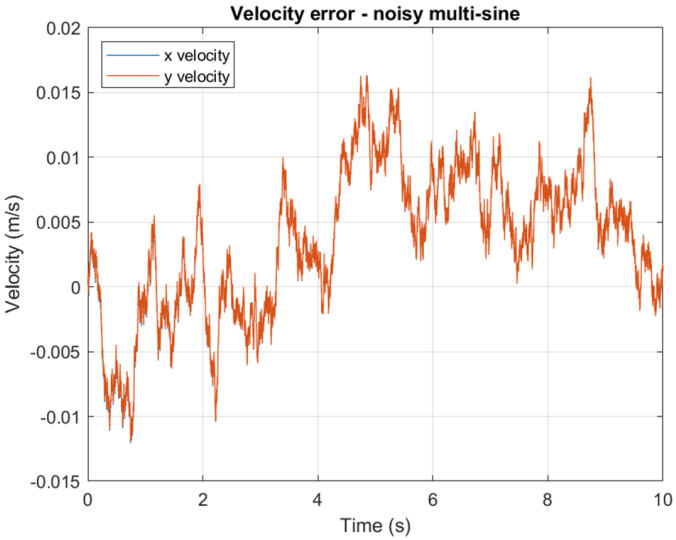


Fig. 8: Velocity error, using input multi-sine/sine inc. noise

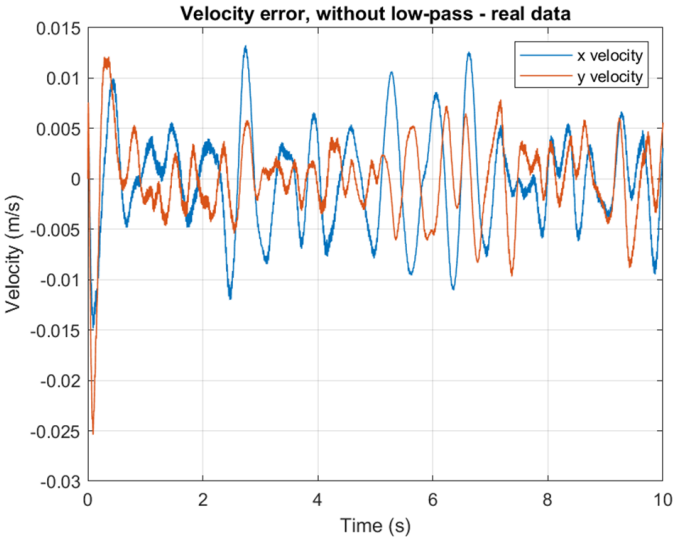


Fig. 9: Initial velocity error using real data, without low-pass filter

mized velocity error is shown. The low-pass filter is therefore shown to not only improve the velocity estimation by reducing the deviation from the equilibrium point, but also by reducing the noise level, since no additional filtering of the velocity estimation was needed this time. It should be noted that the accelerometer signal might be attenuated slightly using the mentioned cut-off frequency. However, even with this, the Kalman filter is shown to benefit from the achieved noise reduction.

VII. Simulation Controller

A. Simulation procedure

Before trying to implement the controller in real-time, simulation studies had to be conducted to test the controller behaviour using the Kalman filter and to make adjustments

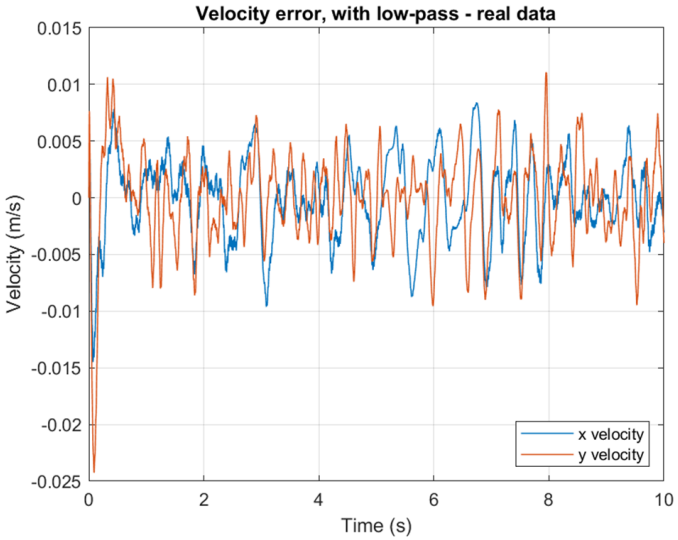


Fig. 10: Optimized velocity error using real data, with low-pass filter

and optimizations where necessary. The closed-loop control scheme of Figure 2 was implemented in a simulation model in Simulink, where the plant could be represented by the fourth order MIMO state space model obtained for system identification, using the System Identification Toolbox. The encoder and accelerometer measurements were constructed as close to reality as possible, by using the noise power that was obtained from noise measurements for both sensors, like in Section VI *Simulation Kalman Filter*. The virtual and system impedances are represented by the compensator part as in Figure 4, according to Eqs. (1) and (2).

Important to note is that the main focus for this part of the simulation study is on the controller behaviour as a whole, which is mainly focused on the (virtual) impedance behaviour. To simulate this, the plant in the closed-loop model was fed with a force step input from 10 N to 0 N on the x-actuator, in order to simulate the position response and make improvements and changes to the model accordingly. The natural frequency and damping ratio of the position response were studied to get an idea of the ability of the controller to mimic a certain impedance and of the general working of the controller. Standard values for the virtual impedance matrices were chosen as: $m_v = 3$ kg, $\zeta_v = 0.2$ and $k_v = 150$ N/m. These respective virtual mass, virtual damping ratio and virtual stiffness parameters are used according to Eqs. (3) - (5) to create the virtual impedance matrices.

B. Results - step input

When applying a force step input to the plant, it was observed that the natural frequency resembled the desired natural frequency quite well. However, the damping was found to be too high. While the virtual damping ratio was set to $\zeta_v = 0.2$, the observed damping ratio was around $\zeta = 0.75$. Several changes to the model were considered and tried, which is

elaborated upon.

1) Low-pass filter: One of the reasons for the high damping ratio was found to be the low-pass filter in front of the acceleration input of the Kalman filter. The usage of this low-pass filter showed promising results for the Kalman filter simulation. However, while it still improves the velocity estimation itself, the low-pass filter does not yield better results in closed-loop implementation, regarding the overall behaviour of the controller. This is because the low-pass filter removes higher frequency components, corresponding to fast changes in the system response, which can effectively yield higher damping in the system. Removing the low-pass filter slightly improves the damping ratio, to around $\zeta = 0.68$. Although the damping ratio is still too high, the behaviour of the controller was visually improved, which is shown in the comparison in Figure 11. The plot is zoomed in, which clarifies the illustration of the difference in overshoot of the peaks. The initial value of these step responses is $p = 0.0355$ m.

2) Plant model: Through a debugging process, the remaining gap between the expected and simulated damping ratio was thought to be caused by the used plant model, which is the state space model from the identification. This fourth order, continuous, MIMO state space model was replaced with multiple other models to find a more representative plant model to eliminate the source of the faulty damping behaviour. The below options were implemented and tested.

- Second, third and fifth order models
- Discrete fourth order model
- SISO fourth order model
- Changed preprocessing of the identification data
- Replaced identification data
- Usage of additional accelerometer identification data

All of the above alternatives were found to be either unstable, not providing a good enough fit or not improving the damping behaviour. Therefore, the original state space model was used.

3) System matrices: The mass, damping and stiffness matrices used in the compensator were adjusted and replaced to test their effect on the damping behaviour. Changing the SIMO mass matrix and MIMO damping matrix to their MIMO and SIMO alternatives or tweaking the matrices slightly did not result in significant change in the damping behaviour. Therefore, it was chosen to keep the SIMO mass and MIMO damping matrices implemented in the simulation, which provided the best natural frequency and damping behaviour. For the stiffness matrix implementation, the used lookup table was also found to be optimal, compared to using the 2×2 stiffness matrix from (12).

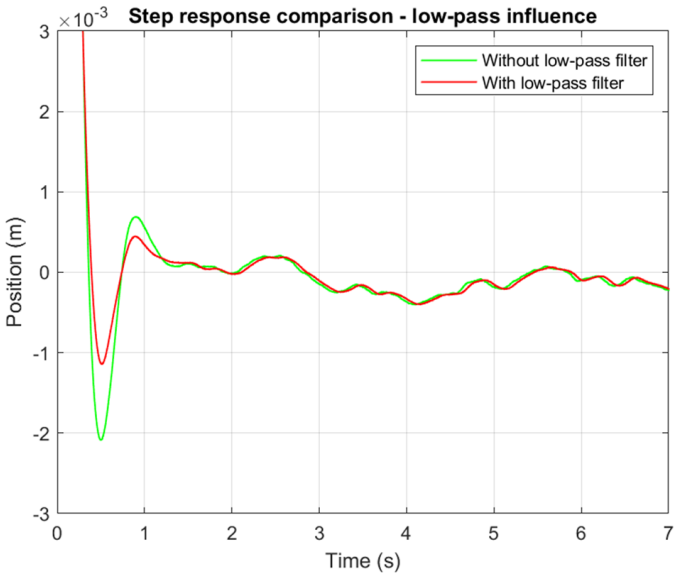


Fig. 11: Step response position comparison, showcasing the influence of the low-pass filter on the damping

The resulting controller simulation still did not show the correct damping, however, the behaviour of the damping did agree with the expectations. The latter is shown in Section VIII *Validation*.

VIII. Validation

A. Validation procedure

For validation of the controller, the natural frequency and damping ratio of the system, set to mimic a certain virtual system, were achieved by analysing the step response. A reference system representing the desired virtual impedance is used as comparison for the controller. Plotting the expected and obtained natural frequency and damping ratio for varying values of the virtual mass, damping and stiffness in three respective tests, gave insight in the ability of the controller to mimic a certain impedance. The step response was achieved by using an initial force on the x-actuator, which changes from 10 N to 0 N at a specified time. Ideally, the initial force value would be around 2 N, but because this gives uncertain results, the initial force was chosen higher. Using forces above around 5 N gave more readable results, so a force of 10 N was chosen for the validation measurements to ensure readable results over the whole virtual mass, damping and stiffness intervals. Also, it was chosen to do the validation of the controller in a simulation environment, instead of in real-time, because of the difficulty with reading the measurements, as described above. More about these considerations can be found in Section IX *Discussion*.

The natural frequency ω_n and the damping ratio ζ can be expressed in terms of the mass, damping and stiffness parameters. This makes analysing the natural frequency and damping ratio a good validation method for impedance

control. The natural frequency and damping ratio are described as

$$\omega_n = \sqrt{\frac{k}{m}} \tag{47}$$

$$\zeta = \frac{c}{c_c} = \frac{c}{2\sqrt{mk}} \tag{48}$$

where c is the damping coefficient and c_c is the critical damping coefficient. The implementation of the virtual damping as shown in Eqs. (4) and (6), corresponds to Eq. (48) above. Important to note is that this method of implementing the virtual damping does not only rely on the virtual damping ratio ζ_v, but also on the virtual mass m_v and virtual stiffness k_v. So, although a specific ζ_v is desired and used in computing the corresponding damping, this damping implementation will not achieve the exact desired damping ratio for all values of the mass and stiffness, which is also shown in the validation plots.

To achieve the resulting natural frequency ω_n and damping ratio ζ parameters from the simulation, the step response data of the system was analysed. The natural frequency could be determined from the time difference between the first two peaks, according to

$$\omega_n = \frac{2\pi}{T} \tag{49}$$

where T is the period, measured between the two peaks. The damping ratio could be determined from the percentage overshoot (PO) of the response of the system, according to

$$\zeta = \frac{|\ln(\frac{PO}{100\%})|}{\sqrt{\pi^2 + \ln^2(\frac{PO}{100\%})}} \tag{50}$$

where the percentage overshoot PO was determined for the step response, according to

$$PO = \frac{p_{steady} - p_{peak}}{p_{init} - p_{steady}} \cdot 100\% \tag{51}$$

where p_{steady} is the steady state position after converging, p_{peak} is the position of the first peak and p_{init} is the initial position. Note that the step response from 10 N to 0 N results in a descending position response, such that the signs in the equation had to be inverted. Since the damping ratio remains between 0 and 1, meaning the system remains underdamped, the above method of calculating the damping ratio remained valid throughout the entire virtual mass, stiffness and damping ratio intervals.

For all validation tests it holds that the achieved damping ratio is higher than expected. However, the damping behaviour itself can be said to be according to expectations, which is explained in the corresponding sections that follow.

B. Results - virtual mass validation

The virtual mass validation is considered to be the most important method to validate the acceleration influence, since the mass of a system is directly linked to its acceleration through Newton's second law. The corresponding results are shown in Figure 12.

Shown is that the simulated natural frequency is quite close to the expected natural frequency, except for very low masses. Shown is that for a virtual mass of $m_v = 2$ kg, a good result is achieved, and that for $m_v = 1.5$ kg, the resulting frequency is slightly off compared to the expected value. For even lower virtual masses, the system becomes unstable, which is already indicated by Eq. (47) and the mass validation plot. Also, a slight deviation from the expectation can be seen at the $m_v = 10$ kg and $m_v = 15$ kg. Because the other parts of the graph fit quite well, the slightly higher results at these two masses are expected to be due to interaction with the parameters of the system. Another explanation could be errors in the simulation process.

The achieved damping ratio is shown to have an additional offset of around $\zeta = 0.3$, as was discussed in Section VII *Simulation Controller*. However, with increasing virtual mass, the desired constant damping ratio behaviour is slightly visible in the convergence of the achieved damping ratio to around $\zeta = 0.5$. When disregarding the offset, the behaviour can still be examined. For lower virtual mass, a deviation from the desired damping ratio is observed, which is caused by the implementation of the damping discussed in Section VIII-A *Validation procedure*. The mass is related to the damping ratio, according to Eq. (48). Also, the mass of the system and virtual system being similar might cause resonance, because the natural frequency will be similar according to Eq. (47). This resonance requires larger damping, resulting in a peak in the damping ratio. For higher virtual mass, the damping ratio converges, so it can be compared better with the desired damping ratio.

So, the system accurately mimics the desired natural frequency over almost the whole virtual mass interval, in the studied region of $m_v \geq 2$ kg. The damping is mimicked best for virtual masses in the studied region of $m_v > 10$ kg, disregarding the offset.

C. Results - virtual stiffness validation

The virtual stiffness validation is considered to be the most important method to validate the position influence, since the stiffness of a system is directly linked to its position through the equation of motion. The corresponding results are shown in Figure 13.

Shown is that the simulated natural frequency behaves according to a partially different relationship than the expected square root relationship. For lower virtual stiffness, the frequency does not start as low as expected, namely at

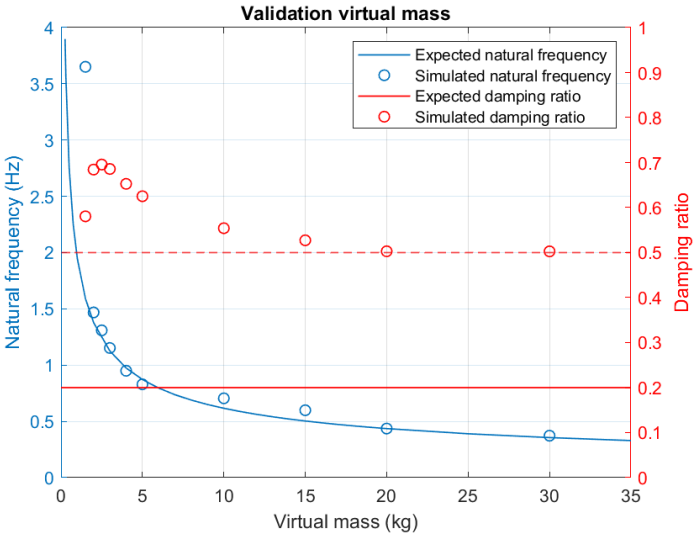


Fig. 12: Validation virtual mass; natural frequency and damping ratio against changing virtual mass matrix

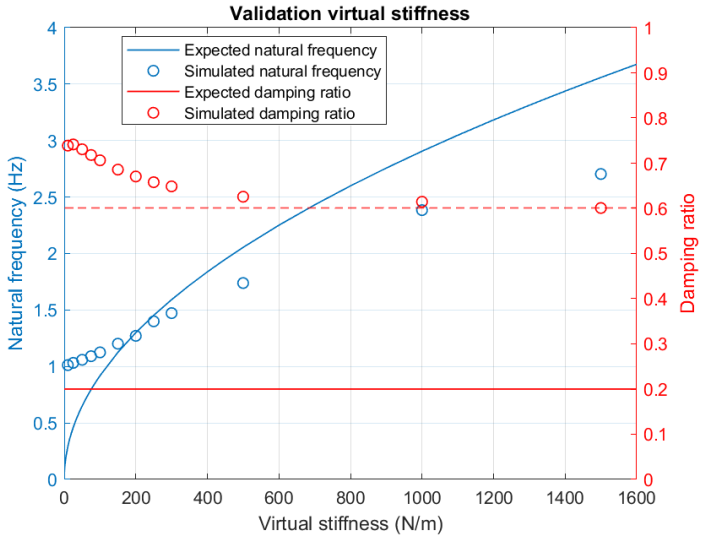


Fig. 13: Validation virtual stiffness; natural frequency and damping ratio against changing virtual stiffness matrix

$\omega_n = 1$ Hz instead of at $\omega_n = 0$ Hz. Also, where the slope is expected to start steep and decrease with increasing stiffness, the slope is less steep and approximately constant, increasing a bit. For stiffness values above $k_v = 200$ N/m, where the simulation results cross the expected results, the slope is still approximately constant, decreasing a bit. Over the whole stiffness interval, the magnitude of the slope is less than the reference slope. From Eq. (47) can be concluded that this could be caused by the mass value, which determines the slope. Alternatively, the difference might arise from the dynamic effects in the controller that are not captured in the reference model.

The achieved damping ratio is shown to have an additional offset of around $\zeta = 0.4$. However, with increasing virtual stiffness, the desired constant damping ratio behaviour is visible in the convergence of the achieved damping ratio to around $\zeta = 0.6$. When disregarding the offset, the behaviour can still be examined. Again, as for the mass validation, the observed deviation from the desired damping ratio, shown for lower stiffness values, is caused by the implementation of the damping discussed in Section VIII-A *Validation procedure*. The stiffness is related to the damping ratio, according to Eq. (48). Also, the stiffness of the system and virtual system being similar might cause resonance, because the natural frequency will be similar according to Eq. (47). This requires larger damping, resulting in a larger damping ratio. For higher virtual stiffness, the damping ratio converges, so it can be compared better with the desired damping ratio.

So, the system mimics the desired natural frequency best in the region of 100 N/m $\leq k_v \leq$ 300 N/m. The damping is mimicked best for virtual stiffness in the studied region of $k_v >$ 300 N/m, disregarding the offset.

D. Results - virtual damping validation

The virtual damping validation is considered to be the most important method to validate the velocity influence, since the damping of a system is directly linked to its velocity through the equation of motion. The corresponding results are shown in Figure 14. Note that for the damping validation, only the damping ratio is considered.

The achieved damping ratio is shown to have an additional offset of around $\zeta = 0.45$. However, with an increasing virtual damping ratio, the desired linear damping ratio behaviour for $0 \leq \zeta_v < 1$ is visible when comparing with the shifted expectation. Since the mass and stiffness are both constant in this test, linear behaviour without deviations can be expected, according to Eq. (48). This is indeed observed, disregarding the small errors, which are caused by the data fetching process. However, the magnitude of the slope is slightly smaller than expected. From Eq. (48) can be concluded that this could be caused by the mass and stiffness values, which determine the slope. Also, the damping ratio is expected to saturate at $\zeta = 1$, as shown in the reference plot. This saturation happens already at around $\zeta = 0.91$, which can be caused by various things, such as dynamic effects in the controller that are not captured in the reference model.

So, the system mimics the desired damping best in the linear region of $0.1 \leq \zeta_v \leq 0.5$, disregarding the offset.

IX. Discussion

The validation of the controller was conducted using a force step with an initial force of 10 N, where an initial force of 2 N keeps the end-effector within the desired space, corresponding to the desired precision of the end-effector movement. This is done because the velocity estimation deviates too much for accurate measurements using a force of 2 N. Because the

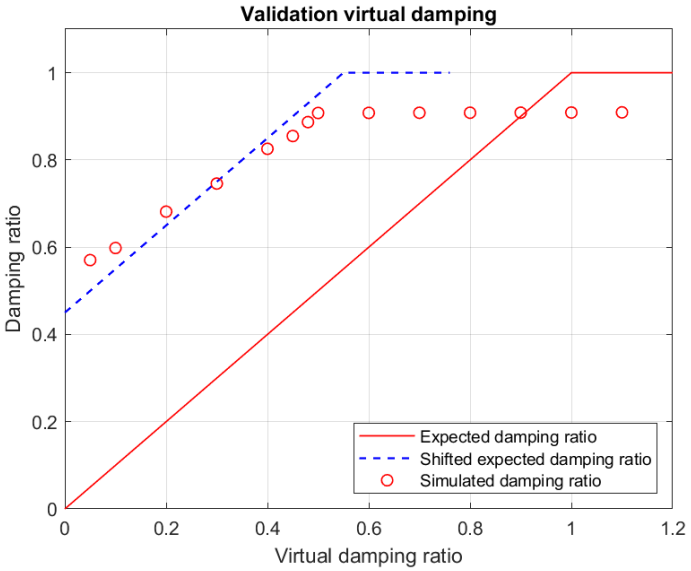


Fig. 14: Validation virtual damping; damping ratio against changing virtual damping matrix

natural frequency and damping ratio should not change with such an increase in initial step force, it could be said that the same natural frequency and damping ratio that are achieved in validation would have been achieved when using a force of 2 N. However, even if the natural frequency and damping ratio would be the same, this can not be validated, because of the uncertain and fluctuating position response. Regardless, the position fluctuation caused by the fluctuation in the velocity estimation is too large for the impedance control to be effective for high-precision movement. Where using 10 N is a perfectly available option to perform validation on a scale where the effect of the uncertainty is reduced, this can only be partially used to comment on the results for the desired precision, as explained above. The controller in real-time did show partially working impedance tracking behaviour, but because of these difficulties and debatable simulation results, the controller could not be used for validation measurements in real-time.

In the validation plots, the simulated damping ratio is shown to have an offset compared to the expected result, although the damping behaviour corresponds to the expectations quite well. The reason behind this offset is investigated, of which the most important considerations and results are discussed in Section VII *Simulation Controller*. This problem was not solved in this study. In further research, this problem has to be either solved or avoided at all through other implementation methods.

The validation is only conducted for the x-actuator. This choice was made because the x-direction proved to give the most consistent movement from previous research[5] and in conducted real-time measurements. The latter is relevant, as the state space model, which is obtained from measurements on the setup, is used in simulation. Additionally, since Lübbermann also performed validation for the x-direction[3], this made it possible to compare the results.

When comparing to the results of the study of Lübbermann[3], where the position derivative is used for velocity computation, similar behaviour is shown for the obtained natural frequency. However, it is found that the method of Lübbermann results in slightly better natural frequency behaviour than the method of the current study, especially for lower virtual mass and stiffness values. Also, their results show a better damping ratio behaviour, because there is no offset in the result and the obtained values are closer to the desired values, regarding the mass and stiffness validation tests. In the damping validation test, the current study shows a better matching slope for lower virtual damping ratio values. However, the overall results of the implementation method of Lübbermann are found to be more desirable compared to the results of the current study.

X. Conclusion

This study investigated the application of a linear Kalman filter as sensor fusion and state estimation method for improved impedance control on a 2-DOF compliant manipulator setup, using two motor encoders and an accelerometer. While validation of the influence of the virtual system parameters to the resulting behaviour of the system did partially result in expected behaviour, the results are not considered good enough to state that the accelerometer implementation is an improvement on the previous control scheme introduced by Lübbermann[3]. This conclusion is also based on the fact that the validation could not be performed for the desired precision in terms of end-effector movement, as discussed in Section IX *Discussion*. The inability to provide accurate results with deviations of the end-effector being within a limit of 0.015 m from the equilibrium position, rejects the current impedance control implementation proposal. The substantial noise in the accelerometer measurements in combination with the current implementation of the Kalman filter provides a velocity estimation that deviates too much from the true velocity. Therefore, for the desired precision movement the controller should provide, it is better to use the velocity obtained from the encoder position derivative, like in the implementation of Lübbermann. This does not mean that the accelerometer and Kalman filter can not be used for the implementation in the impedance control scheme, but rather that the current implementation is not an improvement. In Section XI *Recommendations*, suggestions can be found for possible improvements and alternatives to continue this research.

XI. Recommendations

For further research into the application of an accelerometer and Kalman filter in the impedance control scheme, there are some important thoughts and opportunities to consider. Firstly, the Kalman filter implementation can be redesigned

by replacing the linear Kalman filter with an extended Kalman filter (EKF), which is an alternative to the linear Kalman filter that handles nonlinear systems better. Since the manipulator setup is a nonlinear system, an EKF can provide a more optimal state estimation, especially for small nonlinearities. Furthermore, alternative sensor fusion methods could be investigated, such as a H-infinity filter, which minimizes the worst-case estimation error. This filter might provide better results, because of its ability to handle uncertainties in the system or measurements. Also, an alternative sensor can be used for a better state estimation, such as a gyroscope or a less noisy accelerometer. The advantage of the gyroscope is the direct (angular) velocity measurement, which could be a good alternative for the accelerometer. The encoders and accelerometer could even be used together with a gyroscope for direct measurements of position, velocity and acceleration, which could improve the feedback loop and impedance mimicking ability.

XII. Acknowledgements

I would like to express my appreciation to dr. ir. R.G.K.M Aarts and dr. J. Dasdemir for the supervision and help during this study. Also, I would like to thank all my fellow students who worked on the setup at the time for the pleasant cooperation.

References

[1] L. Gu and Q. Huang, "Adaptive impedance control for force tracking in manipulators based on fractional-order pid," *Applied Sciences*, 2023.

[2] A. Dietrich, *Whole-Body Impedance Control of Wheeled Humanoid Robots*. Springer Tracts in Advanced Robotics (STAR, volume 116), 2016, pp. 13–22.

[3] T. Lübbermann, "Impedance control for a compliant manipulator," *University of Twente, Bachelor Assignment*, 2019.

[4] K. Folkersma, "Design and evaluation of a 2-dof large stroke stage with elastic elements," *Master thesis, University of Twente*, 2011.

[5] J. Uitbeijerse, "Impedance control of a 2-dof compliant manipulator," *University of Twente, Faculty of Engineering Technology, Bachelor Assignment*, 2021.

[6] M. Hakvoort, "2-dof force control with a compliant manipulator," *University of Twente, Faculty of Engineering Technology*, 2020.

[7] T. Kluin, "Sensorless dynamic force control on a 2-dof compliant manipulator," *University of Twente, Faculty of Engineering Technology, Bachelor Assignment*, 2021.

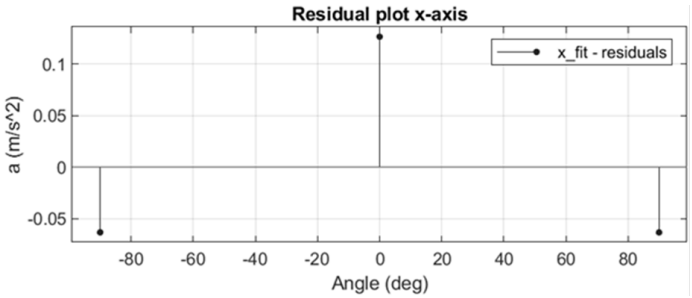
[8] G. Welch and G. Bishop, "An introduction to the Kalman filter," *University of North Carolina at Chapel Hill*, 2006.

[9] A. A. Madhloom, F. A. Raheem, and A. R. Kareem, "A modified Kalman filter-based mobile robot position measurement using an accelerometer and wheels encoder," *Engineering and Technology Journal 40 (01) 267-274*, 2022.

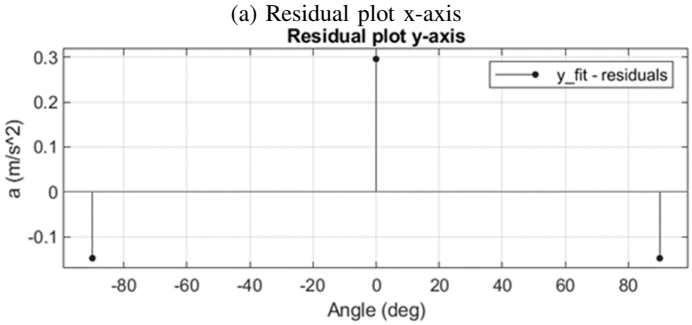
[10] T. van der Holst, "External force estimation on a nonlinear compliant 2dof manipulator system," *University of Twente, Master Thesis*, 2022.

APPENDIX A
ACCELEROMETER CALIBRATION

When accelerometer calibration was conducted, residual plots could be used in order to test the validity of the accelerometer measurements. This was achieved by measuring the gravitational acceleration in positive and negative direction and for a stationary position, for both axes. The Curve Fitting Toolbox in Matlab was used to obtain the corresponding residual plots, shown in Figure 15.



(a) Residual plot x-axis



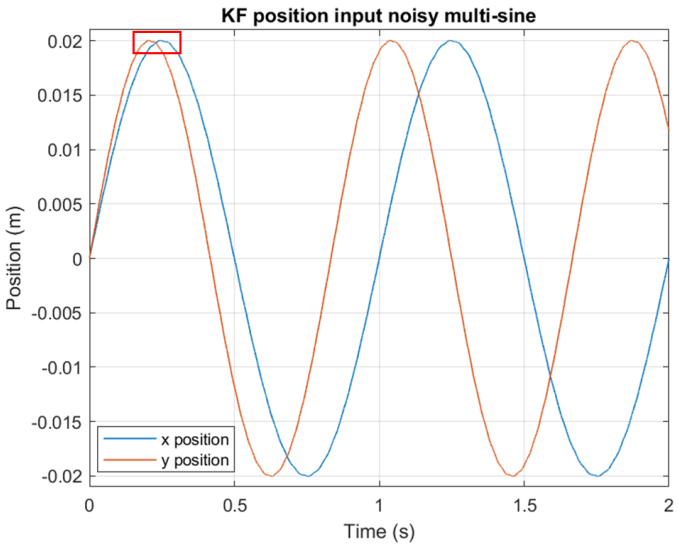
(b) Residual plot y-axis

Fig. 15: Residual plots of linear regression for x and y, respectively; accelerometer calibration

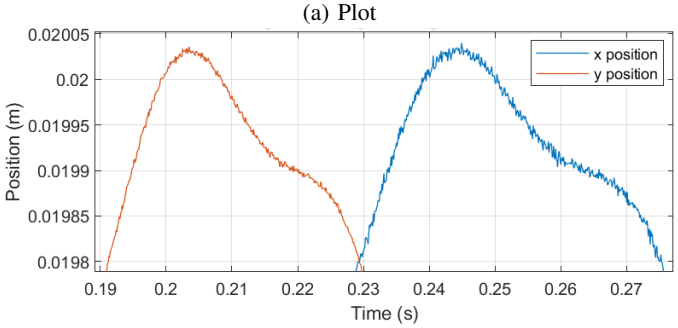
APPENDIX B
SIMULATION KALMAN FILTER

A. Generated sinusoidal signals

1) Inputs: To illustrate the simulation environment of the Kalman filter simulation, additional plots are provided regarding the input and output of the Kalman filter when using generated sinusoidal signals. The given input position, including multi-sinusoidal behaviour and suiting noise according to the encoder noise power, is shown in Figure 16. The matching input acceleration signal, including multi-sinusoidal behaviour and suiting noise according to the accelerometer noise power, is shown in Figure 17.



(a) Plot



(b) Zoom

Fig. 16: Kalman filter position input signal, multi-sine inc. noise

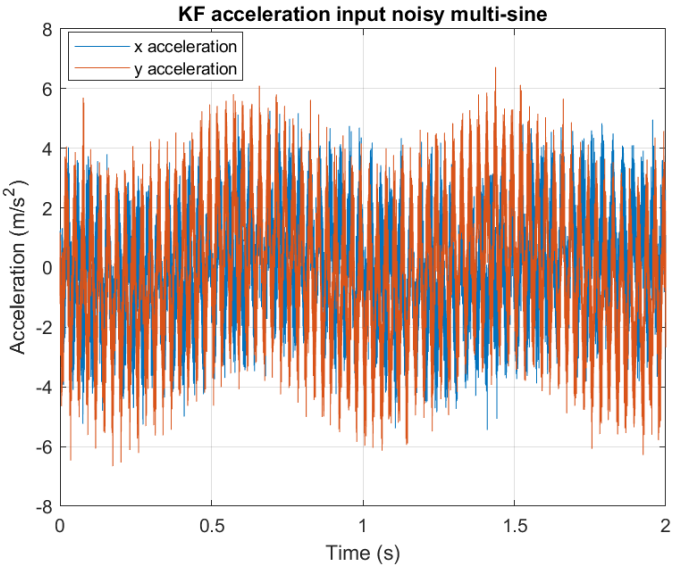
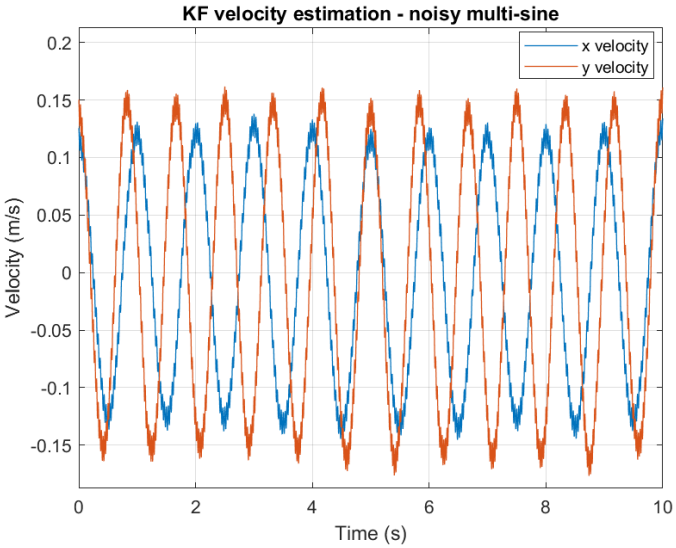
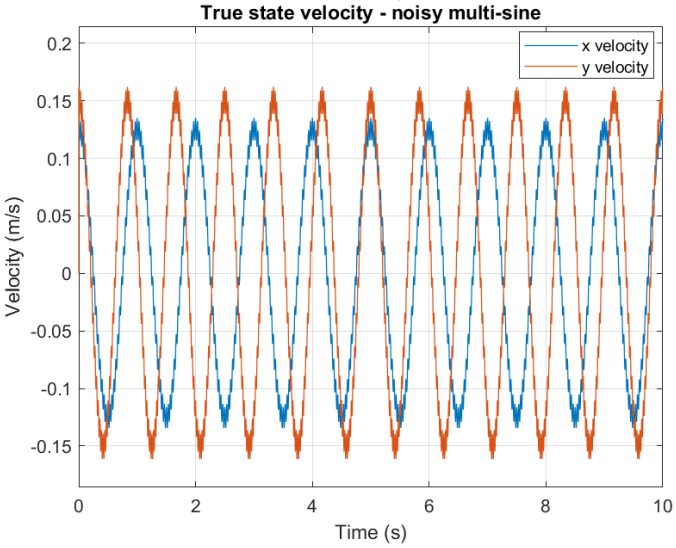


Fig. 17: Kalman filter acceleration input signal, multi-sine inc. noise

2) Outputs: The resulting true state and estimated velocities for a multi-sine input are shown in Figure 18. The corresponding velocity error is shown in Figure 8.



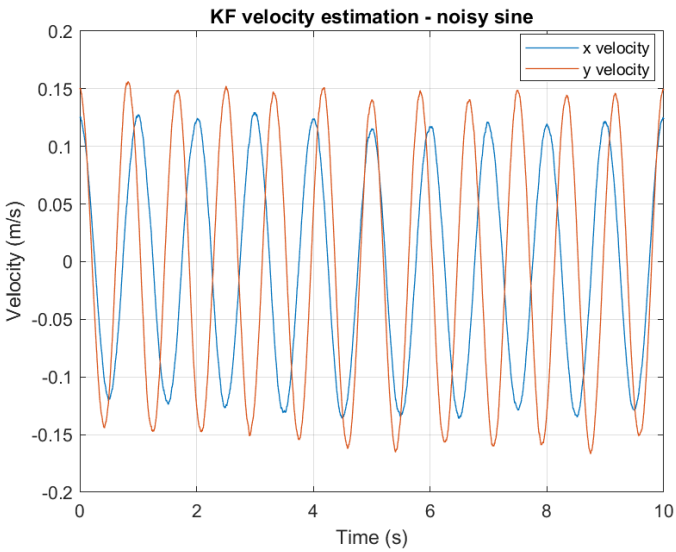
(a) Kalman filter velocity estimation



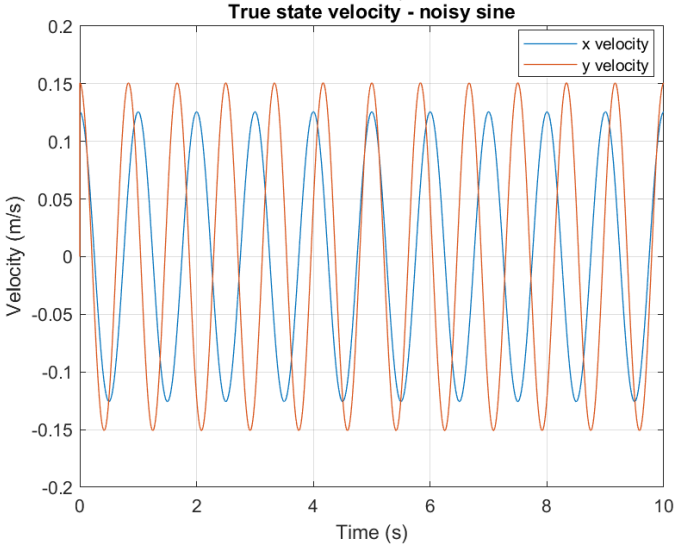
(b) True state velocity

Fig. 18: Velocity true state and corresponding Kalman filter estimation, using input multi-sine inc. noise

The resulting true state and estimated velocities for a single sine input are shown in Figure 19. The corresponding velocity error is shown in Figure 8.



(a) Kalman filter velocity estimation



(b) True state velocity

Fig. 19: Velocity true state and corresponding Kalman filter estimation, using input sine inc. noise

B. Real data input

Using real data as input to the Kalman filter, without a low-pass filter for the acceleration data, the velocity estimation contained a substantial amount of noise. The readable plot in Figure 9 was obtained by slightly filtering the signal to reduce this noise level. However, to give an illustration of the real (raw) output velocity data of the Kalman filter in this situation and the corresponding noise level, the raw velocity estimation signal is shown in Figure 20. This indicates the true difference between using and not using an additional low-pass filter.

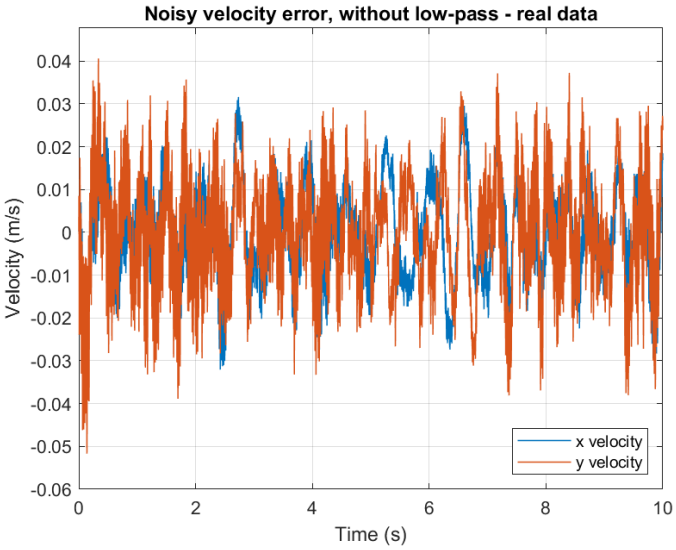


Fig. 20: Initial velocity error using real data, without low-pass filter, unfiltered Kalman filter output

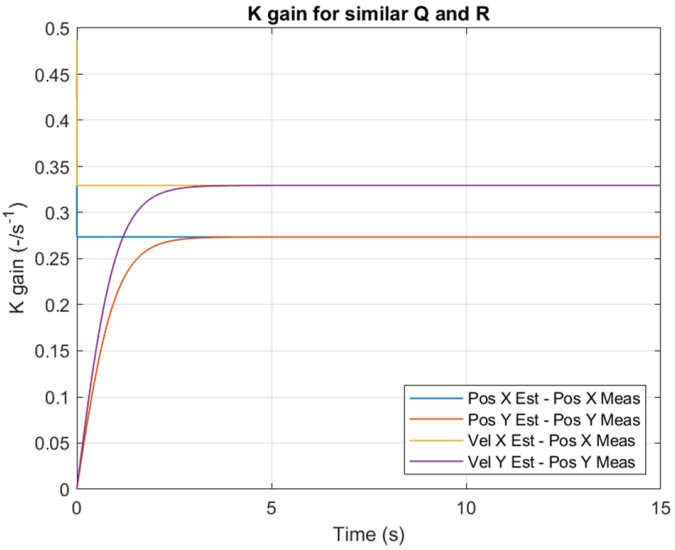


Fig. 22: Kalman gain matrix K entries convergence over time for similar Q matrix entries compared to R

C. *Kalman gain*

The convergence behaviour of the Kalman gain is illustrated for several relative choices for the Q and R matrices. The corresponding plots are shown in Figures 21 - 23. Shown is that that the overall behaviour is different for different noise covariance matrices. The most important difference is in the convergence to different values. For a relatively high Q matrix, the Kalman gain matrix elements are shown to converge to the expected limit of 1 $(-/\mathrm{s}^{-1})$.

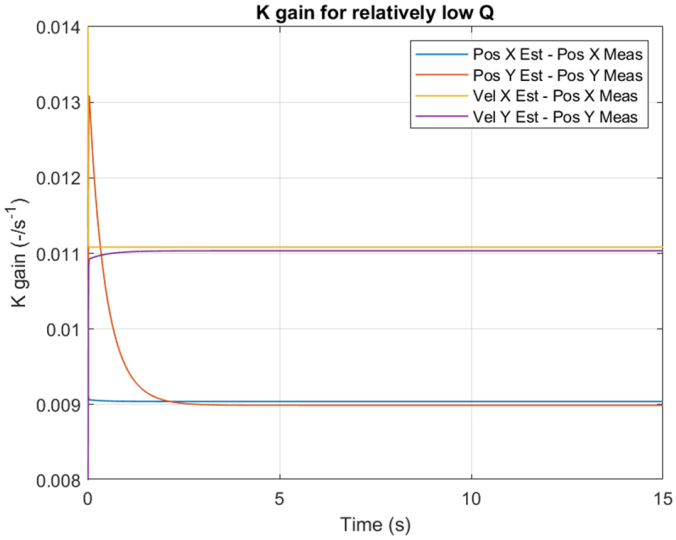


Fig. 21: Kalman gain matrix K entries convergence over time for relatively low Q matrix entries (difference with R of about 10^2)

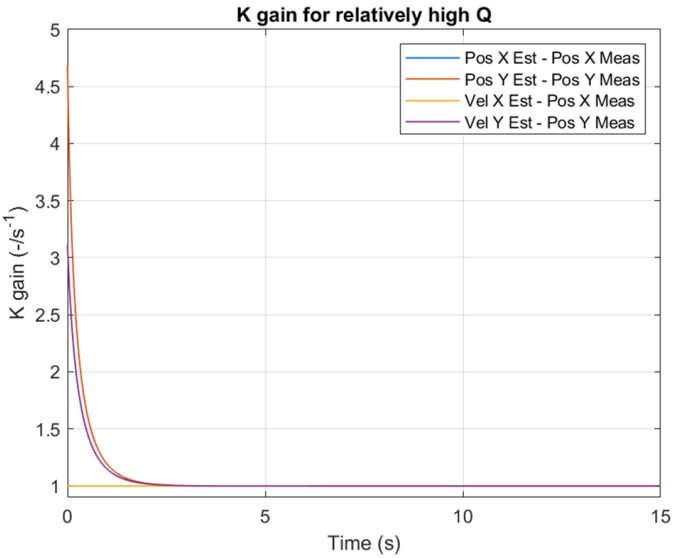


Fig. 23: Kalman gain matrix K entries convergence over time for relatively high Q matrix entries (difference with R of about 10^2)