University of Twente

EEMCS / Electrical Engineering
Robotics and *Mechatronics*

Real-Time Environmental Impedance Estimation Using a Variable Stiffness Actuator as a Variable Stiffness Sensor

R.M. (Rick) van Keken

MSc Report

Committee:
Prof.dr.ir. S. Stramigioli
Dr. R. Carloni
A. Yenehun Mersha, MSc
Dr.ir. R.G.K.M. Aarts

July 2013

Report nr. 011RAM2013
Robotics and Mechatronics
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Summary

This report describes the design and implementation of an estimation algorithm to estimate environmental impedance parameters, the stiffness and damping of a wall, using a variable stiffness actuator as a variable stiffness sensor. The estimation algorithm uses an observer, which is first shown to be stable with a bounded error, in combination with an extended kalman filter. The estimation algorithm is validated by both simulation and experimental results. Correct stiffness and damping estimates can be shown in the simulations but experiments on the real setup were unable to obtain correct damping estimates. Since the goal with respect to damping has not been accomplished, first a paper is presented which only treats the stiffness. The next chapters will introduce the damping to the system and will go into more detail.

Contents

1 Introduction

In the field of robotics, actuators are very often in interaction with an external environment. This environment can be anything, from an object that is grasped to a wall that is cleaned to a human body. For the control of the actuator and the performance that is achieved it is often important or beneficial to have information about the environment that needs to be interacted with. Frequently it is not known beforehand what type of environment will be encountered, for example when piloting an Unmanned Areal Vehicle (UAV). Because of this research is done on real time impedance estimation of the environment.

For this estimation multiple sensors are required which add to the complexity, cost and uncertainty of the system. This report introduces a way to use a Variable Stiffness Actuator (VSA) as a Variable Stiffness Sensor (VSS). A VSA is a type of actuator with internal springs and degrees of freedom that let it mechanically change its apparent output stiffness without changing its output position. The report shows that these special types of actuators are capable of this feat. The advantage is that a system using these types of actuators is then capable of obtaining environmental data during its normal operation.

In this report the stiffness and damping factor of a wall will be estimated in real time with a VSA. The estimation algorithm uses an observer, which is first shown to be stable with a bounded error, in combination with an extended kalman filter. The estimation algorithm is validated by both simulation and experimental results. The results of the damping estimation part of this MSc project were unfortunately not yet satisfactory enough to treat in the paper that has been written. Because of this the paper only shows the results of the estimation algorithm in the case where there is only a stiffness to estimate. The rest of the report will treat the damping in combination with the stiffness.

In the section after the paper, first the environmental damping will be introduced to the overall system. Next the estimation algorithm will be extended to also include this damping. Then the results from simulations and a real setup are shown which leads to the conclusion and some recommendations.

In the appendices a short manual on how to operate the used setup can be found for researcher that will continue with this project. The appendices also contain some more details about the model and the used system and in the end some of the used code is displayed.

Real-Time Environmental Stiffness Estimation Using a Variable Stiffness Actuator as a Variable Stiffness Sensor

R.M. van Keken, A.Y. Mersha and R. Carloni*

Abstract— **In many applications where a robot is in interaction with an environment it is important or beneficial to have information about that environment. This paper shows that a Variable Stiffness Actuator can be used to estimate the stiffness parameter of an external environment. This way the VSA can dually be used as a sensor system. By using the VSA as a Variable Stiffness Sensor there is no need for extra sensors and additional electronics which are normally necessary when estimating environmental parameters. An estimation algorithm is introduced that uses the inputs and outputs of a VSA to execute this task. The estimation algorithm uses a combination of an observer with an Extended Kalman Filter. The correct workings of the algorithm is validated through simulation and verified through actual experiments.**

I. INTRODUCTION

When a device interacts with an external environment, be it a structure or an object, the characteristics of the subject being manipulated has a big influence on the operation of the device. Hence it is advantageous to have more information of the environment in situations where (precise) manipulation or interaction is required. The stiffness of the environment can severely influence the controller action of the actuator. In general it is beneficial for controllers to know the stiffness of the controlled system accurately such that efficient and stable control can be obtained. Stiffness estimation of the environment is thus of interest as is for example shown in [1].

For the estimation of the environmental stiffness it is normally necessary to attach multiple extra sensors to the system. This can lead to more complex electronics, room management issues and more power consumption. An actuator that can serve dually as a sensor at the same time would thus be very interesting.

This paper proposes proper exploitation of Variable Stiffness Actuators (VSAs) as Variable Stiffness Sensors (VSSs). This choice is due to their wide range of applications and their adaptability in dynamic situations. Multiple types of VSAs have been developed, with the common ability to mechanically change their apparent output stiffness independently of their output position. This is done by changing internal Degrees of Freedom (DoFs) that modify the way internal springs are felt at the actuator output. Two main configurations for a VSA are available. The first is an agonist-antagonist setup where the difference between two DoFs determine the output position and they together change the apparent output stiffness. The second is a serial configuration where one DoF changes the output position and one DoF changes the stiffness. VSAs are useful in any situation where their unique ability can be beneficial. For example applications where safety [2], human-robot interaction [3] or saving energy [4] play a role.

This paper combines the unique features of a VSA together with a real-time stiffness estimation algorithm to estimate environmental parameters. More precisely, the stiffness of an environment is estimated. This way the flexibility in actuating a robotic system, introduced by a VSA's properties, can be combined with an enhanced understanding of the environment. Showing that the VSA can be used as a VSS is the main contribution of this paper. A major advantage of this is that a VSA, while doing its normal tasks, can automatically obtain information about the environment. No extra sensor systems are necessary since the actuator serves as a sensor as well.

The used estimation algorithm also gives estimates of different states of the system including the stiffness of the VSA which is not directly measurable. Accurately estimating the stiffness of a VSA is a challenge on its own since the stiffness of VSAs can change and is often highly nonlinear. Several papers have addressed the problem of how to estimate the current output stiffness of a VSA or a flexible robot joint. Multiple methods are available that only use the encoder outputs from the output of the device and the DoFs, that control the device together with its inputs. In [5] an observer is shown that uses an update law by calculating the error dynamics of the expected forces from derivatives obtained from the system. The mentioned observer also serves as a basis for an extension on an Extended Kalman Filter (EKF) used in this work. Other methods for estimating VSA stiffnesses have been presented in [6] and [7], where derivatives are circumvented by using a parametric observer. Other examples are [8] and [9] which use a two step approach. First residuals, based on first or second order filtered signals from the system, are generated after which a least square fitting method is used based on a parametric model.

This paper is organised as follows. In Section II a generic view of the overall system is given together with the intrinsic dynamics of a VSA after which in Section III the used estimation algorithm is presented. In Section IV a more detailed view is given of the actual setup and the used VSA. Then in Section V and VI simulated data and experimental results are shown respectively. In the end a discussion and conclusion are given in Section VII and VIII.

*R.M. van Keken, A.Y. Mersha and R. Carloni are with the Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, The Netherlands. Email: r.m.vankeken@student.utwente.nl, a.y.mersha@utwente.nl, r.carloni@utwente.nl

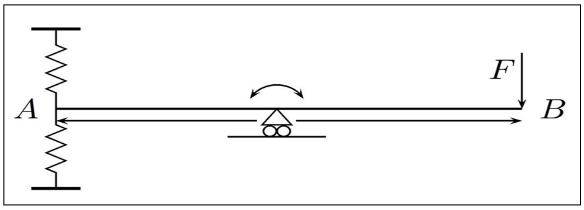


Fig. 1: Conceptual scheme of the variable transmission ratio lever arm, obtained by means of the moving pivot point. The stiffness K is zero when the pivot point is at A and it is infinite when the pivot point is at B [11].

II. Overall System

A generic VSA is controlled by a number of internal elastic elements, such as springs, and a number of actuated DoFs q_i, i.e. the motors. The apparent output stiffness K_{vsa} is determined by both the configuration of the internal DoFs and of the internal springs [10].

In this work a rotational serial configuration type of VSA is considered with two DoFs. The considered class of VSA contains an internal lever arm with a variable effective length. The VSA that is used is the vsaUT-II [11]. The first DoF of the vsaUT-II, q_1, is used to change a pivot point which modifies the effective length of the internal lever arm which changes the output stiffness. The second DoF, q_2, changes the equilibrium position of the output position r of the VSA.

A conceptual scheme of a serial configuration VSA using a moving pivot point principle is shown in Figure 1. The force F, visible in Figure 1, leads to a torque around the pivot point by multiplying it by the effective arm length. The output stiffness of a rotational VSA is defined by the infinitesimal change in torque divided by the infinitesimal change in position caused by this change. K_{vsa} is generally a function of the internal DoFs and the output position:

$$K_{vsa} := \frac{\partial T}{\partial r} = f(q_1, q_2, r) \tag{1}$$

The dynamics of the environment are modelled as a linear spring. The damping of the environment is assumed to be low and by using slow motions it is supposed that the damping can be neglected. The dynamics introduced by the environment are described by the following formula:

$$F_w = K_w(x_w - x_{w0}) \tag{2}$$

$$T_w = K_w(x_w - x_{w0})L \tag{3}$$

Where F_w is the force that is exerted by the environment. T_w is the torque that the environment exerts on the VSA which is obtained by multiplying F_w with the lever arm length of the output of the VSA, L. K_w is the assumed constant translational stiffness of the spring which needs to be estimated, x_w is the deflection of the spring and x_{w0} is the equilibrium position of the spring. Since the environment is modelled as a translational spring and the VSA is rotational, $x_w = rL$. Where rL is the translational distance covered by the endpoint of the lever arm of the VSA.

For the system the following states are defined that are also used in the estimation algorithm:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} r \\ \dot{r} \\ q_1 \\ q_2 \\ K_{vsa} \\ K_w \end{bmatrix} \tag{4}$$

Besides K_w, r, $\dot{r}$, q_1, q_2 and K_{vsa} have been included as states to be estimated by the observer. Measurements of r, q_1 and q_2 are available but due to the limited accuracy of measurements, an estimated value will lead to better results. $\dot{r}$ and K_{vsa} are derived from the measurements and are added as states for the same reason. When the VSA is in contact with the environment the behaviour can be described as in:

$$J_{vsa} * \dot{x}_2 = x_5(x_4 - x_1) - D_{vsa}x_2 - x_6(x_1 L - x_{w0})L \tag{5}$$

$$T_w = x_5(x_4 - x_1) - D_{vsa}x_2 - J_{vsa}\dot{x}_2 = x_6(x_1 L - x_{w0})L \tag{6}$$

D_{vsa} and J_{vsa} are the damping and rotational inertia of the VSA at the output respectively. These are assumed constant. The last part of eq. 6 equals eq. 3.

III. The Estimation Algorithm

The estimation algorithm consists of several separate parts. The global overview is shown in Figure 2. The Update Law (UL) is an observer that generates an update law used to estimate the stiffness of the environment. The Extended Kalman Filter (EKF) gives the optimal estimate of the states.

A. The Update Law

Hereafter, the workings of an observer that estimates the parameters of an environment using inputs from the VSA is explained. There is only interaction when the VSA and the environment are in contact, thus when the output $rL > x_{w0}$. The observer is based on [5]. The observer makes use of derivatives which are obtained through a State Variable Filter (SVF).

1) Error Dynamics: The torque that the environment exerts on the VSA is:

$$T_w = x_6(x_1 L - x_{w0})L \tag{7}$$

See eq. 6. However to obtain the update law the error dynamics are necessary. The derivative of eq. 7 is taken:

$$\dot{T}_w = x_6 x_2 L^2 \tag{8}$$

Since the real value of the parameter of the environment is not known, only an estimate can be made:

$$\dot{\hat{T}}_w = \hat{x}_6 x_2 L^2 \tag{9}$$

This leaves an error due to the estimate of K_w in the derivative:

$$\dot{\tilde{T}}_w = \dot{T}_w - \dot{\hat{T}}_w = \tilde{x}_6 x_2 L^2 \tag{10}$$

To calculate the error in the torque the real torque must be known as well as the estimated torque. This real torque is

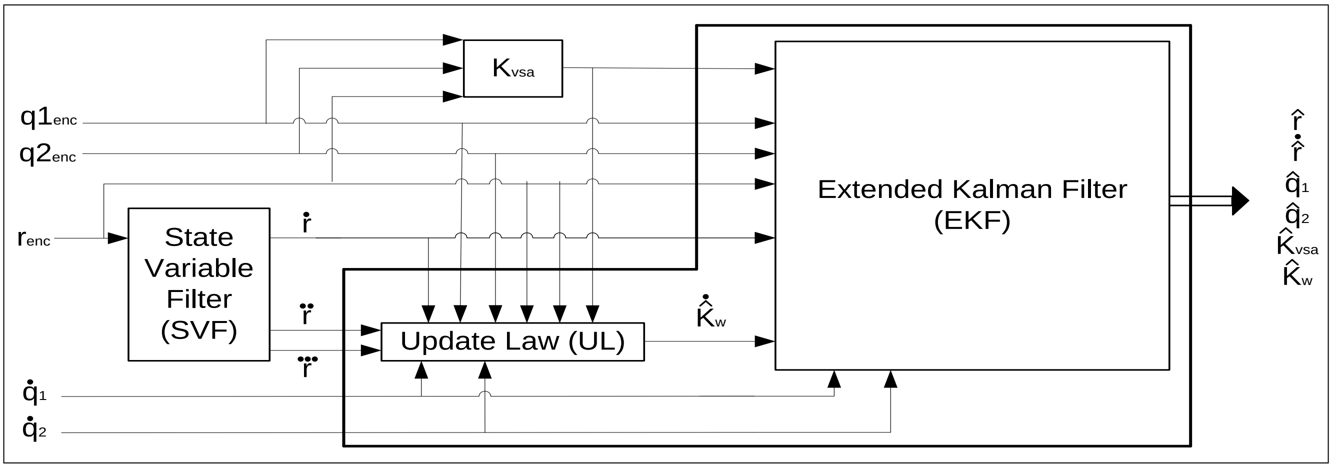


Fig. 2: Overview of the estimation algorithm consisting of an extended kalman filter in combination with an observer. The measured outputs of the VSA are used to calculate K_{vsa} and the derivatives of r. These signals are used, together with the known input signals of the VSA, by the estimation algorithm.

obtained from the VSA from the first part of the torque balance shown in eq. 6. The derivative of the first part of eq. 6 becomes:

$$\dot{T}_w = \dot{x}_5(x_4 - x_1) + x_5(\dot{x}_4 - x_2) - D_{vsa}\dot{x}_2 - J_{vsa}\ddot{x}_2 \quad (11)$$

Here x_i represent the different states of the system according to eq. 4. With this the error in the torque derivative due to the error in the estimated K_w can be calculated:

$$\dot{\tilde{T}}_w = \dot{T}_w - \dot{\hat{T}}_w \quad (12)$$

2) Update Law: First a positive definite error function is defined:

$$V = \frac{1}{2}\tilde{K}_w^2 = \frac{1}{2}\tilde{x}_6^2 \quad (13)$$

If the derivative can be shown to be negative semi definite then the error in K_w is bounded [5].

$$\dot{V} = \tilde{x}_6\dot{\tilde{x}}_6 \quad (14)$$

$$\dot{V} = \tilde{x}_6(\dot{x}_6 - \dot{\hat{x}}_6) \quad (15)$$

$$\dot{V} = -\tilde{x}_6\dot{\hat{x}}_6 \quad (16)$$

For $\dot{\hat{x}}_6$ the following update law is chosen:

$$\dot{\hat{x}}_6 = \alpha\dot{\tilde{T}}_w x_2 = \alpha\tilde{x}_6 x_2 L^2 x_2 = \alpha\tilde{x}_6 L^2 x_2^2 \quad (17)$$

Which leads to:

$$\dot{V} = -\tilde{x}_6\alpha\tilde{x}_6 L^2 x_2^2 \quad (18)$$

$$\dot{V} = -\alpha\tilde{x}_6^2 L^2 x_2^2 \quad (19)$$

Which is negative semi definite for $\alpha > 0$. Note that α can be used as a design parameter to influence the convergence speed and the errors on the steady state value.

B. Extended Kalman Filter

To be less dependent on derivatives which tend to generate noise, a Kalman filter has been implemented. Because of the intrinsic non-linear behaviour of the VSA an EKF has been introduced. The EKF uses five direct inputs, q_1, q_2 and r which are measured and $\dot{q}_1$ and $\dot{q}_2$ which are the motor inputs. Furthermore the EKF uses two inputs derived from these five ($\dot{r}$ and K_{vsa}). $\dot{r}$ is generated using a SVF and K_{vsa} is calculated using a model based function dependent on the measurements, see eq. 1. Using $\dot{r}$ and K_{vsa} as measurements lowers the convergence time of the estimate to its final value. Since the real value of K_w is assumed constant, normally in the prediction phase of the EKF it would use:

$$\hat{K}_w(t_2) = \hat{K}_w(t_1) \quad (20)$$

Where t_i represents a time and $t_2 > t_1$. By adding the update law this changes to:

$$\hat{K}_w(t_2) = \hat{K}_w(t_1) + \int_{t_1}^{t_2} \dot{\hat{K}}_w \quad (21)$$

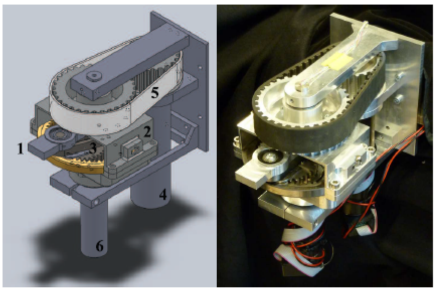
This further lowers the convergence time.

IV. The Physical Setup

The physical setup which is used for validation of the proposed estimation is briefly described. A CAD drawing and a photograph are displayed in Figure 3. Due to the mechanical configuration of the VSA, the theoretical apparent output stiffness can be calculated using:

$$x_5 = 2kl^2\frac{(l - x_3)^2}{x_3^2}cos(2(x_1 - x_4)) \quad (22)$$

Here k is the individual stiffness of the two internal springs of the VSA and l is the length over which the internal pivot point can move. However the compliance of the driving belt of the VSA, see Figure 3 label 5, has an effect on the effective apparent output stiffness as well. It is assumed that the effective stiffness of the belt, K_{belt}, at the output is placed



(a) CAD drawing. (b) Physical prototype.

Fig. 3: The vsaUT-II variable stiffness actuator - The labels indicate 1) the output, 2) the actuator frame, 3) the lever arm and gears mechanism, 4) motor for changing output position, 5) timing belt transmission and 6) motor for varying output stiffness [12].

in series with the theoretical stiffness of the VSA. Hence the real apparent output stiffness becomes:

$$K_{vsa_{eff}} = \frac{x_5 K_{belt}}{x_5 + K_{belt}} \tag{23}$$

The compliance of the driving belt in the VSA is anticipated in the code of the estimation algorithm. The estimated result would otherwise be a factor $\frac{x_5+K_{belt}}{K_{belt}}$ too high.

The vsaUT-II contains two main sources of internal damping. These are the damping occurring in the driving belt and the damping of the actuator frame of the device. The rotational inertia of the VSA and the damping constant at the output of the VSA have been determined in [11]. Their values are $J_{vsa} = 0.0108 Nms^2/rad$ and $D_{vsa} = 1.2 * 10^{-2} Nms/rad$ respectively.

There are three encoders present in the setup. One 10bits absolute magnetic encoder on the output position and two rotational encoders with 2000ppr, one on each of the two motors that drive the DoFs q_1 and q_2. The system is controlled using the real time environment of Matlab Simulink which runs at a sample frequency, F_s, of 200Hz.

The environment is formed using linear springs with an exactly known stiffness factor of $K_w = 200N/m$ and $K_w = 1000N/m$. The used springs are extension springs to prevent unwanted bending and other non-linear effects that occur when using compression springs. In real applications the actuator will most often push against an environment. Hence, compression springs are intuitively closer to resembling an actual application than extension springs. However for the estimation algorithm this makes no difference when the spring is linear. The spring is attached to the VSA on one end and with a hinge to a fixed position on the other end such that the spring can rotate together with the VSA and the spring will never bend sideways, see Figure 4.

V. Simulated Results

In this part first simulations are shown that provide evidence for the proof of concept when there are no limitations caused by encoders or unwanted system properties.

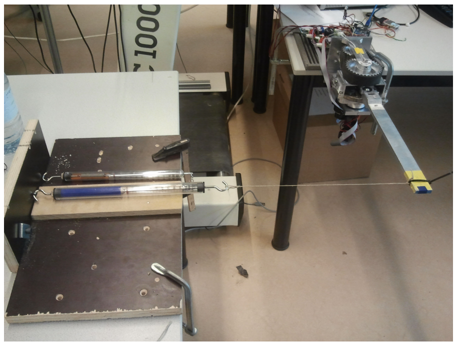
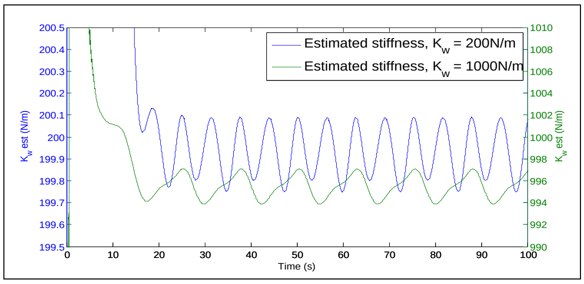


Fig. 4: Photo of the used environmental setup.

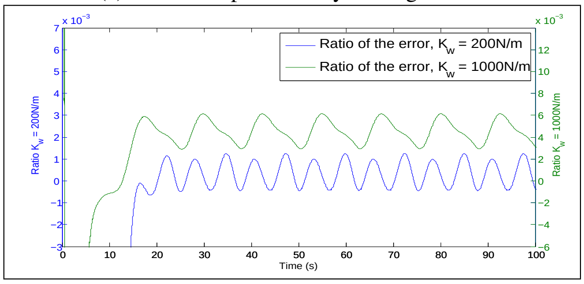
Afterwards simulations are done in a setting where these limitations are present. The complete system is modelled in the simulation software 20Sim. The VSA should be able to be used as a VSS in dynamic applications. To show that the estimation algorithm works when the VSA is in motion, the equilibrium position of the VSA is constantly changed using a sinusoidal motion profile on the setpoint for q_2. The bandwidth of the SVF is low to suppress noise on the signals. In case the system is dependent on faster dynamics the bandwidth should be chosen higher such that there is less delay in the derivatives, this is a direct trade off with the noise. The algorithm is tested on the two different springs used to show the effects of different environments. During the experiments K_{vsa} is kept constant at a value of 400Nm/rad unless mentioned otherwise.

A. Simulations for the Generic Case

During these simulations it is assumed that there are no encoders present in the system and hence there is full accuracy of the measured signals. The algorithm and control are discrete but with twice the sampling frequency than on the real setup. Also unwanted effects in the VSA such as the limited stiffness of the driving belt or the dissipations in the driving belt and the actuator frame are removed. The VSA can be seen as a black box of which the output parameters (stiffness, damping and inertia) are exactly known. It is assumed there is no prior knowledge of the stiffness of the environment, hence the initial condition of the estimate is zero. Figures 5a and 5b show the simulation results of implementing the estimation algorithm on the VSA. As is visible, the error drops down to approximately 0.5% in case a K_w of $1000N/m$ is used and even lower when a K_w of $200N/m$ is used. These simulations show that when there is unlimited accuracy and no unwanted system effects in the VSA and thus the output parameters of the a VSA are exactly known, the estimation algorithm lowers the error in the stiffness estimate of K_w to a small and bounded value like it is supposed to. It should be noted that the used motion profile is visible in the reached steady state value. The frequency of the motion profile is visible in the oscillations on the estimate. This might be due to small errors introduced by the dynamics of the system or delay in the derivative

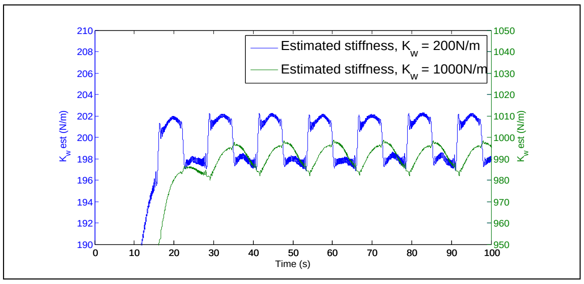


(a) Estimates provided by the algorithm.

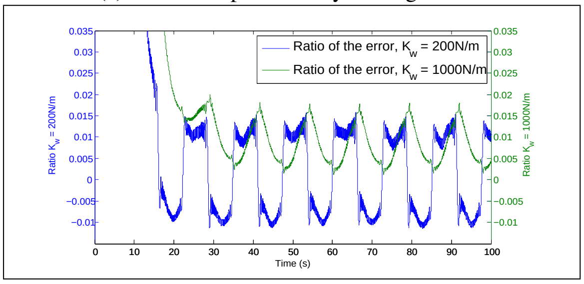


(b) Ratio of the errors in the estimate divided by the actual value.

Fig. 5: Results of the simulation with unlimited accuracy and no unwanted effects.



(a) Estimates provided by the algorithm.



(b) Ratio of the errors in the estimate divided by the actual value.

Fig. 6: Results of the simulation using encoders and in the presence of unwanted system effects.

signals.

B. Simulations Based on the Competent Model of the VSA

The algorithm is also tested in simulations using the actual model of the VSA. The model is as close to the real setup as possible. All input signals are now quantised by encoders and parasitic and unwanted effects in the mechanical system are taken into account.

Movements are kept relatively slow such that the internal damping of the VSA can be neglected. The results can be seen in Figures 6a and 6b. The ratio of the error now oscillates around 1% with a higher amplitude than in the previous case. Since a lot of limitations were introduced this is as expected. However the error is still bounded and small.

Also simulations have been done to see if the final estimate depends on the stiffness of the VSA. Figure 7 shows the results for three simulations where different constant stiffnesses for the K_{vsa} were used. It is clear that although the estimates differ slightly, the estimates come close to the real value.

VI. Experimental Results

In this section the experimental results are shown and discussed. Results of the experiments are shown in Figure 8a and 8b where it is visible that the steady state error is bounded and within a margin of 10%. The higher final error can possibly be explained by noise in the system and modelling errors.

The same experiment has been done using a random motion profile to better mimic real applications. The results can be seen in Figure 9a and 9b. The error in the estimate stays bounded. The frequency of the deflections around the

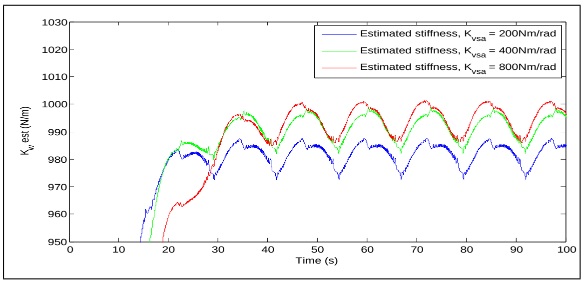


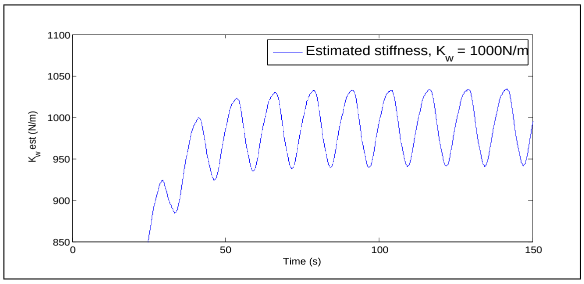
Fig. 7: Effect of using a different K_{vsa} on the estimate, $K_w = 1000N/m$.

steady state value are now more random as is to be expected with the used random motion profile.

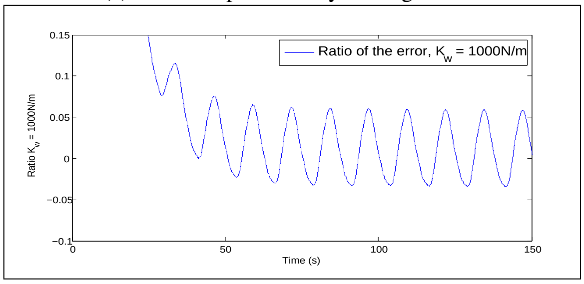
During the experiments it was found that the estimate is dependent on the stiffness of the VSA in contrast to the simulations where this is not the case, see Figure 7. It is suspected that this is due to a backlash effect in the setup of the VSA caused by the connection between the input pulley and the gearbox. At a certain externally applied torque at the output, the connection slips briefly before engaging again. This causes the apparent output stiffness of the VSA to be lower than according to the theory. It should be noted that this is an imperfection in the realisation of the vsaUT-II and not an inherent fault in its concept. See Figure 10 for a visualisation of the effect. When the K_{vsa} is lowered, the estimate will be lower as well. When the K_{vsa} is increased, the estimate is also higher.

VII. Discussion

During the experiments on the real setup it became clear that the results suffered from an unmodelled mechanical



(a) Estimate provided by the algorithm.



(b) Ratio of the error in the estimate divided by the actual value.

Fig. 8: Results of the experiments on the actual setup, $K_w = 1000N/m$.

limitation causing the estimate to be dependent on K_{vsa}. Because of this the vsaUT-II can only estimate the order of magnitude of K_w at the moment. This effect needs to be compensated for if the estimation algorithm is to be implemented more precisely on the vsaUT-II.

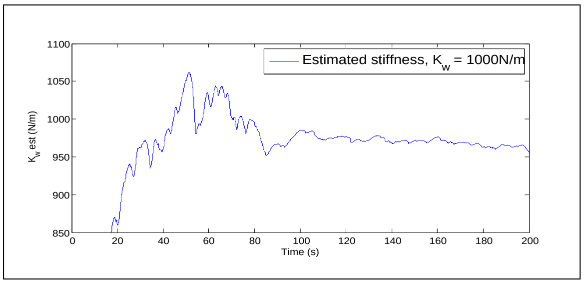
VIII. Conclusion

In this paper it has been shown that a VSA can be used as a VSS, making extra sensors for environmental parameter estimation redundant. This further shows the broad applicability of these types of actuators. The estimation algorithm has been demonstrated in both simulations and experiments where the stiffness was estimated till within a small and bounded error. It has been shown that when the output parameters (stiffness, damping and inertia) of a VSA are known, the algorithm will give accurate estimates.

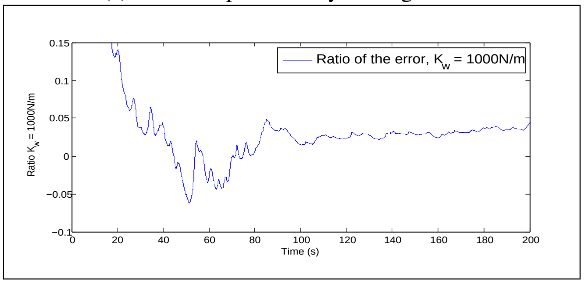
In future work there will be a focus on adding a damping estimate of the environment to the algorithm as well. The VSA will then be able to be used as a Variable Impedance Sensor (VIS) which further enhances the applicability.

References

[1] N. Diolaiti, C. Melchiorri and S. Stramigioli, "Contact impedance estimation for robotic systems", IEEE Transactions on Robotics, vol. 21, no. 5, pp. 925-935, 2005.
[2] R.J. Wang and H.P. Huang, "An active-passive variable stiffness elastic actuator for safety robot systems", in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3664-3669, 2010.
[3] G. Tonietti, R. Schiavi and A. Bicchi, "Design and Control of a Variable Stiffness Actuator for Safe and Fast Physical Human/Robot Interaction", in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 526-531, 2005.
[4] L. C. Visser, R. Carloni and S. Stramigioli, "Energy-Efficient Variable Stiffness Actuators", IEEE Transactions on Robotics, vol. 27, no. 5, pp. 865-875, 2011.



(a) Estimate provided by the algorithm.



(b) Ratio of the error in the estimate divided by the actual value.

Fig. 9: Results of the experiments on the actual setup with a random motion profile, $K_w = 1000N/m$.

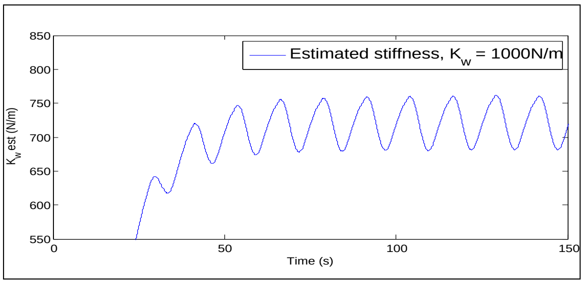


Fig. 10: Effect of using a lower K_{vsa} on the estimate, $K_{vsa} = 200Nm/rad$.

[5] G. Grioli and A. Bicchi, "A Non-Invasive, Real-Time Method for Measuring Variable Stiffness", in Proceedings of Robotics: Science and Systems, 2010.
[6] G. Grioli, A. Bicchi, "A real-time parametric stiffness observer for VSA devices", in Proceedings of the IEEE International Conference on Robotics and Automation, 2011.
[7] T. Menard, G. Grioli and A. Bicchi, "A real time robust observer for an agonist antagonist variable stiffness actuator", in Proceedings of the IEEE International Conference on Robotics and Automation, 2013.
[8] F. Flacco and A. De Luca, "Residual-based stiffness estimation in robots with flexible transmissions", in Proceedings of the IEEE International Conference on Robotics and Automation, 2011.
[9] F. Flacco, A. De Luca, I. Sardellitti and N.G. Tsagarakis, "On-line estimation of variable stiffness in flexible robot joints", International Journal of Robotics Research, vol. 31, issue 13, pp. 1556-1577, 2012.
[10] R. Carloni and L. Marconi, "Limit Cycles and Stiffness Control with Variable Stiffness Actuators", in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5083-5088, 2012.
[11] S.S. Groothuis, G. Rusticelli, A. Zucchelli, S. Stramigioli and R. Carloni, "The Variable Stiffness Actuator vsaUT-II: Mechanical Design, Modeling and Identification", in Proceedings of the IEEE/ASME Transactions on Mechatronics, 2013.
[12] S.S. Groothuis, G. Rusticelli, A. Zucchelli, S. Stramigioli and R. Carloni, "The vsaUT-II: a Novel Rotational Variable Stiffness Actuator"

3 Overall System

The overall system considered is now extended to also incorporate the damping factor of the wall. For this the linear Kelvin-Voigt model is used which means the wall is assumed to be a linear spring damper system (variables that are mentioned in the paper are not mentioned again):

$$F_w = K_w(rL - x_{w0}) + D_w\dot{r}L \tag{3.1}$$

Here D_w is the damping factor of the wall that now also needs to be estimated. D_w is considered to be constant just as K_w.

When the VSA is in contact with the wall the behaviour can now be described as in:

$$J_{vsa} * \ddot{r} = K_{vsa}(q_2 - r) - D_{vsa}\dot{r} - K_w(rL - x_{w0})L - D_w\dot{r}L^2 \tag{3.2}$$

$$K_{vsa}(q_2 - r) - D_{vsa}\dot{r} - J_{vsa}\ddot{r} = K_w(rL - x_{w0})L + D_w\dot{r}L^2 = T_w \tag{3.3}$$

Since an extra variable of interest is added to the system, the system states are redefined:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} r \\ \dot{r} \\ q_1 \\ q_2 \\ K_{vsa} \\ K_w \\ D_w \end{bmatrix} \tag{3.4}$$

Notice that the damping of the wall, D_w, which needs to be estimated is now added to the states. The update law that is generated by the observer is also altered to produce a $\dot{\hat{D}}_w$. The total estimation algorithm overview is visible in figure 1. The changes inside the different blocks of the estimation algorithm will be treated further on.

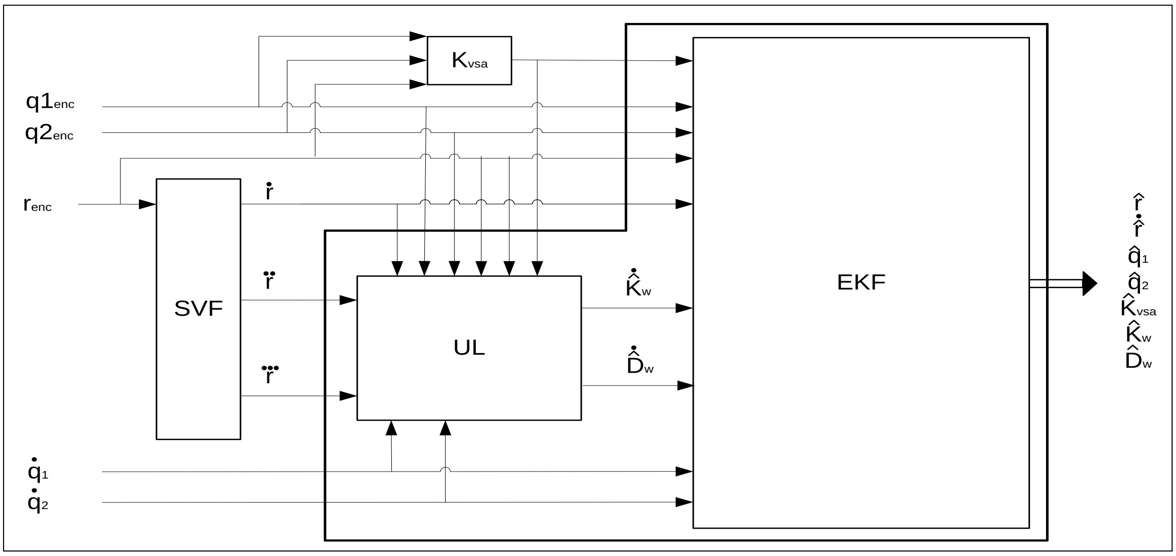


Figure 1: Overview of the estimation algorithm consisting of an extended kalman filter in combination with an observer. The measured outputs of the VSA are used to calculate K_{vsa} and the derivatives of r. These signals are used, together with the known input signals of the VSA, by the estimation algorithm which now also includes D_w.

4 Update Law with Damping

In this part the update law will be treated but now with the damping included. The update law is still based on the observer in [1]. The introduced damping estimate is far more sensitive to the dynamics of the system than the stiffness estimate, this is logical since the damping effect is only perceived while the system is in motion. Because of this a larger value for the bandwidth of the State Variable Filters (SVFs), which generate the time derivatives of the output r, is chosen. This leads to more noise on the signals but less delay in the outputs. Delay in the derivative signals is severely detrimental to the final estimate of D_w. The assumptions made in the paper still hold, the VSA is in contact with the wall $rL \geq x_{w0}$, see figure 2.

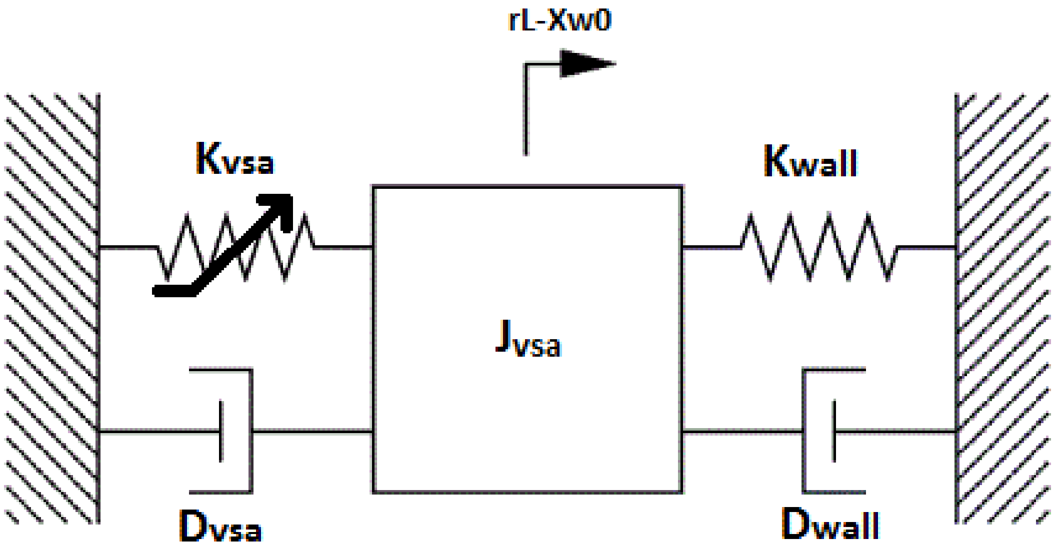


Figure 2: Simplified contact model of the VSA with the wall

4.1 Error Dynamics

The torque that the wall exerts on the VSA is:

$$T_w = K_w(rL - x_{w0})L + D_w \dot{r} L^2 \tag{4.1}$$

Next the error dynamics need to be calculated thus, like in the paper, the derivative of eq. 4.1 is taken:

$$\dot{T}_w = K_w \dot{r} L^2 + D_w \ddot{r} L^2 \tag{4.2}$$

Since the true value of the parameters of the wall are not known, only an estimate can be made:

$$\dot{\hat{T}}_w = \hat{K}_w \dot{r} L^2 + \hat{D}_w \ddot{r} L^2 \tag{4.3}$$

This leaves an error due to the estimation of K_w in the derivative:

$$\dot{\tilde{T}}_w = \tilde{K}_w \dot{r} L^2 + \tilde{D}_w \ddot{r} L^2 \tag{4.4}$$

To calculate the error in the torque the true torque must be known as well as the estimated torque. This true torque is obtained from the VSA using the first part of the torque balance shown in eq. 3.3. The derivative of the first part of eq. 3.3 becomes:

$$\dot{T}_w = \dot{x}_5(x_4 - x_1) + x_5(\dot{x}_4 - x_2) - D_{vsa}\dot{x}_2 - M_{vsa}\ddot{x}_2 \tag{4.5}$$

Here x_i represent the different states of the algorithm according to eq. 3.4. With this the error in the torque derivative due to the error in the estimated K_w can be calculated:

$$\dot{\tilde{T}}_w = \dot{T}_w - \dot{\hat{T}}_w \tag{4.6}$$

4.2 Update Law

First a new positive definite error function is defined:

$$V = \frac{1}{2}\tilde{K}_w^2 + \frac{1}{2}\tilde{D}_w^2 \tag{4.7}$$

The derivative must be shown to be negative definite again to prove that the error in K_w and D_w will be bounded.

$$\begin{aligned} \dot{V} &= \tilde{K}_w\dot{\tilde{K}}_w + \tilde{D}_w\dot{\tilde{D}}_w \\ \dot{V} &= \tilde{K}_w(\dot{K}_w - \dot{\hat{K}}_w) + \tilde{D}_w(\dot{D}_w - \dot{\hat{D}}_w) \\ \dot{V} &= -\tilde{K}_w\dot{\hat{K}}_w - \tilde{D}_w\dot{\hat{D}}_w \end{aligned} \tag{4.8}$$

The following update laws have been chosen to make $\dot{V}$ negative definite:

$$\dot{\hat{K}}_w = \alpha_K\dot{\tilde{T}}_w x_2 = \alpha_K(\tilde{K}_w x_2 L^2 + \tilde{D}_w\dot{x}_2 L^2)x_2 = \alpha_K\tilde{K}_w x_2^2 L^2 + \alpha_K\tilde{D}_w\dot{x}_2 x_2 L^2 \tag{4.9}$$

$$\dot{\hat{D}}_w = \alpha_D\dot{\tilde{T}}_w\dot{x}_2 = \alpha_D(\tilde{K}_w x_2 L^2 + \tilde{D}_w\dot{x}_2 L^2)\dot{x}_2 = \alpha_D\tilde{K}_w x_2\dot{x}_2 L^2 + \alpha_D\tilde{D}_w\dot{x}_2^2 L^2 \tag{4.10}$$

$$\dot{V} = -\alpha_K\tilde{K}_w^2 x_2^2 L^2 - \alpha_K\tilde{K}_w\tilde{D}_w x_2\dot{x}_2 L^2 - \alpha_D\tilde{K}_w\tilde{D}_w x_2\dot{x}_2 L^2 - \alpha_D\tilde{D}_w^2\dot{x}_2^2 L^2 \tag{4.11}$$

The first and the last part are negative definite while the two middle terms are indefinite in sign. The equation can be rewritten as follows, furthermore $\alpha_k = \alpha_D = \alpha$.

$$\begin{aligned} \dot{V} &= -\alpha(\tilde{K}_w^2 x_2^2 + \tilde{K}_w\tilde{D}_w x_2\dot{x}_2 + \tilde{K}_w\tilde{D}_w x_2\dot{x}_2 + \tilde{D}_w^2\dot{x}_2^2)L^2 \\ \dot{V} &= -\alpha(\tilde{K}_w^2 x_2^2 + 2\tilde{K}_w\tilde{D}_w x_2\dot{x}_2 + \tilde{D}_w^2\dot{x}_2^2)L^2 \\ \dot{V} &= -\alpha(\tilde{K}_w x_2 + \tilde{D}_w\dot{x}_2)^2 L^2 \end{aligned} \tag{4.12}$$

By rewriting the formula for $\dot{V}$ like this it shows that $\dot{V}$ is negative definite for $\alpha > 0$ and hence the errors are bounded.

5 Extended Kalman Filter with Damping

In this section the discrete Extended Kalman Filter (EKF) which delivers the final estimate will be described in more detail. In the first step all the necessary matrices inside the EKF are initialised:

$$\begin{aligned} x[7,1] &= x_o[7,1] \\ P[7,7] &= P_0[7,7] \end{aligned} \tag{5.1}$$

Here x is the state and P the state covariance. x_0 and P_0 are the initial state and the initial state covariance respectively.

For ease of reference the system state is repeated:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} r \\ \dot{r} \\ q_1 \\ q_2 \\ K_{vsa} \\ K_w \\ D_w \end{bmatrix} \tag{5.2}$$

In the prediction phase the information obtained from the model is used:

$$x[k+1] = \begin{bmatrix} r(k+1) \\ \dot{r}(k+1) \\ q_1(k+1) \\ q_2(k+1) \\ K_{vsa}(k+1) \\ K_w(k+1) \\ D_w(k+1) \end{bmatrix} = \begin{bmatrix} r(k) + \dot{r}(k)T_s \\ \dot{r}(k) + \ddot{r}(k)T_s \\ q_1(k) + \dot{q}_1 T_s \\ q_2(k) + \dot{q}_2 T_s \\ K_{vsa}(k) + \dot{K}_{vsa} T_s \\ K_w(k) + \dot{\hat{K}}_w T_s \\ D_w(k) + \dot{\hat{D}}_w T_s \end{bmatrix} \tag{5.3}$$

Where T_s is the sampletime and:

$$\ddot{r} = \frac{1}{J_{vsa}}(K_{vsa}(q_2 - r) - D_{vsa}\dot{r} - K_w(rL - x_{w0})L - D_w\dot{r}L^2) \tag{5.4}$$

See eq. 3.2 and:

$$\dot{\hat{K}}_w = \alpha\dot{r}(\dot{K}_{vsa}(q_2 - r) + K_{vsa}(\dot{q}_2 - \dot{r}) - D_{vsa}\ddot{r} - J_{vsa}\dddot{r} - K_w\dot{r}L^2 - D_w\ddot{r}L^2) \tag{5.5}$$

$$\dot{\hat{D}}_w = \alpha\ddot{r}(\dot{K}_{vsa}(q_2 - r) + K_{vsa}(\dot{q}_2 - \dot{r}) - D_{vsa}\ddot{r} - J_{vsa}\dddot{r} - K_w\dot{r}L^2 - D_w\ddot{r}L^2) \tag{5.6}$$

See chapter 4.2 for the explanation about the update law. Now $\dot{K}_{vsa}$ still needs to be calculated. This is done by taking the time derivative of:

$$K_{vsa} = 2kl^2\frac{(l - q_1)^2}{q_1^2}cos(2(r - q_2)) \tag{5.7}$$

$$\dot{K}_{vsa} = \frac{\partial K_{vsa}}{\partial q_1}\dot{q}_1 + \frac{\partial K_{vsa}}{\partial q_2}\dot{q}_2 + \frac{\partial K_{vsa}}{\partial r}\dot{r} \tag{5.8}$$

$\dot{q}_1$ and $\dot{q}_2$ are inputs to the system and $\dot{r}$ is a state.

$$\frac{\partial K_{vsa}}{\partial q_1} = 2kcos(2(r - q_2))L^3(2q_1^{-2} - 2lq_1^{-3}) \tag{5.9}$$

$$\frac{\partial K_{vsa}}{\partial q_2} = 4k(\frac{l}{q_1})^2(l - q_1)^2 sin(2(r - q_2)) \tag{5.10}$$

$$\frac{\partial K_{vsa}}{\partial r} = -4k(\frac{l}{q_1})^2(l - q_1)^2 sin(2(r - q_2)) \tag{5.11}$$

Now the Jacobian of $x(k+1) = f(x(k), u(k))$ can be calculated, u stands for the input signals and equals $u = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$ which are the velocity setpoints for the motors of the two degrees of freedom of the VSA. For ease of notation $f(x(k), u(k))$ will be referred to as $f(x(k))$ from now on. The Jacobian becomes:

$$F[7,7] = \begin{bmatrix} \frac{\partial f(x_1(k))}{x_1(k)} & \cdots & \frac{\partial f(x_1(k))}{x_7(k)} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(x_7(k))}{x_1(k)} & \cdots & \frac{\partial f(x_7(k))}{x_7(k)} \end{bmatrix} = \tag{5.12}$$

$$\begin{bmatrix} 1 & T_s & 0 & 0 & 0 & 0 & 0 \\ \frac{T_s}{J_{vsa}}(-K_{vsa} - K_w L^2) & \frac{\partial f(x_2(k))}{\partial x_2(k)} & 0 & \frac{T_s}{J_{vsa}} K_{vsa} & \frac{T_s}{J_{vsa}}(q_2 - r) & \frac{T_s}{J_{vsa}}(-(rL - x_{w0})L) & \frac{T_s}{J_{vsa}}(-\dot{r}L^2) \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{\partial f(x_5(k))}{\partial x_1(k)} & \frac{\partial f(x_5(k))}{\partial x_2(k)} & \frac{\partial f(x_5(k))}{\partial x_3(k)} & \frac{\partial f(x_5(k))}{\partial x_4(k)} & 1 & 0 & 0 \\ \alpha T_s(-\dot{K}_{vsa} * \dot{r}) & \frac{\partial f(x_6(k))}{\partial x_2(k)} & 0 & \alpha T_s(\dot{K}_{vsa}\dot{r}) & \alpha T_s((\dot{q}_2 - \dot{r})\dot{r}) & 1 + \alpha T_s(\dot{r}^2 L^2) & \alpha T_s(-\dot{r}\ddot{r}L^2) \\ \alpha\ddot{r}T_s(-\dot{K}_{vsa}) & \frac{\partial f(x_7(k))}{\partial x_2(k)} & 0 & \alpha\ddot{r}T_s(\dot{K}_{vsa}) & \alpha\ddot{r}T_s(\dot{q}_2 - \dot{r}) & \alpha\ddot{r}T_s(\dot{r}L^2) & 1 + \alpha\ddot{r}T_s(\ddot{r}L^2) \end{bmatrix}$$

$$\frac{\partial f(x_2(k))}{\partial x_2(k)} = \frac{\partial(\dot{r} + \ddot{r})}{\partial \dot{r}} = 1 + \frac{T_s}{J_{vsa}}(-D_{vsa} - D_w L^2) \tag{5.13}$$

$$\begin{aligned} \frac{\partial f(x_5(k))}{\partial x_1(k)} = \frac{\partial \dot{K}_{vsa}}{\partial r} &= -4kl^3(2q_1^{-2} - 2lq_1^{-3}) sin(2(r - q_2))\dot{q}_1 \\ &+ 8k(\frac{l}{q_1})^2(L - q_1)^2 cos(2(r - q_2))\dot{q}_2 - 8k(\frac{l}{q_1})^2(L - q_1)^2 cos(2(r - q_2))\dot{r} \end{aligned} \tag{5.14}$$

$$\frac{\partial f(x_5(k))}{\partial x_2(k)} = \frac{\partial \dot{K}_{vsa}}{\partial \dot{r}} = -4k(\frac{l}{q_1})^2(l - q_1)^2 sin(2(r - q_2)) \tag{5.15}$$

$$\begin{aligned} \frac{\partial f(x_5(k))}{\partial x_3(k)} = \frac{\partial \dot{K}_{vsa}}{\partial q_1} &= 2kcos(2(r - q_2))L^3(-4q_1^{-3} + 6Lq_1^{-4})\dot{q}_1 \\ &+ 4kl^3(2q_1^{-2} - 2lq_1^{-3}) sin(2(r - q_2))\dot{q}_2 - 4kl^3(2q_1^{-2} - 2lq_1^{-3}) sin(2(r - q_2))\dot{r} \end{aligned} \tag{5.16}$$

$$\begin{aligned} \frac{\partial f(x_5(k))}{\partial x_4(k)} = \frac{\partial \dot{K}_{vsa}}{\partial q_2} &= 4kl^3(2q_1^{-2} - 2lq_1^{-3}) sin(2(r - q_2))\dot{q}_1 \\ &- 8k(\frac{l}{q_1})^2(L - q_1)^2 cos(2(r - q_2))\dot{q}_2 + 8k(\frac{l}{q_1})^2(L - q_1)^2 cos(2(r - q_2))\dot{r} \end{aligned} \tag{5.17}$$

$$\begin{aligned} \frac{\partial f(x_6(k))}{\partial x_2(k)} = \frac{\partial \dot{\hat{K}}_w}{\partial \dot{r}} &= \alpha T_s(\dot{K}_{vsa}(q_2 - r) \\ &+ K_{vsa}\dot{q}_2 - 2K_{vsa}\dot{r} - D_{vsa}\ddot{r} - J_{vsa}\dddot{r} - 2K_w\dot{r}L^2 - D_w\ddot{r}L^2) \end{aligned} \tag{5.18}$$

$$\frac{\partial f(x_7(k))}{\partial x_2(k)} = \frac{\partial \dot{\hat{D}}_w}{\partial \dot{r}} = \alpha\ddot{r}T_s(-K_{vsa} - D_{vsa} - K_w L^2) \tag{5.19}$$

Using these equations the EKF can be implemented. When the update law is not used in combination with the EKF those parts can just be commented out in the code or α can be set to zero.

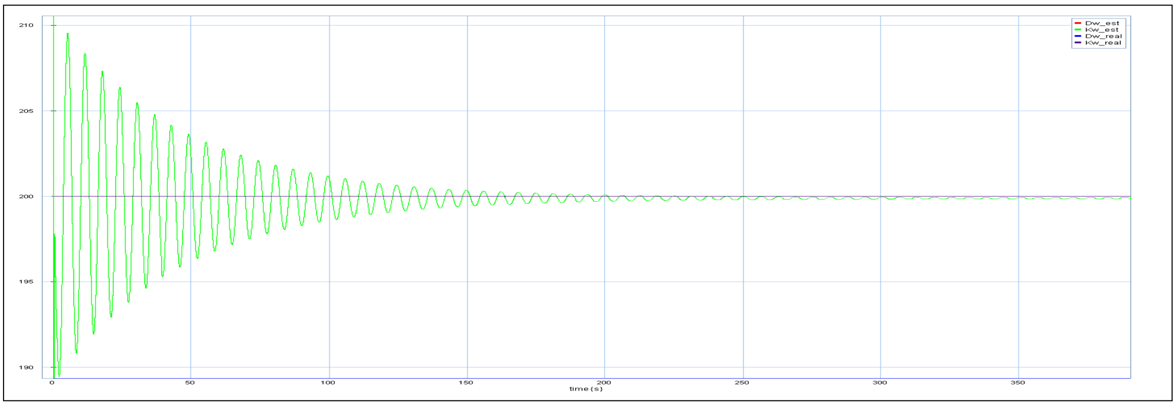
6 Simulations

In this part multiple simulations will be shown which now include the damping estimate as well. Also certain topics like unwanted behaviour, their effects and how they are anticipated are treated more thoroughly. The simulations are done using $K_w = 200$ and $D_w = 2$ unless mentioned otherwise.

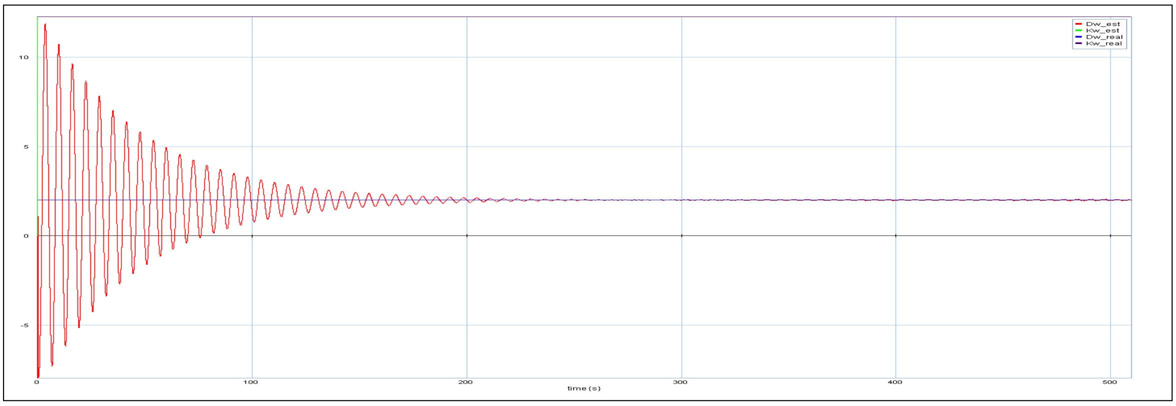
6.1 Proof of Concept

First of all the estimation algorithm and its different parts have been tested to see if they actually work. Simulations have been done in an idealised environment to test the proof of concept. There are no encoders present, hence it is assumed that there is infinite accuracy on the measured signals. Furthermore the main causes of unwanted effects have been eliminated. This means the stiffness of the driving belt of the VSA system has been made practically infinitely stiff, while the dissipation in the belt is set to almost zero. The dissipation caused by the rotation of the actuated device is set to zero. Under these circumstances the VSA can be seen as a black box from which the output stiffness, damping and inertia is known. Hence if it works in this scenario it should theoretically work on different VSAs as well.

The update law First the update law has been tested in the situation just described in continuous time. Because of the high accuracy in the signals there is not much noise present and the design parameter α can be chosen high. The results are shown in figures 3a, 3b, 4a and 4b.

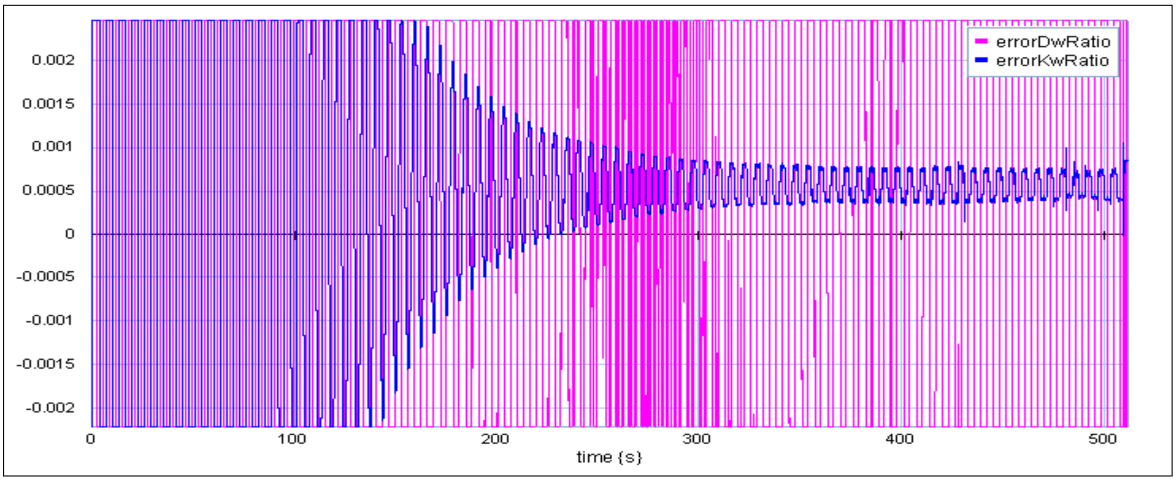


(a) Resulting estimate of K_w

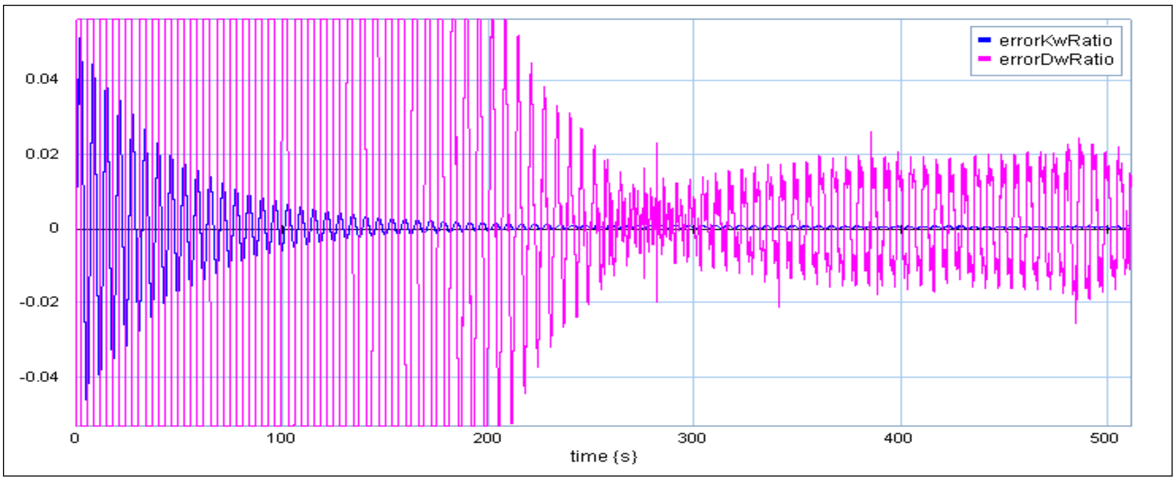


(b) Resulting estimate of D_w

Figure 3: Estimates using the update law in continuous time



(a) Ratio of the error in K_w divided by the real value of K_w



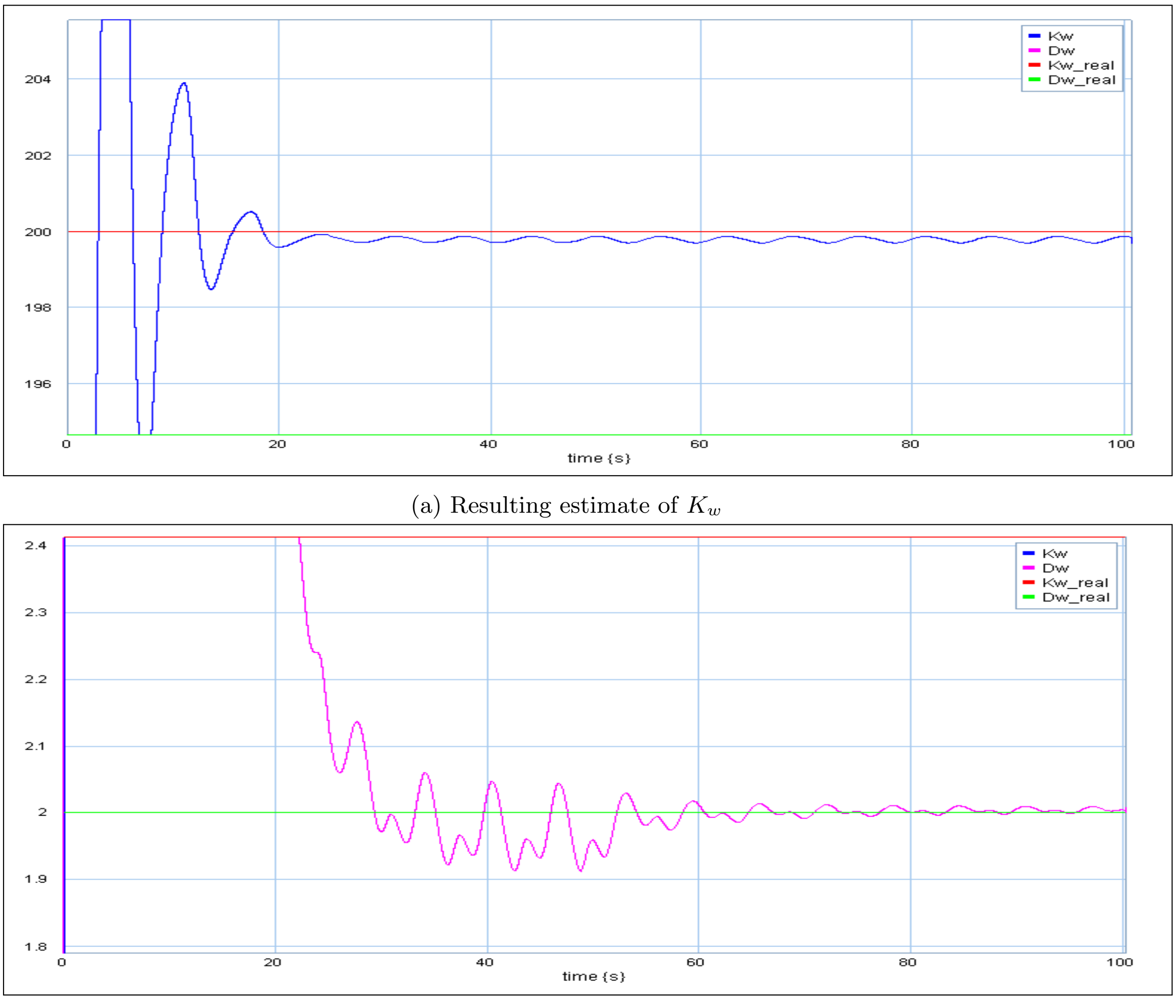
(b) Ratio of the error in D_w divided by the real value of D_w

Figure 4: Ratio of the errors using the update law in continuous time

As is visible the algorithm closely estimates the true values of the stiffness and damping of the environment respectively. It should be noted that the simulations were focused on obtaining correct estimates and not do it in the least amount of time possible. The fact that certain damping parameters of the model have been removed to obtain a more ideal version also causes the system to oscillate longer and have a long settling time. From the figures it is clear the estimates converge to within a very small error.

Extended kalman filter The same has been done for the extended kalman filter separately from the update law. The only difference is that the system has been discretised since a discrete EKF has been implemented. The results are shown in figures 5a, 5b and 6.

The EKF clearly delivers good results as well. The error in the stiffness is higher but still very low and acceptable. The damping however gives a better result since it is less noisy. This is probably due to the fact that the differentiators necessary for the update law to generate $\ddot{r}$ and $\dddot{r}$ are not needed in this method.



(a) Resulting estimate of K_w

(b) Resulting estimate of D_w

Figure 5: Estimates using the discrete EKF

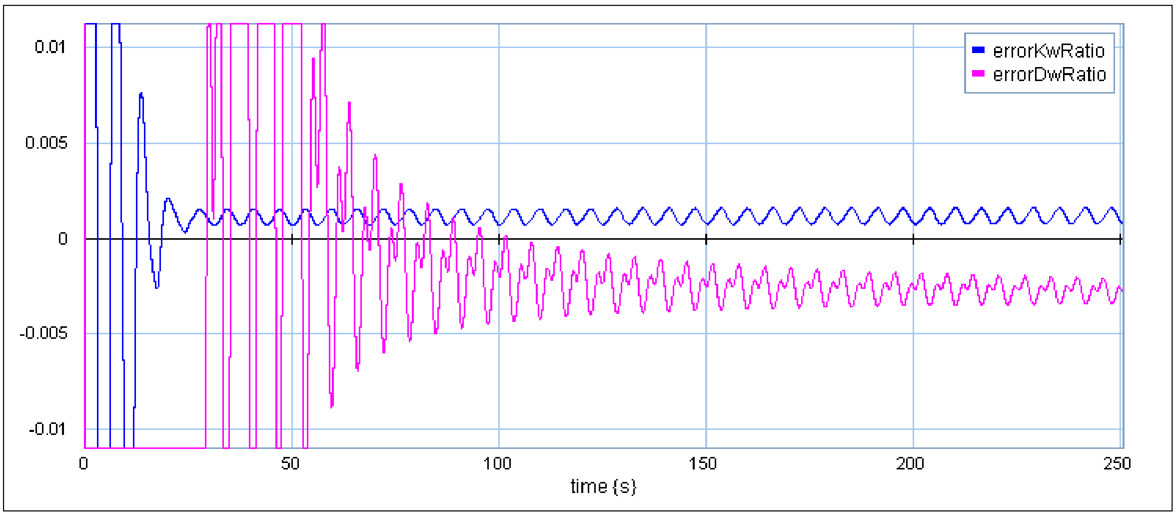
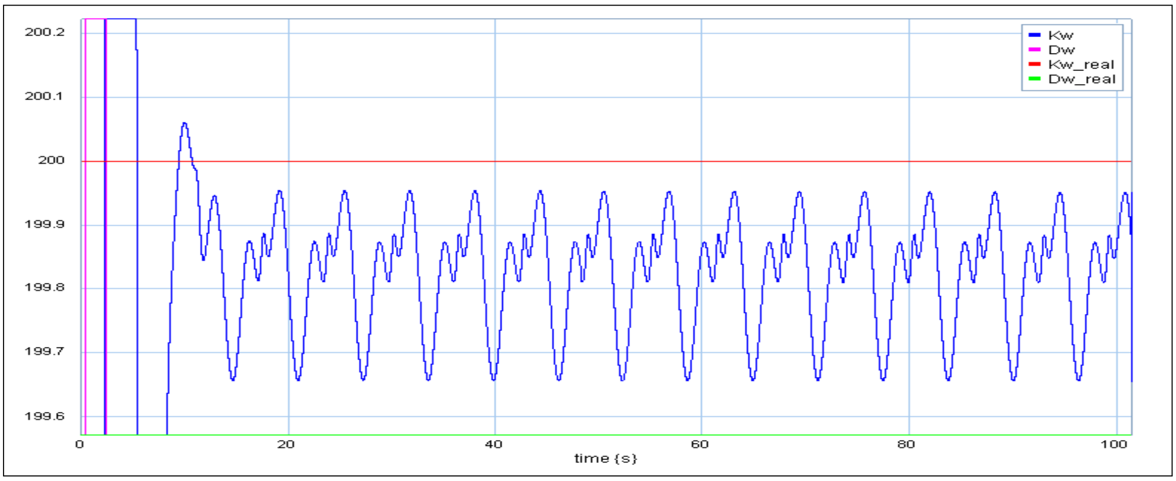


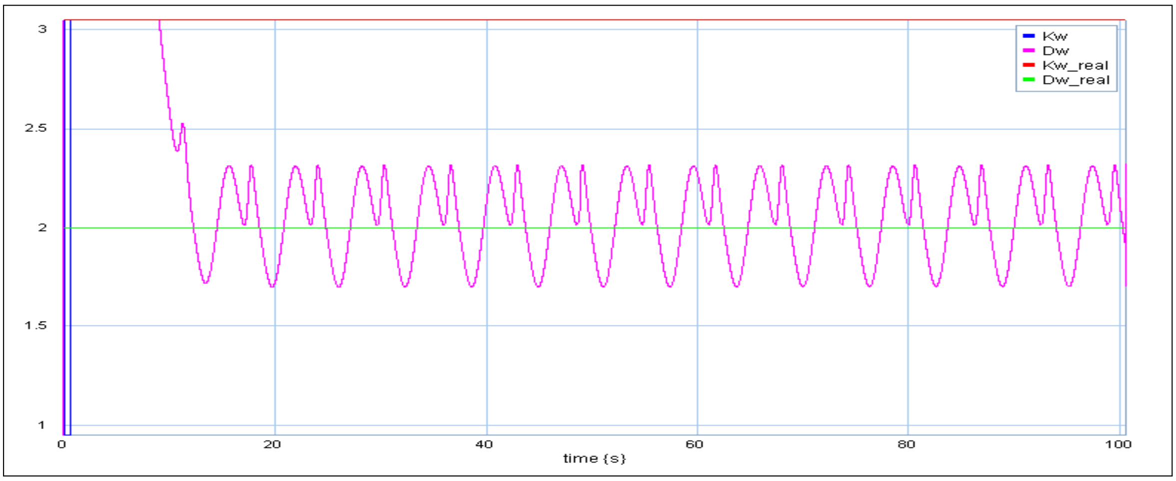
Figure 6: Ratio of the errors divided by the real values using the discrete EKF

The EKF using the update law Next the combination of the update law and the EKF is tested. The result is shown in figures 7a, 7b, 8a and 8b.

The most pronounced distinction is in the damping estimate which has become worse, it oscillates around the correct value with amplitudes of up to 15% off. The reason why the percentage is so high is partly because the damping that needs to be estimated is very low, hence a small error gives a big percentage error. When using the algorithm on a wall with higher damping the ratio becomes closer to zero, see figure 9. The stiffness estimate is still very good.

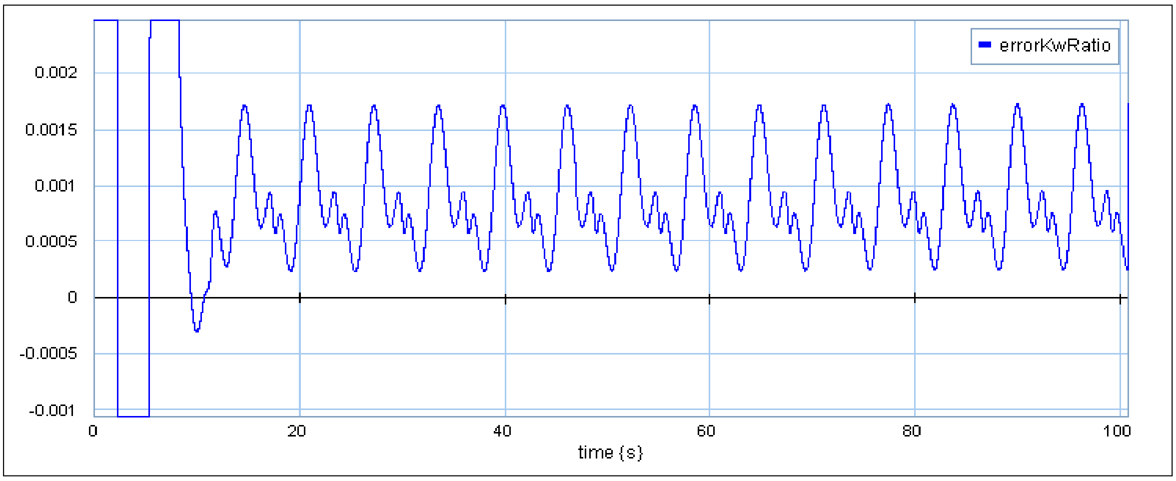


(a) Resulting estimate of K_w

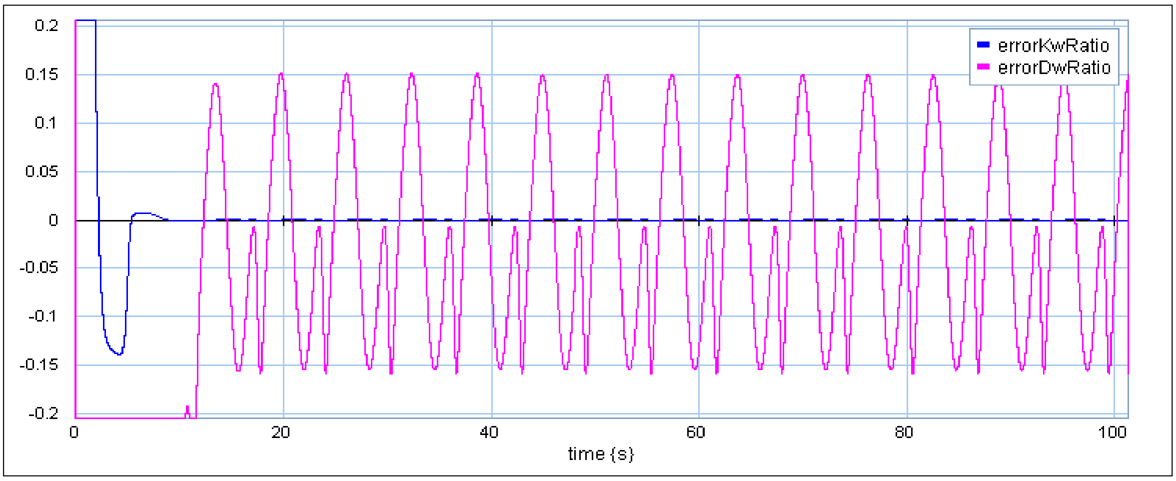


(b) Resulting estimate of D_w

Figure 7: Estimates using the discrete EKF in combination with the update law



(a) Ratio of the error in K_w divided by the real value of K_w



(b) Ratio of the error in D_w divided by the real value of D_w

Figure 8: Ratio of the errors using the discrete EKF in combination with the update law

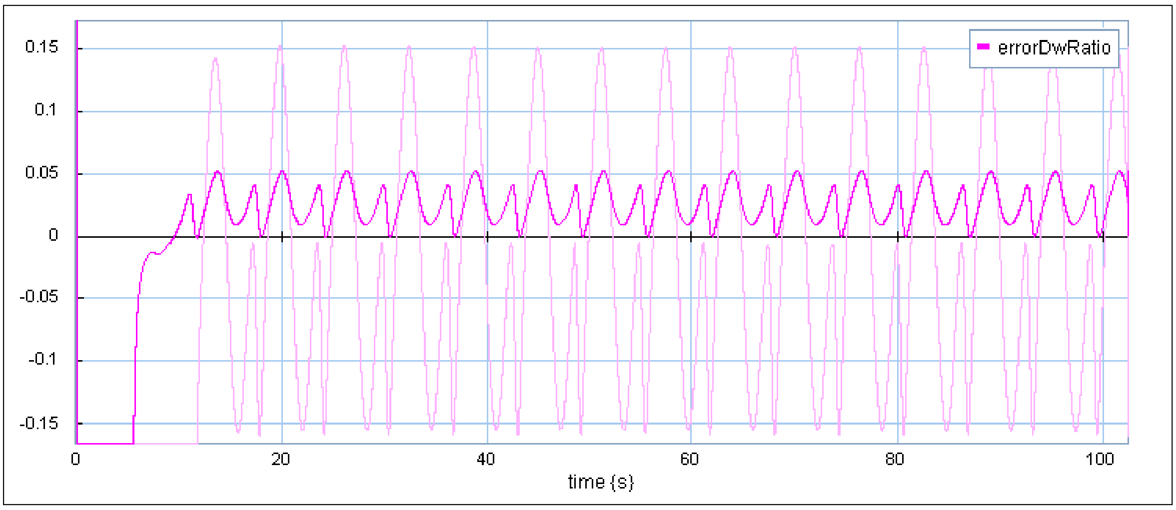


Figure 9: Ratio of the error in D_w divided by the real value of D_w with $D_w = 50$

6.2 Unwanted Effects

In this subsection some device specific unwanted behaviour will be mentioned and explained. For this the model using the EKF will be used, however the effects described are the same for the other estimation algorithms.

Belt stiffness The driving belt of the vsaUT-II has a limited stiffness which severely affects the performance of the system in general. The stiffness of the driving belt has been characterised by Stefan Groothuis to be $K_b = 532361.7N/m$, because of the mechanical configuration there is a transmission ratio which determines the effective $K_{b_{eff}}$ at the output:

$$K_{b_{eff}} = K_b * 0.041^2 = 894.9Nm/rad \tag{6.1}$$

The VSA is able to change its desired stiffness from practically zero (0.7) to infinitely stiff if it was not for the belt stiffness. The belt stiffness is effectively a spring placed in series with the output stiffness, see figure 10, lowering the apparent output stiffness of the entire VSA. When transforming figure 10 to a bondgraph representation and doing some calculations the following effective stiffness and damping are obtained:

$$K_{vsa_{eff}} = \frac{K_b}{D_{vsa}s + K_{vsa} + K_b} K_{vsa} \tag{6.2}$$

$$D_{vsa_{eff}} = \frac{K_b}{D_{vsa}s + K_{vsa} + K_b} D_{vsa} \tag{6.3}$$

The s is a Laplace operator and shows that the damping term is frequency dependent. However D_{vsa} is very small, $0.012Nms/rad$, and the motions are slow, hence this effect is ignored for this case.

The effect is visualised in figure 11.

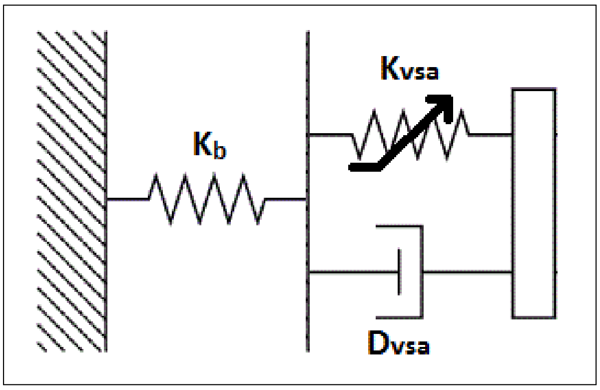


Figure 10: Simplified situation sketch of the driving belt stiffness

The effective K_{vsa} and D_{vsa} can thus be calculated according to:

$$K_{vsa_{eff}} = \frac{K_b}{K_{vsa} + K_b} K_{vsa} Nm/rad \tag{6.4}$$

$$D_{vsa_{eff}} = \frac{K_b}{K_{vsa} + K_b} D_{vsa} Nms/rad \tag{6.5}$$

$K_{vsa_{eff}}$ can never exceed the limit of 894.9. Without anticipation the estimates of K_w and D_w will become a factor $\frac{K_{vsa}+K_b}{K_b}$ too high. By using $K_{vsa_{eff}}$, see eq. 6.4, instead of the theoretical value of K_{vsa} the correct estimates can be re-obtained, see figure 12 and 13. From the figures it is visible that the estimates are restored to their correct values.

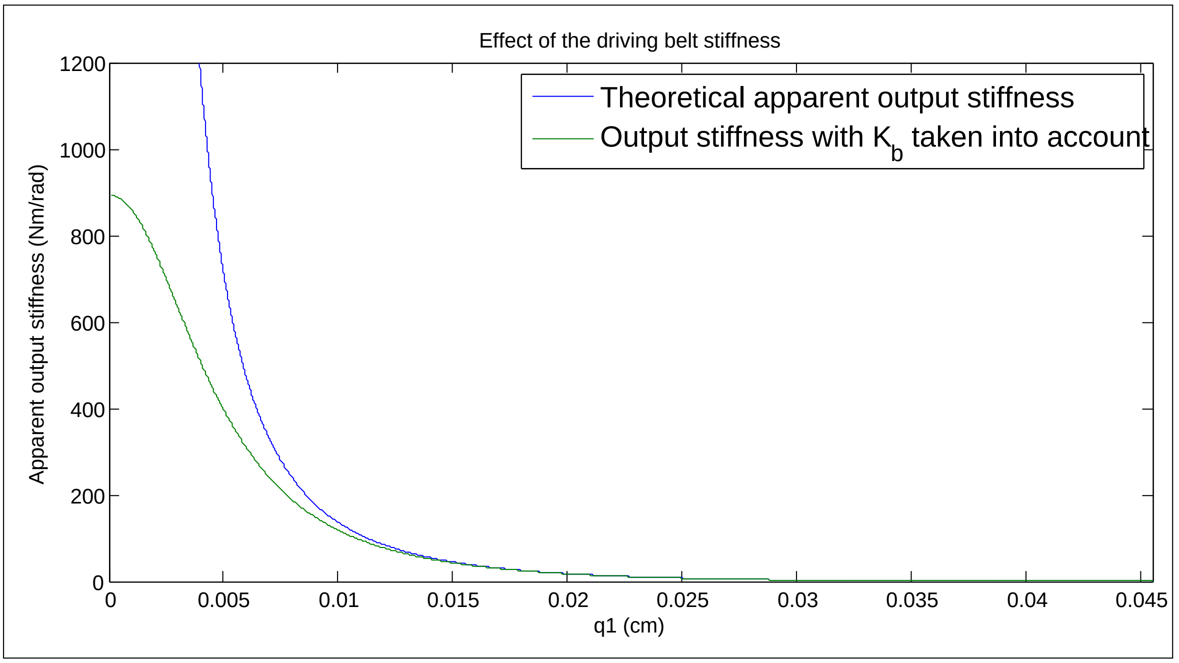


Figure 11: The effect of the driving belt stiffness on the theoretical apparent output stiffness of the VSA

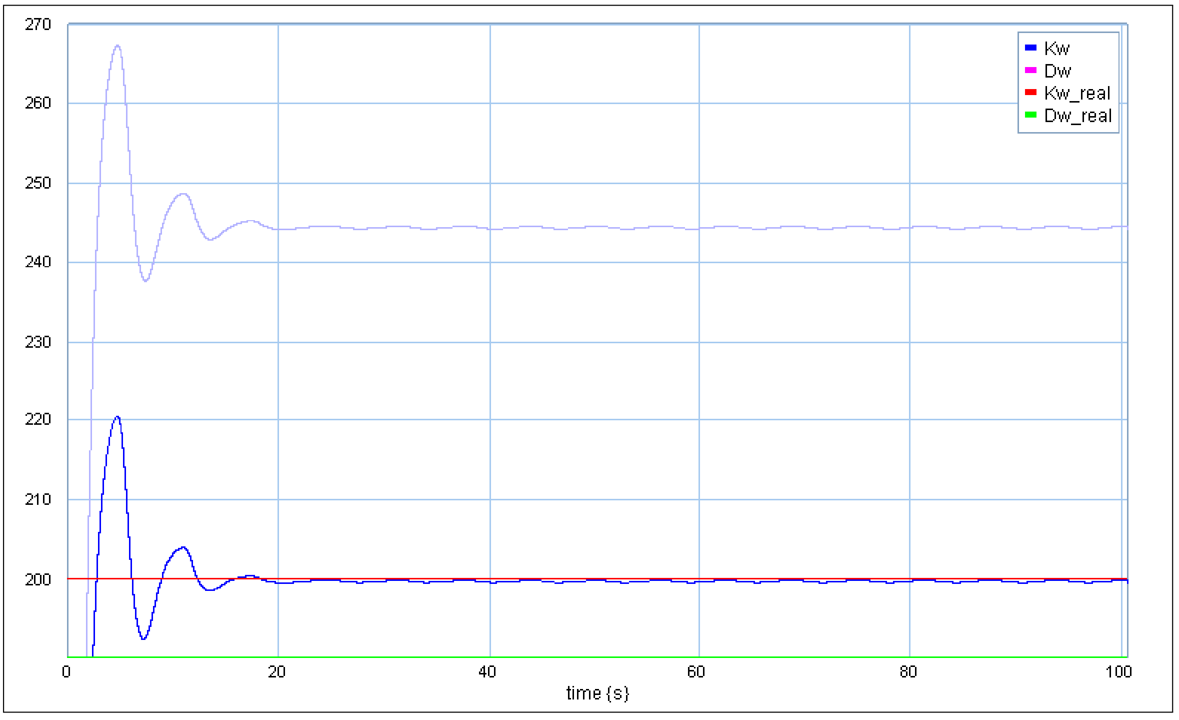


Figure 12: The effect of the driving belt stiffness on the estimated K_w when not anticipated (light line) and when anticipated (dark line)

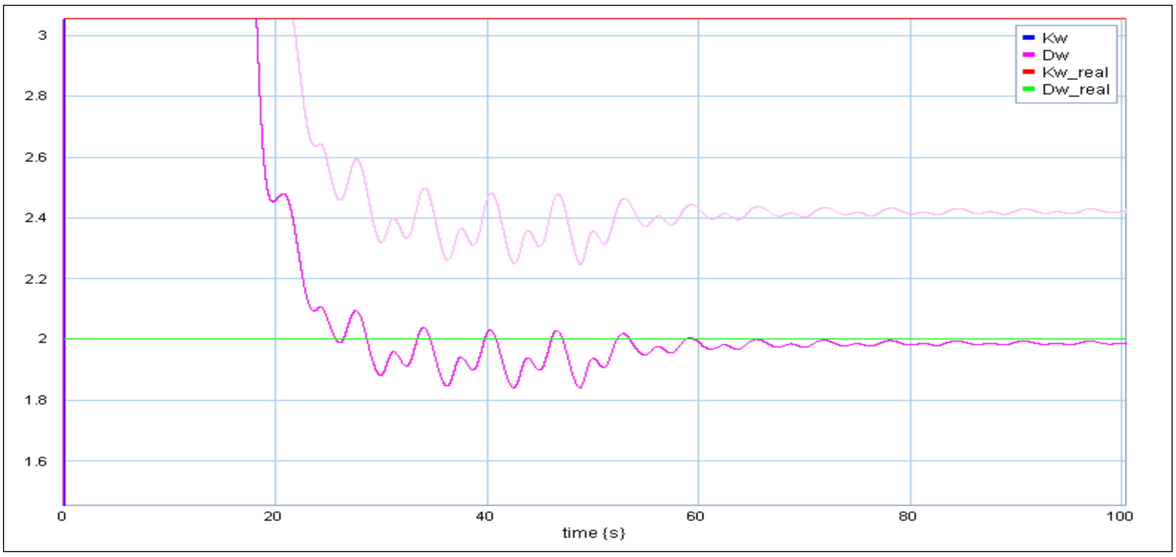


Figure 13: The effect of the driving belt stiffness on the estimated D_w when not anticipated (light line) and when anticipated (dark line)

Unwanted dissipations In the model of the system there are two specific dampings that influence the final estimate of D_w, they have a small influence on the estimate of K_w. The dissipations in question are the dissipation in the rotating part of the actuated system and the dissipation due to the driving belt. The following figures show their effect when they are added to the model again to better represent the true system, see figures 14, 15a, 15b and 15c.

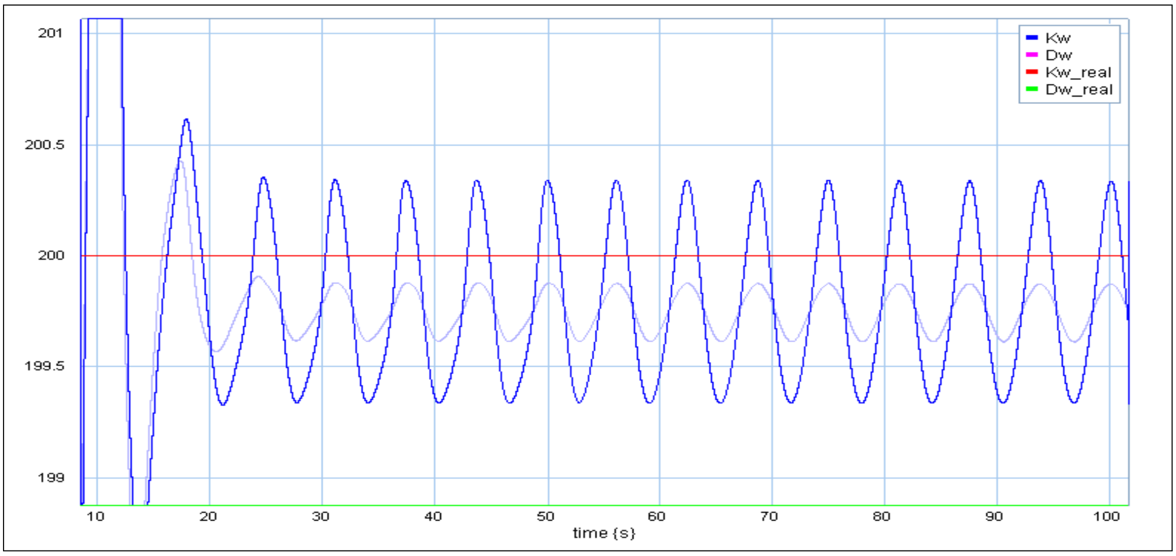
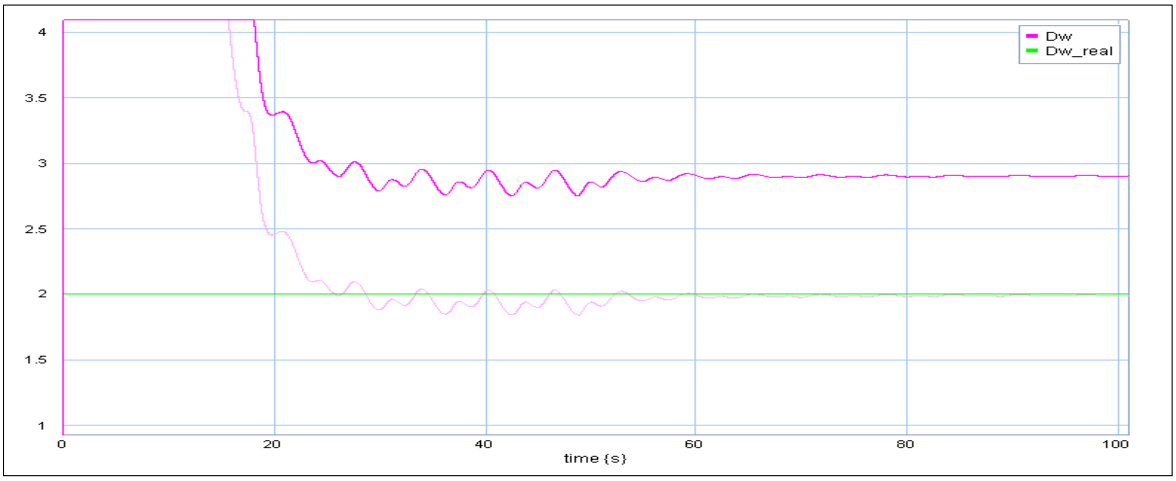
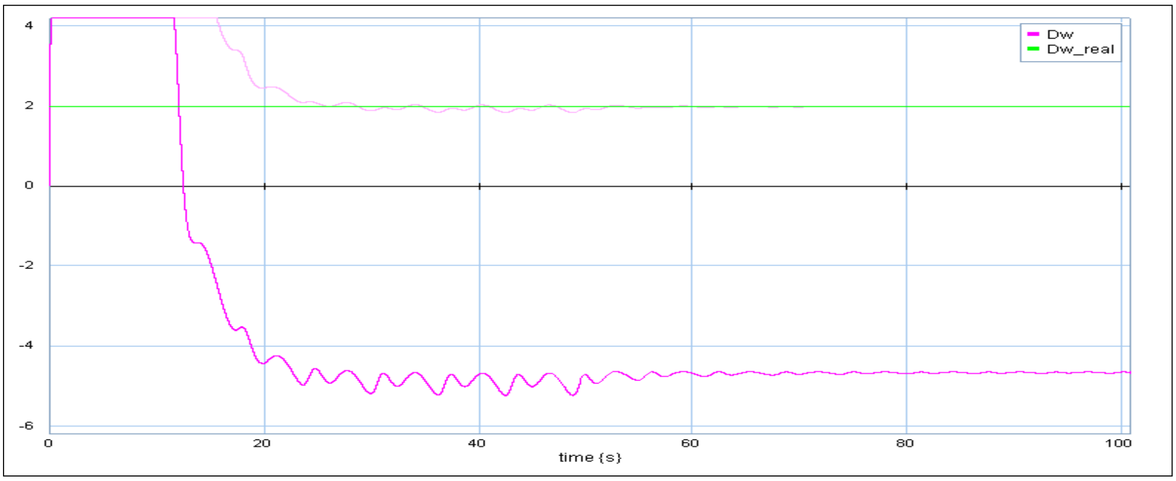


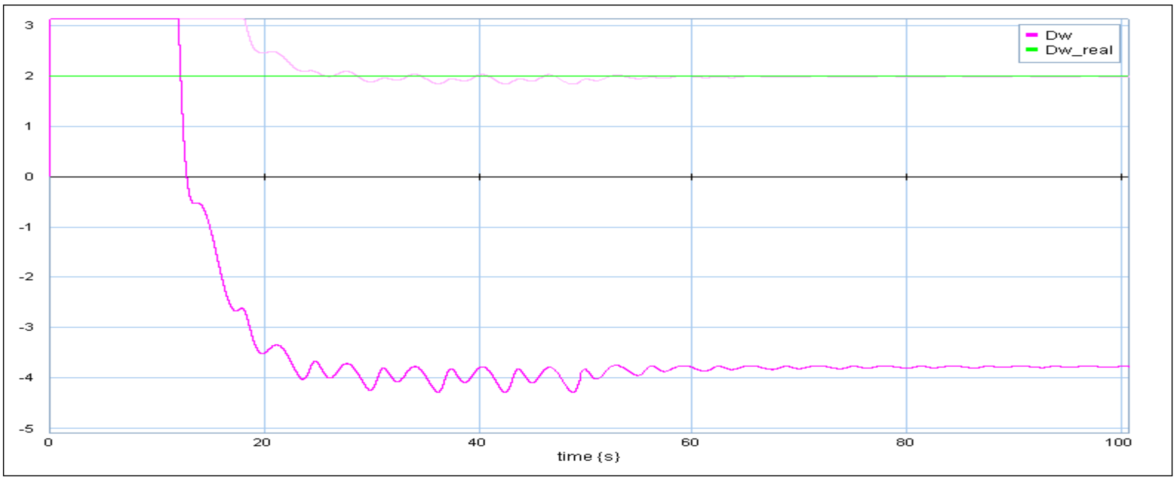
Figure 14: The effect of both dampings on the estimated K_w when present (dark line) and when not present (light line)



(a) The effect of the damping of the actuated device on the estimated D_w when present (dark line) and when not present (light line)



(b) The effect of the damping of the driving belt on the estimated D_w when present (dark line) and when not present (light line)



(c) The effect of both dampings on the estimated D_w when present (dark line) and when not present (light line)

Figure 15: The effect of the dampings on the estimate of D_w

Figure 14 shows that the oscillations on the steady state value of K_w become higher when the dissipations are present. However the total amplitude of these oscillations stays clearly within acceptable boundaries. For the damping estimate the effects are much more detrimental and the estimate becomes too low. This problem is the biggest when estimating low damping factors as is the case in this situation. In this particular case with a true D_w of 2Nms/rad, when both dissipations are put inside the model, the estimated damping becomes negative which is physically not possible. If the damping would be higher, the absolute error in the damping estimate will almost not change. With a damping of $D_w = 500$ for example the estimate stabilises at approximately 490, hence the error expressed in percentages is much lower.

It is much harder to anticipate the effects of the unwanted damping in the system since the formulas involved are frequency dependent. For example in the simplified case with only damping added in the belt, see figure 16, the following effective stiffness and damping at the output are obtained:

$$K_{vsa_{eff}} = \frac{R_b s + K_b}{(D_{vsa} + R_b)s + K_{vsa} + K_b} K_{vsa} \tag{6.6}$$

$$D_{vsa_{eff}} = \frac{R_b s + K_b}{(D_{vsa} + R_b)s + K_{vsa} + K_b} D_{vsa} \tag{6.7}$$

If the frequency is zero, thus in the static case, the damping parts drop out and the same anticipation method as shown earlier remains. At the moment there is no anticipation for the unwanted dampings implemented yet.

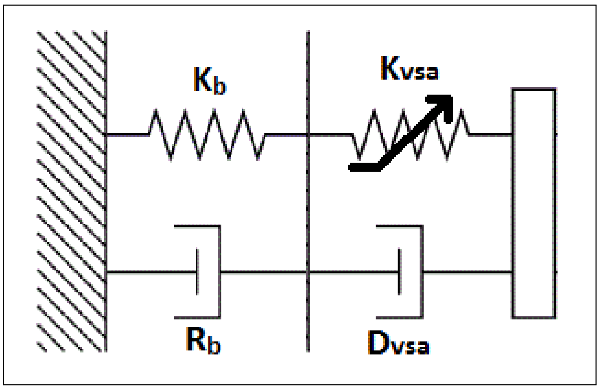


Figure 16: Simplified situation sketch of the driving belt stiffness and damping

Due to the frequency dependency it is expected the estimated value will change according to the frequency with which the VSA is moved. The movement of the VSA is done by changing the equilibrium position of the output with a sinusoidal motion $A + Bsine(F_m time)$. Figure 17 shows the results for $F_m = 0.5, 1, 2, 5$. The faster the VSA moves the worse the estimate becomes. Also the damping of the wall has been increased to 100 in figure 17 to show it does not become negative then.

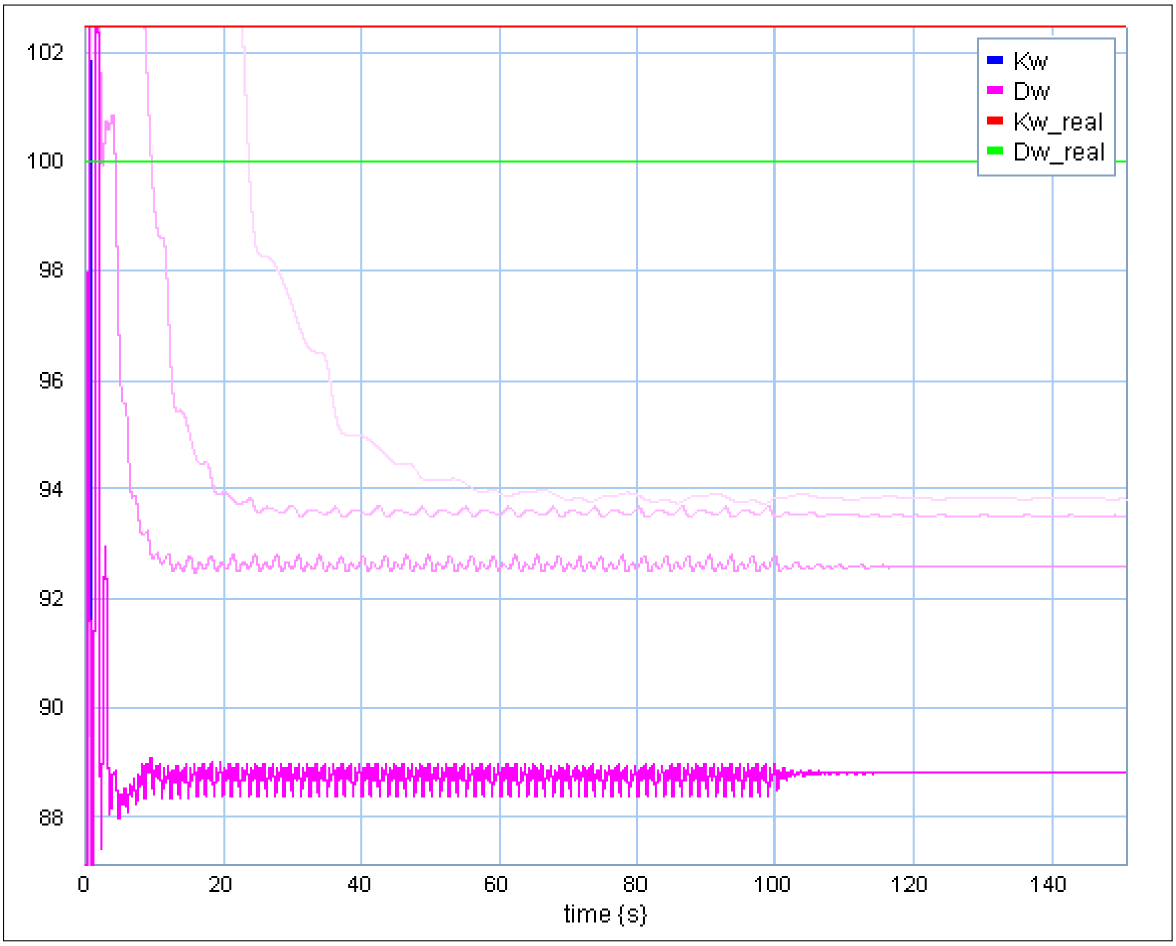
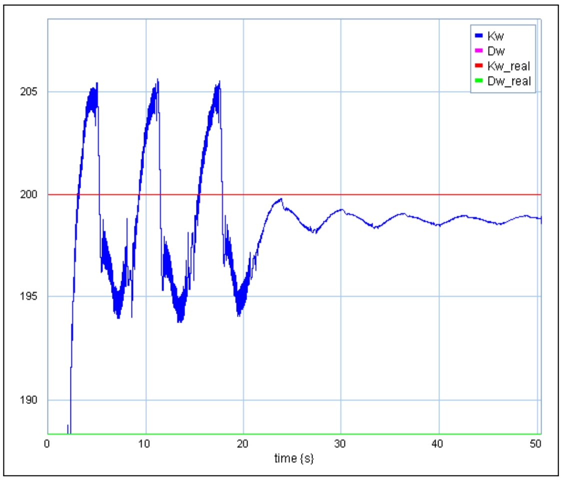


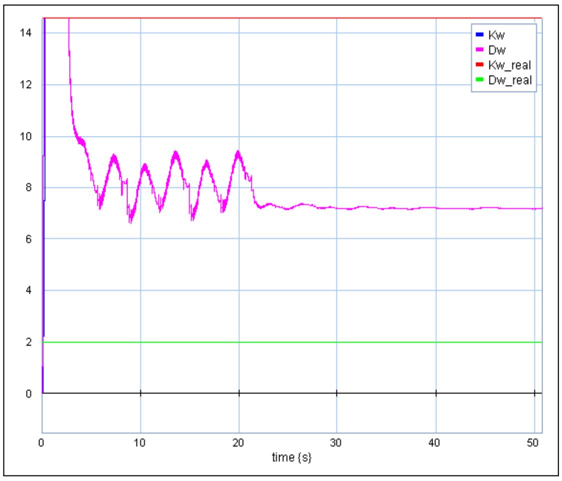
Figure 17: The effect of the movement frequency on the damping estimate, thicker lines coincide with faster movements

6.3 Simulations Based on the True Model of the VSA

As preparation for the real experiments, simulations have been done using a model as closely resembling the real setup as possible. This means the limited stiffness of the driving belt is present (which is compensated in the algorithm) and the two dissipations as well. Furthermore there is no more infinite accuracy, the encoders are placed inside the model. Their specifications are repeated for clarity. The encoder on the output of the VSA is a 10bit absolute magnetic encoder and the encoders on the motors that handle the degrees of freedom q_1 and q_2 have an accuracy of 2000ppr. The following results were obtained for the EKF and the combination of the EKF with the update law, see figures 18a, 18b, 19a, 19b, 20 and 21. The clear decline in the amplitude of the oscillations on the steady state value that is sometimes visible in the figures is due to the fact that some parameters in the EKF are time dependent. The time dependence is chosen such that in the beginning the estimates are sensitive to change and thus converge fast to their approximate final value. Later in time the values are changed such that the estimates are less sensitive to change, causing there to be less noise and oscillations on the steady state value. It follows from the figures that the discretisation of the update law has a detrimental effect on the estimates, this is especially visible in the noisy damping estimate. To be able to successfully implement the update law a better resolution is necessary with less noise due to the derivatives. The extend to which the update law affects the estimates can be regulated with the design parameter α.

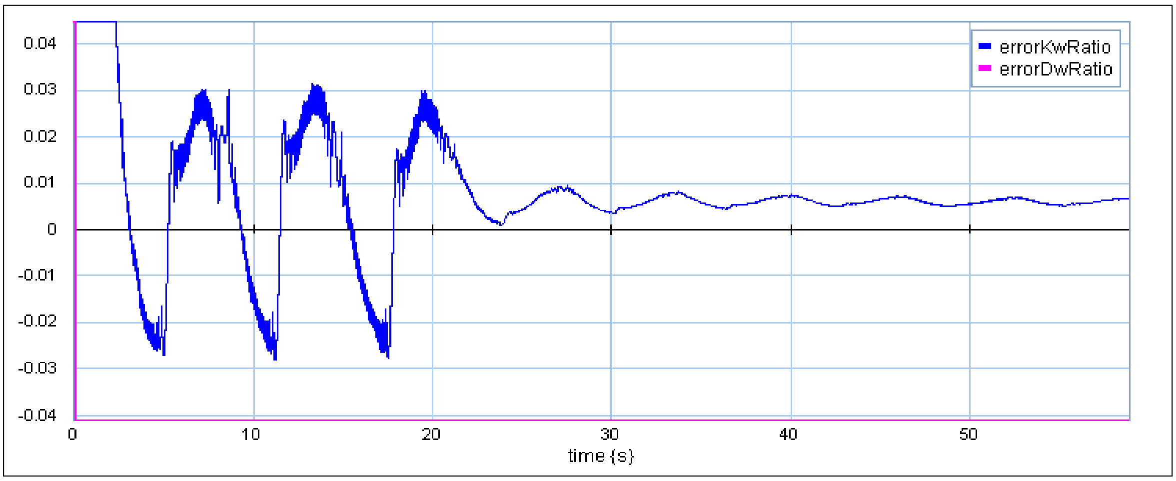


(a) Result of the K_w estimate

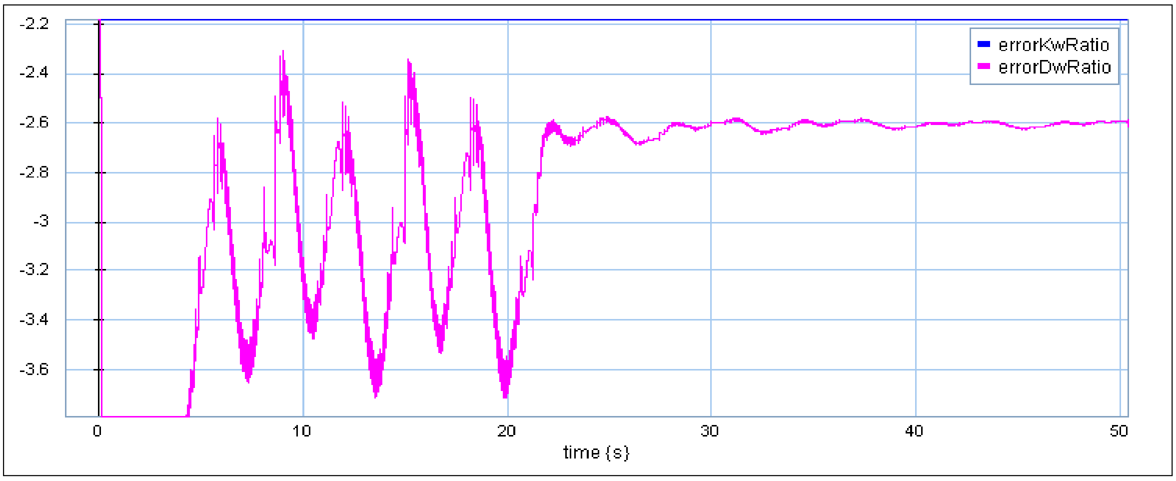


(b) Result of the D_w estimate

Figure 18: Estimates using the EKF



(a) Ratio of the error in the K_w estimate divided by the true value



(b) Ratio of the error in the D_w estimate divided by the true value

Figure 19: Ratio of the errors using the EKF

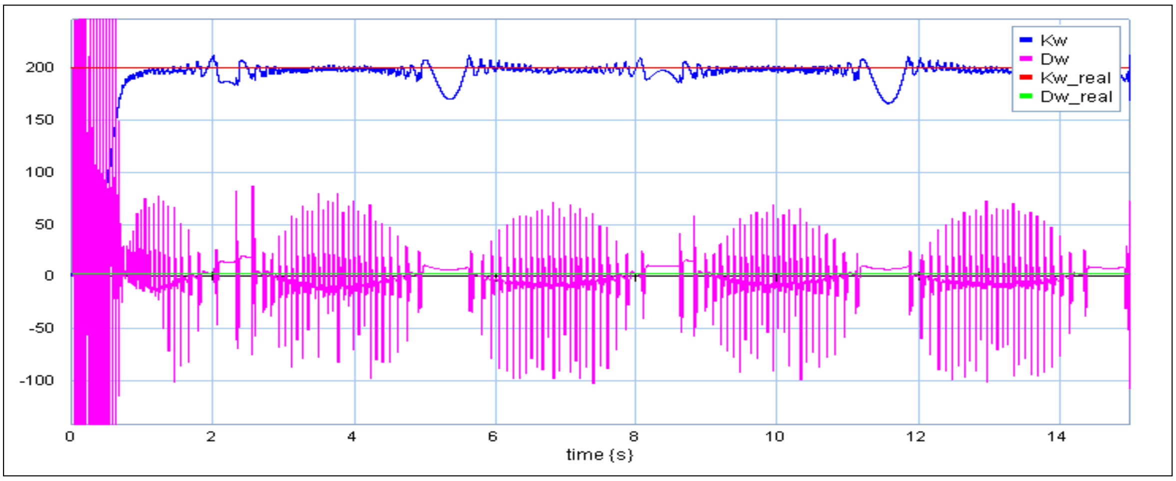


Figure 20: Result of the estimates using the EKF with the update law

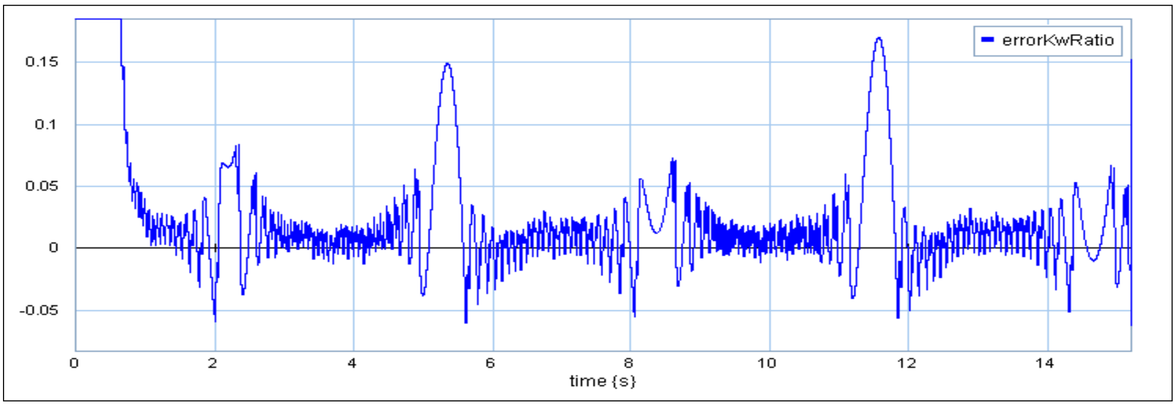


Figure 21: Ratio of the error in the K_w estimate divided by the true value using the EKF with the update law

6.4 Accuracy Effect

During the simulations it was noticed that even when the stiffness of the belt was made infinitely stiff in the model and the unwanted dampings were removed, the estimate of the damping D_w was still erroneous, see figure 22. Because of this multiple simulations have been done to test the effect of the accuracy present in the system, meaning the accuracy of the encoder on the output, the encoders on the internal degrees of freedom and the used sampling frequency for the discrete system. The encoder accuracy of the internal degrees of freedom did not have any significant effect but the encoder accuracy on the output does have a strong influence and the sampling frequency has a significant effect as well, although less significant than the encoder on the output. By enhancing the encoder to a 14bit encoder and doubling the sampling frequency to 400Hz, a much better estimate can be generated, see figure 23. The higher the encoder accuracy the better, see figure 24. This means that on the current setup, even when an anticipation method is designed and implemented that can correctly anticipate the effect of the dampings, it is not possible to obtain a correct estimate of D_w. For this to be possible the accuracy needs to be increased.

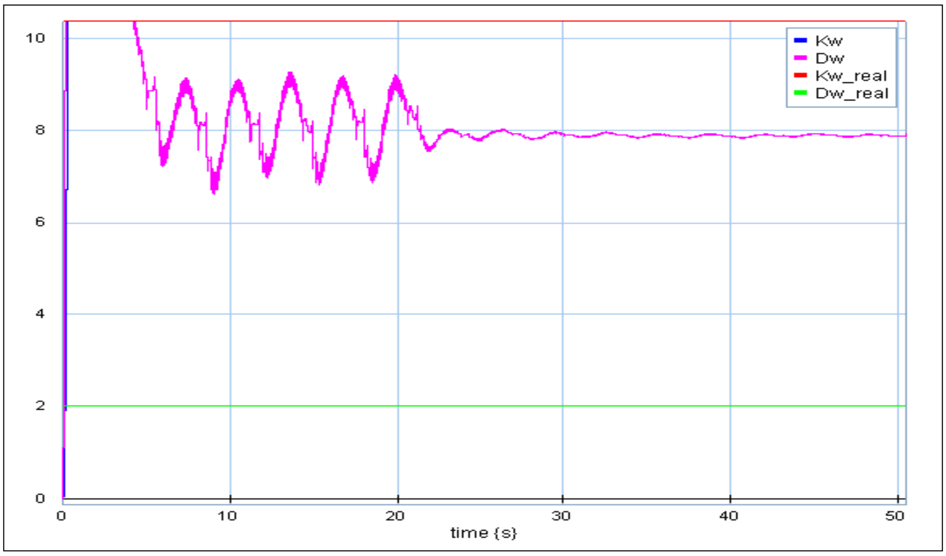


Figure 22: D_w estimate using the EKF with an infinite stiff driving belt and without the two unwanted dampings

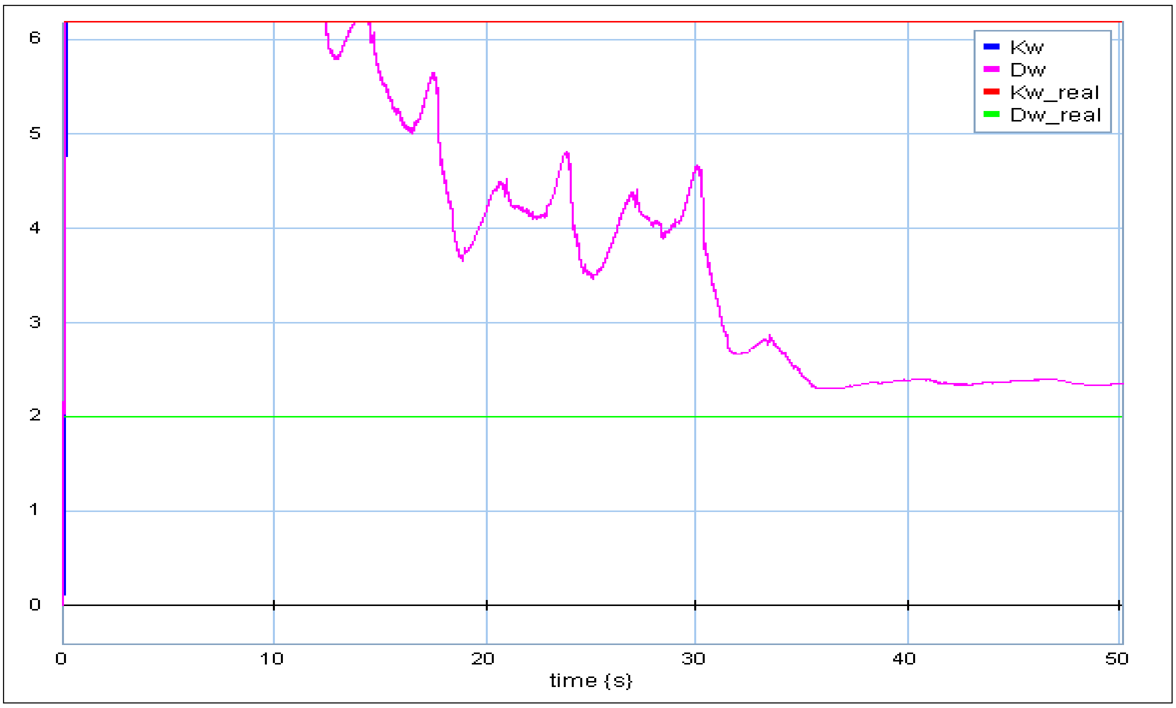


Figure 23: D_w estimate using the EKF, with a 14bit encoder on the output and a sampling frequency of 400Hz

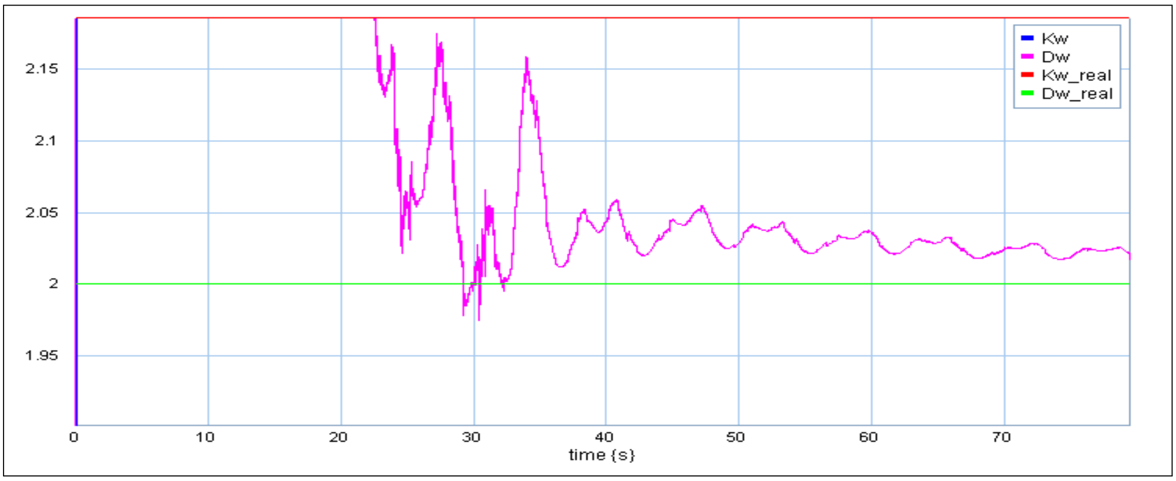
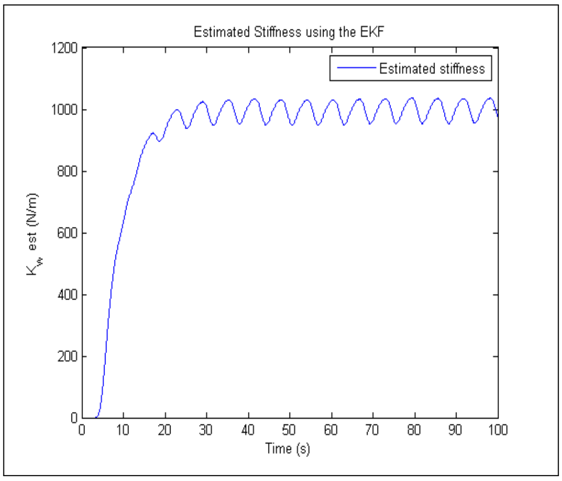


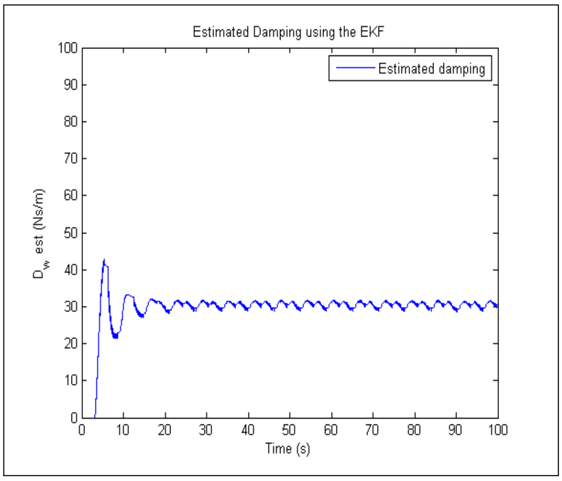
Figure 24: D_w estimate using the EKF, with a 16bit encoder on the output and a sampling frequency of 400Hz

7 Results

In this chapter some results obtained from experiments on the real setup will be shown. During the experiments an environment with a K_w of 1000N/m is used. First the results will be shown of the estimation algorithm using only the EKF without the update law, see figures 25a, 25b and 26.



(a) Estimated Stiffness



(b) Estimated Damping

Figure 25: Estimates using the EKF

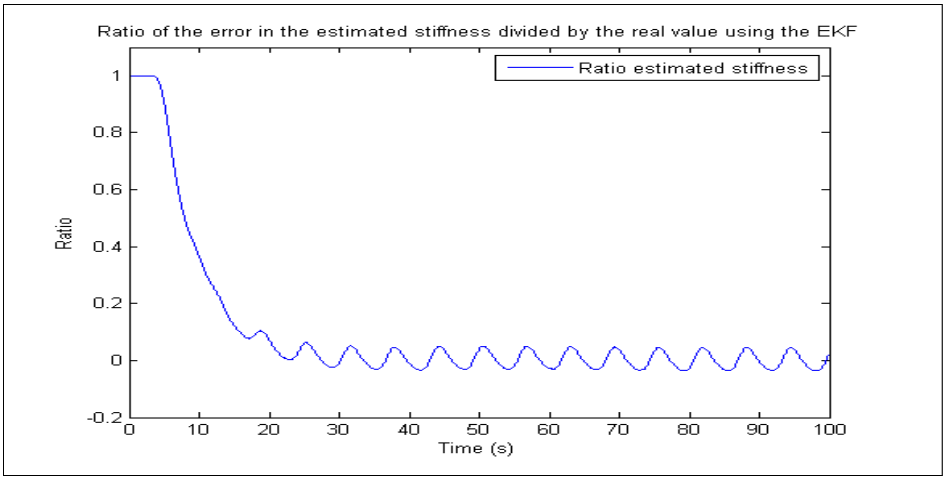
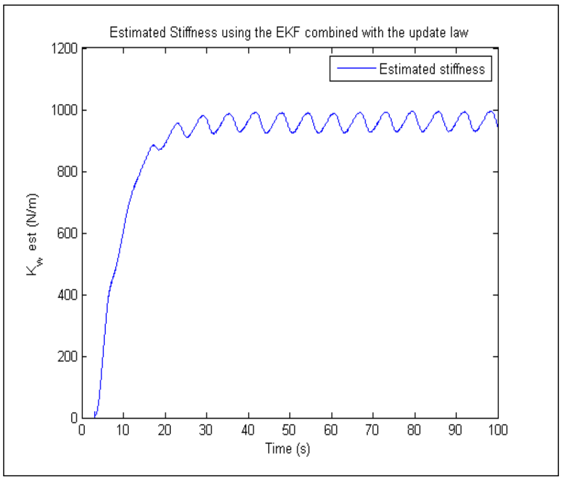


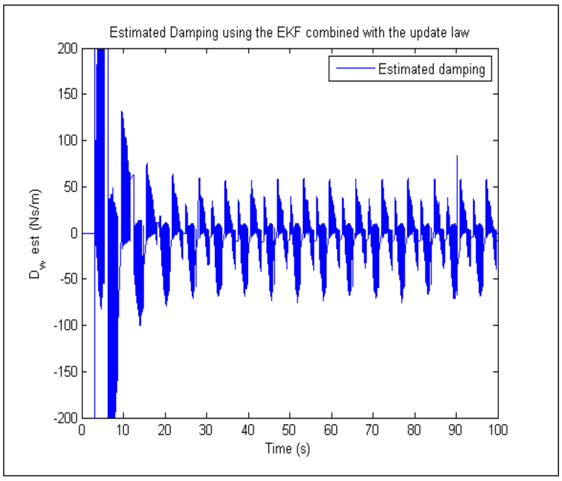
Figure 26: Ratio of the error in the estimated stiffness divided by the real value using the EKF

From the results it is clear that the stiffness converges to approximately the correct value. The VSA is in constant excitation by changing the equilibrium position of the output. The oscillations on the the steady state have the same base frequency as the movement of the equilibrium position. The damping however does not converge to a value close to 2 but is much higher, see figure 25b. This behaviour is consistent with what has been observed in the simulations when the accuracy of the simulated system was brought down to the accuracy of the real setup. Hence this is a good indication that the model used is of good quality and capable.

The results using the EKF in combination with the update law are shown in figures 27a, 27b and 28. The estimate of the stiffness is in both cases very similar, however in the damping estimate a big change is visible. Due to noise introduced by the differentiators of the update law the damping estimate which is more sensitive to higher dynamics becomes noisy as well. The amount of noise or the sensitivity to the noise can be configured by changing the design parameter α. Although the update law adds noise to the system it is still a valuable addition since it brings an extra design parameter. α could for example first be chosen high for quicker convergence and then set low or even to zero, canceling out the update law entirely, such that the steady state value is more stable.



(a) Estimated Stiffness



(b) Estimated Damping

Figure 27: Estimates using the EKF in combination with the update law

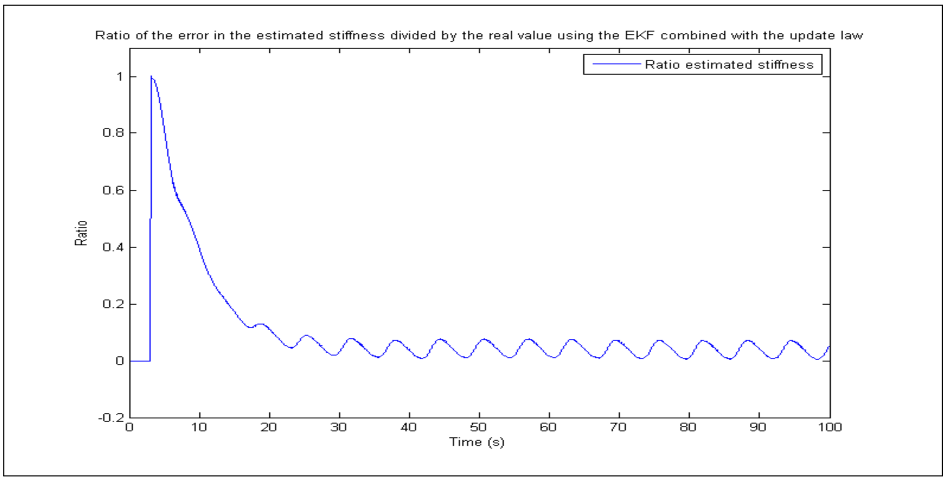
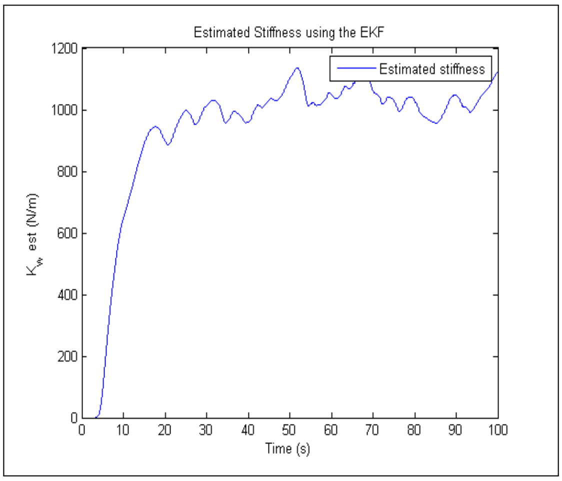
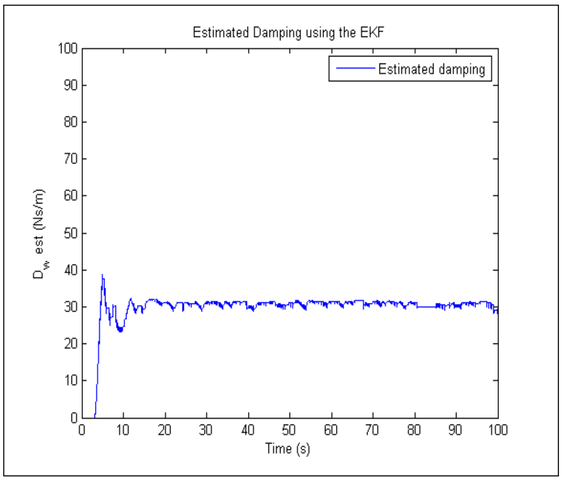


Figure 28: Ratio of the error in the estimated stiffness divided by the real value using the EKF in combination with the update law

Since the stiffness estimates are comparable but the damping estimate of the algorithm using only the EKF is more stable, and thus more conclusions can be drawn from that data, in the next figures the estimation algorithm using only the EKF will be used. Up till now the experiments were done by keeping the system in constant motion by applying a sinusoidal motion to q_2. In real life applications it is not realistic to expect the same kind of motion. To test the workings of the estimation algorithm in combination with the VSA under more realistic circumstances, tests have been done using a random motion source. Care is taken that the motion of the VSA stays within its working area. The results are shown in figures 29a, 29b and 30. The damping estimate shows no significant change with the case where a sinusoidal motion profile was used. In the stiffness case a difference is visible. It shows that the estimate is to a certain extend dependent on the movement of the VSA, this dependence can be reduced by changing the parameters in the algorithm over time to make the estimate less sensitive when time progresses.



(a) Estimated Stiffness



(b) Estimated Damping

Figure 29: Estimates using the EKF, random motion profile

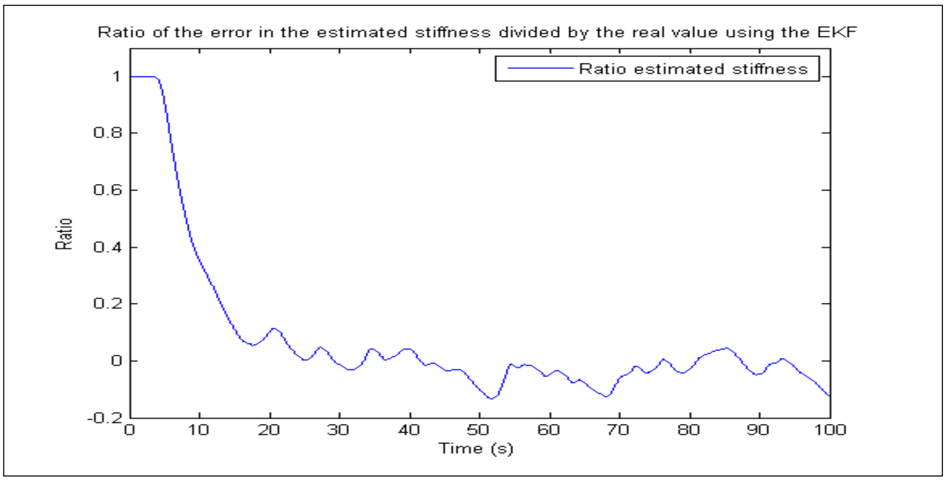
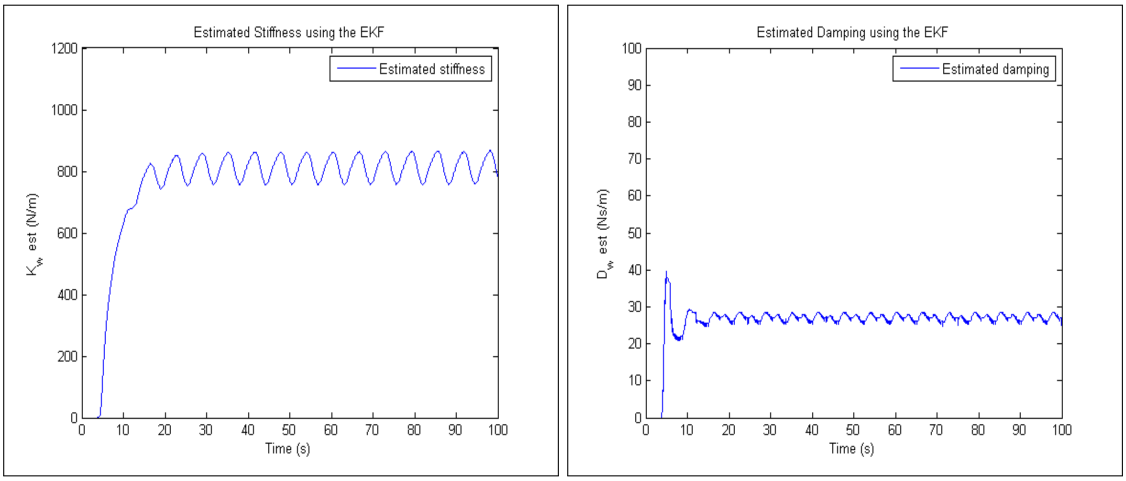


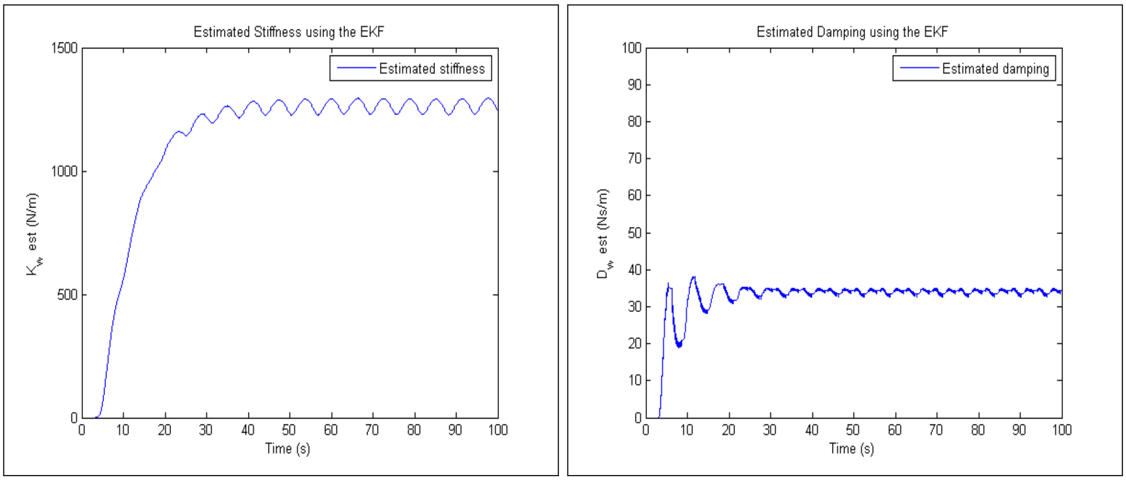
Figure 30: Ratio of the error in the estimated stiffness divided by the real value using the EKF, random motion profile

During the experiments on the real setup a serious problem was encountered which was not present in the simulation. The apparent output stiffness of the VSA seems to have an effect on the steady state value of the estimates done by the estimation algorithm. The previous experiments were done with a K_{vsa} of 400Nm/rad, this value was chosen since it lies approximately in the middle of the working range of the VSA (0.7-900Nm/rad). Results of the estimates with different used K_{vsa} stiffnesses can be found in figures 31a, 31b, 32a and 32b.



(a) Estimated Stiffness (b) Estimated Damping

Figure 31: Estimates using the EKF, $K_{vsa} = 200Nm/rad$



(a) Estimated Stiffness (b) Estimated Damping

Figure 32: Estimates using the EKF, $K_{vsa} = 800Nm/rad$

When the K_{vsa} becomes higher, the estimate of K_w also becomes higher, when it becomes lower the estimate becomes lower as well. For the damping estimate D_w this seems to hold as well but to a lesser extent. However since the damping estimate is erroneous anyway no real conclusions can be drawn from that observation. Since this effect did not occur in the simulations, it is likely that there is a discrepancy between the model and the real system behaviour. No solution to this problem has been found yet.

The real setup suffers from a backlash introduced by the connection between the input pulley and the gearbox. This means that whenever a certain external torque is applied on the output of the VSA, the connection slips briefly before engaging again. This causes the apparent output stiffness of the VSA to be too low. This behaviour is not modelled in the simulations and might be the cause or at least part of the cause why the estimates are dependent on the stiffness of the VSA. It must be said that the observed backlash is an imperfection of the realisation of the VSA and it is not an inherent error in the concept of the vsaUT-II.

8 Conclusion

The goal of this MSc project was to show that a variable stiffness actuator can be used as a sensor to estimate parameters of an external environment with which the actuator is in contact in real time. That goal has been accomplished. This work combined the unique properties of a variable stiffness actuator with an estimation algorithm to obtain this goal and estimate the stiffness and damping factors of a 'wall'.

The proof of concept has first been obtained using simulations where full control of the situation is possible, meaning that if the VSA is seen as a black box with exactly known output parameters (apparent output stiffness, damping and inertia) the algorithm will estimate the correct values. The simulations however also show that undesired characteristics in a VSA, like parasitic stiffnesses and dissipation, will have a serious and detrimental effect on the estimation of the parameters of the environment. Because of this it is important to know these effects and have a good understanding of the actual system. These effects need to be anticipated and compensated before feeding the values of the output signals of the VSA to the estimation algorithm to obtain correct estimates.

The update law that has been combined with the EKF is due to its dependence on the multiple derivatives of signals sensitive to noise. The bandwidth of the derivatives can be adjusted to suppress the noise but this introduces a delay in the signals which has a detrimental effect on the damping estimate. By combining the update law with the EKF an extra design parameter α is introduced which can be used to speed up the convergence time which makes the update law still a valuable addition. When α is set to zero the update law will not have any effect.

Furthermore it is concluded that estimating the damping factor of an environment is harder than estimating the stiffness parameter. The damping estimate is more sensitive to errors, the limited accuracy of encoders, noise and other unwanted or unknown dynamics in the system. According to the simulations done it is shown that on the specific device that has been used for the real experiments it is as of yet not possible to obtain a correct damping estimate with this algorithm due to the limited accuracy of the sensors in combination with the height of the sampling frequency. These limitations caused by the accuracy have been confirmed by experiments on the real setup.

The estimation algorithm is to a certain extent dependent on the excitation of the VSA, the frequency of the motion of the VSA can be found back in the oscillations of the steady state of the estimation. Also on the real setup the estimation depends on the apparent output stiffness of the VSA. Since this is not the case in the simulations it is assumed there is still some discrepancy between the model and the real setup. One effect that occurs on the real setup and is not modelled in the simulation is a backlash effect when a certain torque at the output is applied. This might be (part of) the cause of the unexpected behaviour.

9 Recommendations

For future work the following topics should be considered. This section is seperated in two parts, first topics are mentioned that should get precedence over other future work. In the second part topics are mentioned that would be interesting to have a look at as well.

9.1 Strong Recommendations

Higher accuracy From the simulations it is obvious that better accuracy in the used encoder on the output will result in far better estimates, especially with regards to the damping. A higher sampling frequency on which the estimation algorithm runs will also result in better estimates. There is also an encoder available, from the same manufacturer of which the current output encoder is used, which has 14 bits accuracy. Even though replacing the encoder might give extra work if the interface between the VSA and the chip needs to change, it is still recommended for when reliable damping estimates need to be obtained.

The sampling frequency and delay might be able to become better if the currently used Arduino board is replaced with something with more processing power. This would allow the sensing algorithm to be able to run on the board such that the need for extensive communication between Simulink on the computer and the VSA is kept to a minimum.

Dependency of the estimates on K_{vsa} The dependency of the estimates on the apparent output stiffness of the VSA when doing experiments on the real setup should be looked deeper into. If the VSA is to be reliably used as a VSS in a real application then this effect should first be solved. Otherwise only order of magnitude estimations can be done. It is recommended to first look into the backlash effect that happens when a certain external torque is applied at the output of the VSA.

Frequency dependent anticipation In the current system there are unwanted dissipations present that cause the damping estimate to be off. A way to anticipate their effect should be designed and implemented like has been done for the limited stiffness of the driving belt. This should take into account that due to the nature of dissipations this anticipation should be frequency dependent. The effects of the dissipations are only visible when the system is in motion.

9.2 Interesting Future Work

Robust derivatives The update law that is used in this report is sensitive to noise due to the derivatives it depends on. This is partly canceled out by the use of the EKF, however it would create a more robust system if the derivatives can be obtained in a more reliable manner.

Extending the 20Sim model It would be nice if the 20Sim model could be extended to encompass more effects that happen on the real setup. That way better and more reliable simulations can be run. It is recommended to try and put the backlash effect and the play that is present in the system in the simulation.

Hunt-Crossley model of the environment At the moment the linear Kelvin-Voigt model for the environment is used. It would be nice to extend the estimation algorithm to the case where the non linear Hunt-Crossley model can be used. The Hunt-Crossley model closer resembles the dynamics that occur when there is penetration into a wall, the damping is then also dependent on this penetration and not only on the velocity.

Time dependent design parameters In this project it has been done a little already and using time dependent design parameters looks very promising in the simulations. By changing the α of the update law or certain parameters inside the EKF over time the estimates are able to converge faster and still have less oscillations on their steady state values. It might be hard to do this in cases where the environment is changed during runtime but still it is recommended to look further into it.

Different environments During the experiments and simulations it was noticed that it is hard to estimate low damping parameters, if the estimate is off it can even become negative which is physically not possible. Using environments with a higher damping constant can help in showing the correct working in real life. An environment of which the damping is exactly known and can be adjusted would be helpful.

Tests in which the environment is changed during runtime would give an interesting perspective on how well the estimation algorithm would be able to cope with a dynamic environment. This should first be tested in simulations where it is easy to control the environment.

Different VSA Using a different VSA to implement the estimation algorithm on and showing that it still works would give a good argument for the applicability of using VSAs for environmental parameter estimation purposes.

10 Bibliography

References

[1] G. Grioli and A. Bicchi, ”A Non-Invasive, Real-Time Method for Measuring Variable Stiffness”, Robotics: Science and Systems, 2010.

[2] S.S. Groothuis, G. Rusticelli, A. Zucchelli, S. Stramigioli and R. Carloni, ”The Variable Stiffness Actuator vsaUT-II: Mechanical Design, Modeling and Identification”, IEEE/ASME Transactions on Mechatronics, 2013.

A Manual

This appendix gives a short overview on how to use the hard- and software of this project. It will also mention how to troubleshoot some commonly encountered problems.

A.1 Quick User Guide

This part will give a step by step approach on how to get the vsaUT-II operational and running, assuming nothing is in working order:

Preparations

- Boot the PC in the lab labeled VSA-II and login with the password 'ram'.
- Navigate to the directory where the firmware of the Arduino is saved.
- Open the firmware and **do not update** to the newest version.
- Turn on the power supply of the vsaUT-II.
- Press the upload firmware button.

Now the VSA should do its homing and calibration procedure. These steps are only necessary if you are not sure the correct firmware is uploaded to the Arduino.

Start up

- Launch Matlab and navigate to the directory where the 'loadModelParameters.m' file is stored.
- Run 'loadModelParameters.m'.
- Now open the desired Simulink model.
- Change the parameters inside the model to your liking and press the 'Incremental build' button.
- Turn on the power supply of the vsaUT-II if this is not done already.
- Once the computer is finished compiling press the 'Connect To Target' button, the VSA will do its homing and calibration procedure.
- Disconnect and connect to the target a couple more times until you think the lever arm output of the VSA is nicely centered.
- Press the 'Start real-time code' (the play) button.

Simulink

In the Simulink file a couple of Data Store Memory blocks can be found. These blocks hold some global variables that multiple functions in Simulink use. Here you can indicate what the used stiffness of the environment is, whether you want to anticipate for the driving belt stiffness or not etc. The names explain themselves.

The controller block consists of two simple PI controllers for q_1 and q_2. A reference value for the desired K_{vsa} can be set which is then converted to a setpoint for q_1.

The MO_SVFs (Multiple Order State Variable Filters) generate $\dot{r}$, $\ddot{r}$ and $\dddot{r}$. By changing 'Wn' the bandwidth can be chosen and by changing 'a' the damping factor.

Then there are the EKF blocks which hold the estimation algorithm and all the scopes that plot the data. The data can be saved from the scopes to the Matlab workspace where they can easily be analysed and saved.

The other parts in the Simulink file not treated are the parts that are necessary for the communication between the Arduino and the computer.

Finishing up

When you hit the stop button in Simulink the computer will stop sending commands to the VSA. The VSA will then keep its last inputs! Since the inputs $\dot{q}_1$ and $\dot{q}_2$ are velocity setpoints, the output will most often walk away to one of the edges of the system and start pushing against the side. The power supply should be turned off before this happens. Another method is to quickly reconnect to the VSA after pressing the stop button in Simulink. This will cause the VSA to do its homing procedure again and move its output to the center.

A.2 Troubleshooting

In this section some problems that might occur are treated.

No communication between the Arduino and the computer
When there is no communication between the Arduino and the computer it might be possible that the COM port is not set properly. During this project COM port 10 has been used. Go on the computer to 'Devices and Printers', select the Arduino Mega 2560 and set the COM port to 10. Next this should be the same in the used Simulink file. Open the Simulink file and go to the 'CMD_Plant' block. First go to the Stream Input block and set the Standard Devices Serial Port to [Ah] ($A = 10$ in hexadecimal notation). Then go to the 'Packet Output' block and then deeper down into another 'Packet Output' block. Here the Standard Devices Serial Port should also be set to [Ah].

The VSA runs towards its edges
This happens quite randomly when the VSA starts to move, right after pushing the play button in Simulink. Quickly turn the power off and try again. Make sure the homing and calibration routine set the VSA to the correct position, this seems to influence this behaviour. Often it can be predicted a short time before this happens, the gears setting the stiffness turn to maximum stiffness just before the VSA starts to move. It could be that there is some communication loss or delay causing the VSA to stay still for a couple of seconds, in the meantime the integrating part of the controller becomes big and once the VSA starts to receive signals it hits the edges.

Loss of communication
Another semi-random occurring event is loss of communication. During online experiments, suddenly the VSA stops receiving and sending new signals from and to Simulink. The VSA keeps its latest setpoints and will most likely move towards its edges. Simulink keeps receiving the values of the latest signals it received before the loss of communication. Sometimes after a short while the communication is restored again but most often the experiment needs to be redone. This seems to happen more often when the VSA is moving slowly or is stopped in one position.

Overflow
There is a possibility of overflow occurring in the Arduino if the $\dot{q}_1$ and $\dot{q}_2$ setpoints that are received are too high. When this happens the signals in the Arduino can become negative and wrong steady state values are reached. A way to prevent this is to go in the Simulink file to the vel2CPS block. Here the output can be divided by a certain number, for example 100. In the Arduino firmware code, the signals need to be multiplied by this number again and be put in a variable that can hold the correct size. This way the VSA is capable of obtaining much faster speeds but there will be a loss of accuracy in the velocity setpoint commands. Hence this is a trade off that needs to be considered.

Errors in the simulation
20Sim has some problems with errors during the simulation, especially when multiple derivatives are generated. The simulation might crash or unrealisticly high values of signals are plotted. These are often caused by numerical errors in 20Sim's solving approach, because of this care must be taken that the accuracy of the system is not set too low. The simulations might run quickly but the result can be unreliable. Therefore it is better to have slower simulations unless you want to have a quick look at what kind of effect something has on the system.

B Hardware and Software

In this part a more detailed view is given of all the used equipment in this project. For an even more detailed view on the mechanical structure and configuration of the vsaUT-II that has been used in this project see [2].

B.1 Hardware

As mentioned before the used VSA is the vsaUT-II designed by Stefan Groothuis. The VSA is able to change its apparent output stiffness from a minimum of $0.7Nm/rad$ to a maximum of approximately $900Nm/rad$. The reason it is not from zero to infinity is due to mechanical limitations and compliances in the system.

There are three sensors placed on the VSA, one encoder on each of the motor outputs that control q_1 and q_2 and one encoder on the output shaft that measures r. The two encoders on the motors are Maxon HEDL 5540-500 incremental optical encoders with 500 counts per revolution or 2000 pulses per revolution when using quadrature encoding, hence the accuracy is equal to $\frac{2\pi}{2000} = 3.1416 \cdot 10^{-3} radians$. The encoder on the output is an Austria Microsystems AS5043 absolute magnetic encoder with a resolution of 10 bits and thus an accuracy of $\frac{2\pi}{2^{10}} = \frac{2\pi}{1024} = 6.1359 \cdot 10^{-3} radians$.

As the 'wall' a spring balance is used of which the stiffness parameter is exactly known and from which it can be assumed to be linear in its working range. An extension spring is used to counter non-linear bending effects that would emerge when using a compression spring that is more resembling to a wall. For the estimation algorithm this does not make any difference. Two springs where used, one of $200N/m$ and one of $1000N/m$. The order of magnitude of the damping parameter of the springs has been characterised by modeling a simple mass, spring and damper system in Matlab. The springs are extended with a known weight and the time is measured that it takes for the spring motion to damp out. This is then compared to the time this takes in the model and from this the D_w of the springs are obtained. D_w is approximately $2Ns/m$ for both springs. In figure 33 a photo of the setup can be seen. As is visible the spring is translational while the VSA is rotational, hence the distance of the output shaft of the VSA needs to be taken into consideration in the equations. It is assumed that in the used working range of the VSA (approximately from $r = 0rad$ to $r = 0.3rad$) the motion of the endpoint of the spring balance can be assumed to be just translational. The piece of string connecting the spring and the VSA makes sure that when the output shaft of the VSA moves towards the spring, at some point contact will be lost.

For the communication between the computer which runs the estimation algorithm and the VSA an Arduino board is used. The Arduino is used as an interface between the sensors of the VSA, the motor inputs and the computer. The communication between the computer and the Arduino happens through a serial interface. The motor inputs are first send to two elmos which control the actual motors. It takes approximately 5ms before data is available in the elmos after a command is given.

The used VSA suffers from some issues which will shortly be addressed now. The internal two springs of the VSA are preloaded in the VSA and attached in such a way that whenever the output r is different from the equilibrium position q_2, one of the springs will be extended and the other one will be compressed. However the internal springs are a little too long which means that if the deflection is too big, the spring that is compressed will be back to its rest length and will push against its own windings. This in fact turns the spring into a solid bar pushing on the insides of the VSA. If the force becomes too high the spring can become damaged. Furthermore, the VSA has some calibration issues which makes it unreliable to assume the starting position of the VSA is correct after one calibration attempt. Often multiple tries are necessary. Lastly, there is some play in the output of the VSA introduced by the gear mechanism and the motor of q_2.

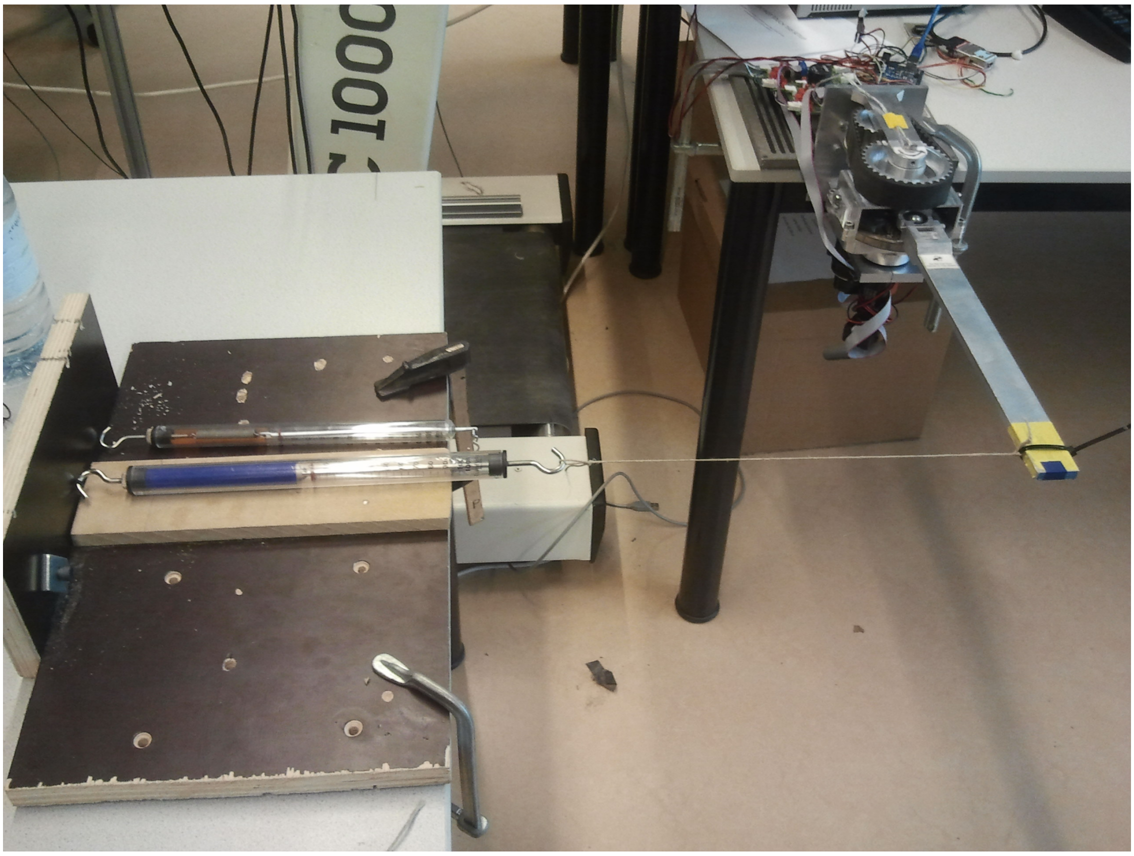


Figure 33: Photograph of the used environmental setup

B.2 Software

During this project extensive use has been made of certain software packages. The main simulation program used was 20Sim. 20Sim was used as a means to simulate the dynamics and the working of the VSA and the estimation algorithm before implementing it on the real setup. Other simulations using simpler models for testing small parts of the project were also done using 20Sim.

Next Matlab was used to help characterising the springs, test the state variable filters and process some of the data obtained from the real setup. The real time environment of Simulink, Matlab's simulation part, was used to send commands to the VSA through the Arduino and to process the data from the VSA. Simulink is also where the estimation algorithm was implemented. The calculations in Simulink were running at a sampling frequency of 200Hz.

Previously written software from earlier projects was used on the Arduino. One problem with the software is that there is a danger of overflow in the control commands which can cause the VSA to move in the wrong direction. This has been solved by dividing the control inputs by a specific value in Simulink and multiplying the signals in the Arduino again in a new variable.

Another apparent software issue is that sometimes there is a loss of communication between the Arduino and Simulink. At that point Simulink keeps receiving the last obtained sensor signals and the VSA will keep its last velocity setpoints, causing it to move towards the edges of its working space. This happens quite randomly.

C Model

To get a better understanding of the VSA the 20Sim model that was made by Stefan Groothuis [2] has been used as a basis. This model has been very useful in testing the correct workings of the estimation algorithm. As mentioned the model was made using 20Sim. 20Sim is a simulation program which can make use of bond graphs. A global view of the model can be seen in figure 34. The bottom part are the bondgraphs that represent the dynamics of the wall, as is visible the

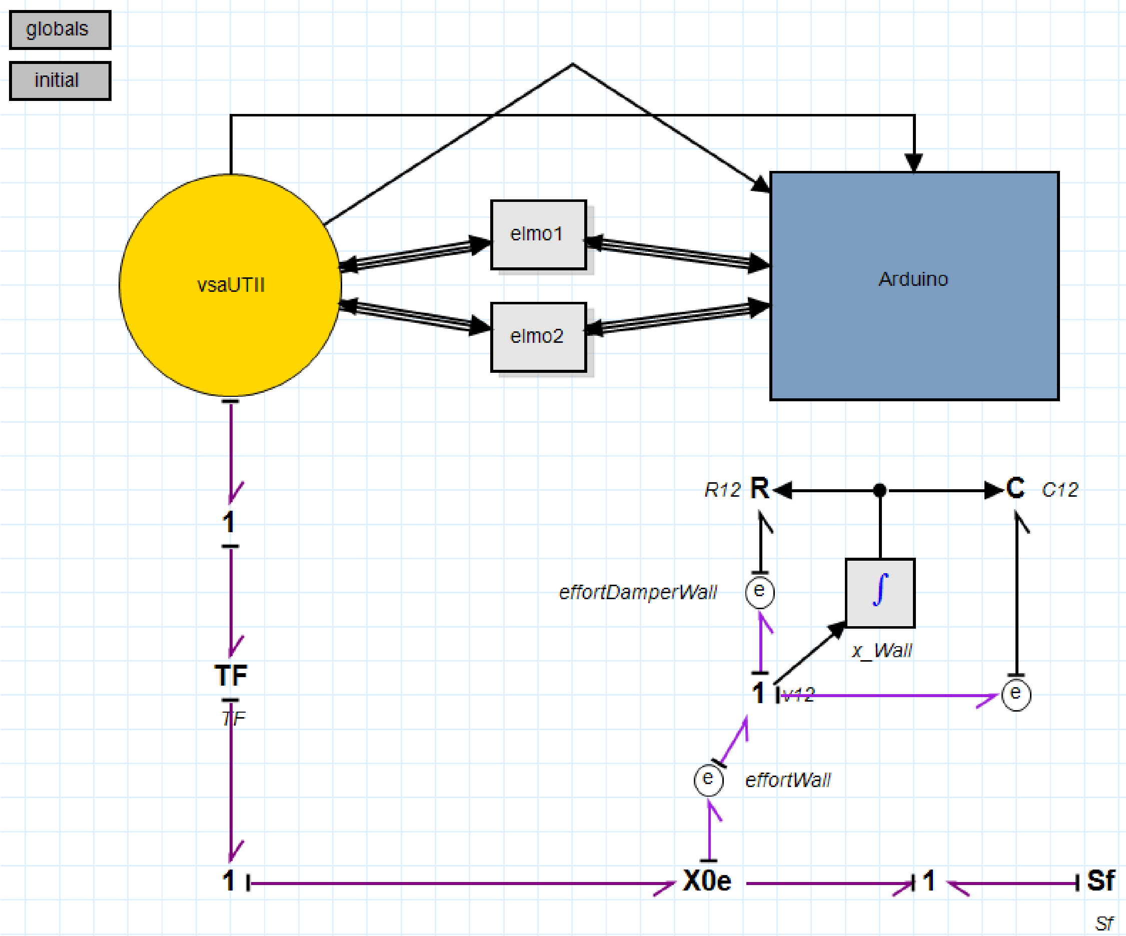


Figure 34: Global overview of the used 20Sim model, the interconnection between the VSA, the environment and the Arduino are displayed.

wall is connected through a switched zero junction, the $X0e$, which checks whether the VSA is in contact with the wall. The flow source connected to the switched zero junction has a flow of zero, meaning the wall is stationary. By adding the flow source the model can easily be extended to a model with a moving environment. The Arduino block contains the system controller and the estimation algorithm. On the real setup the Arduino is only used to communicate signals to and from the computer (Simulink) and to set limiters on the control signals such that the system cannot damage itself. These limiters are not present in the 20Sim model. The elmos are the motor controllers. The VSA itself is modelled using the Dirac structure principle, see figure 35. The storage port contains only the dynamics associated with the internal springs of the VSA, see figure 36, the interaction port is where the VSA is connected to the outside world, see figure 37, the control port contains all the system behaviour that controls the system (i.e. motors, gears, transmission belt, etc.), see figure 38, and the Dirac structure is the power continuous connecting element, see figure 39.

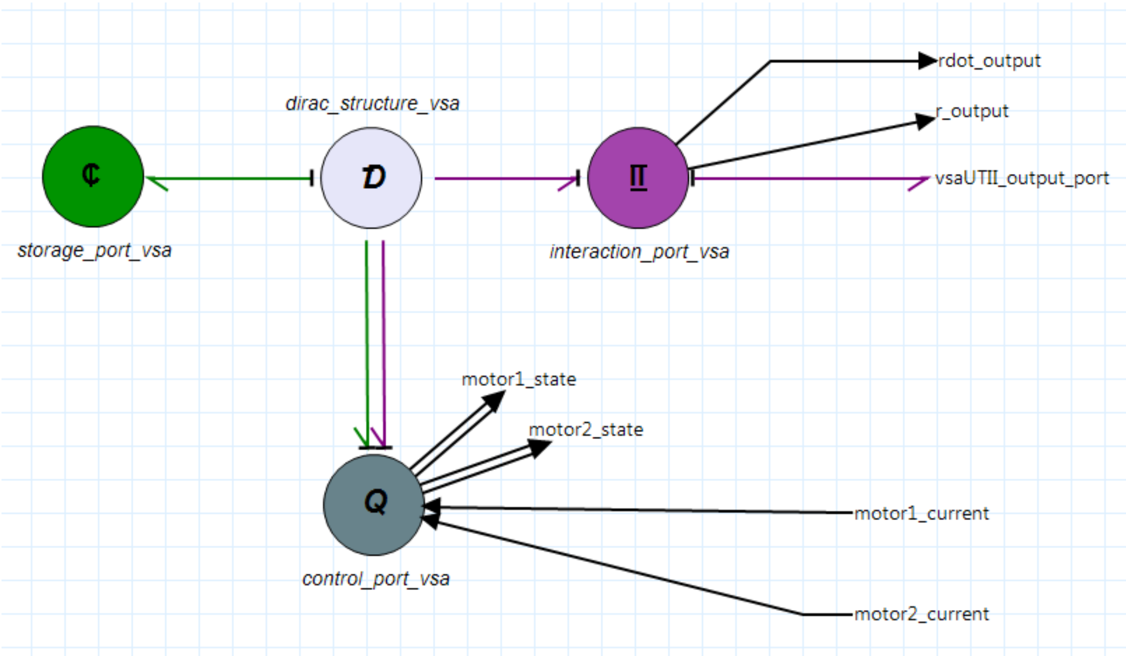


Figure 35: Model of the VSA where all significant elements are split and connected using a power continuous Dirac structure.

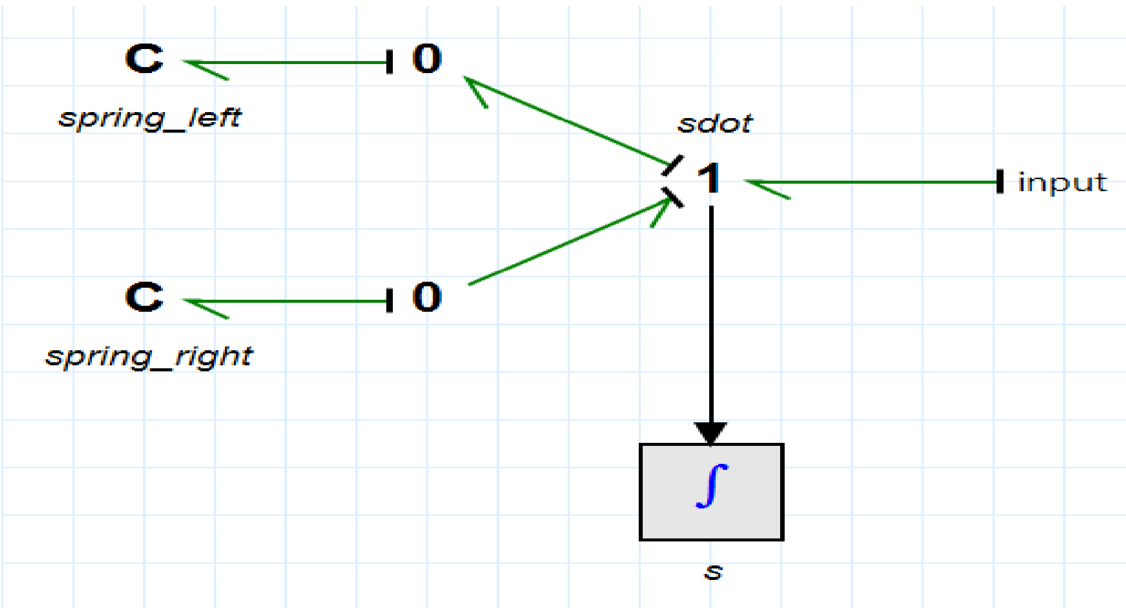


Figure 36: Model of the storage port, s is the state of the internal springs.

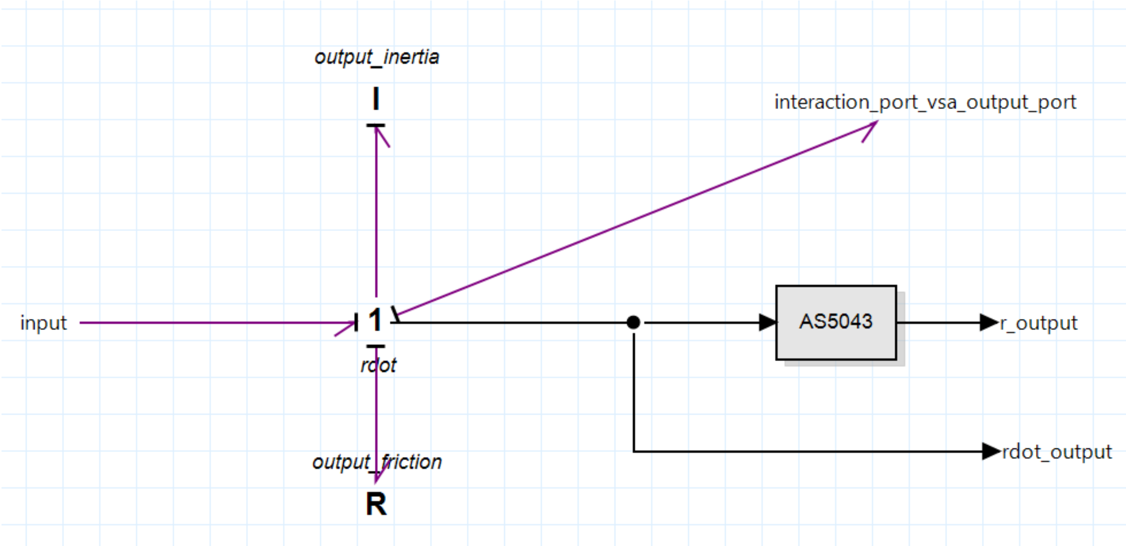


Figure 37: Model of the interaction port with the encoder.

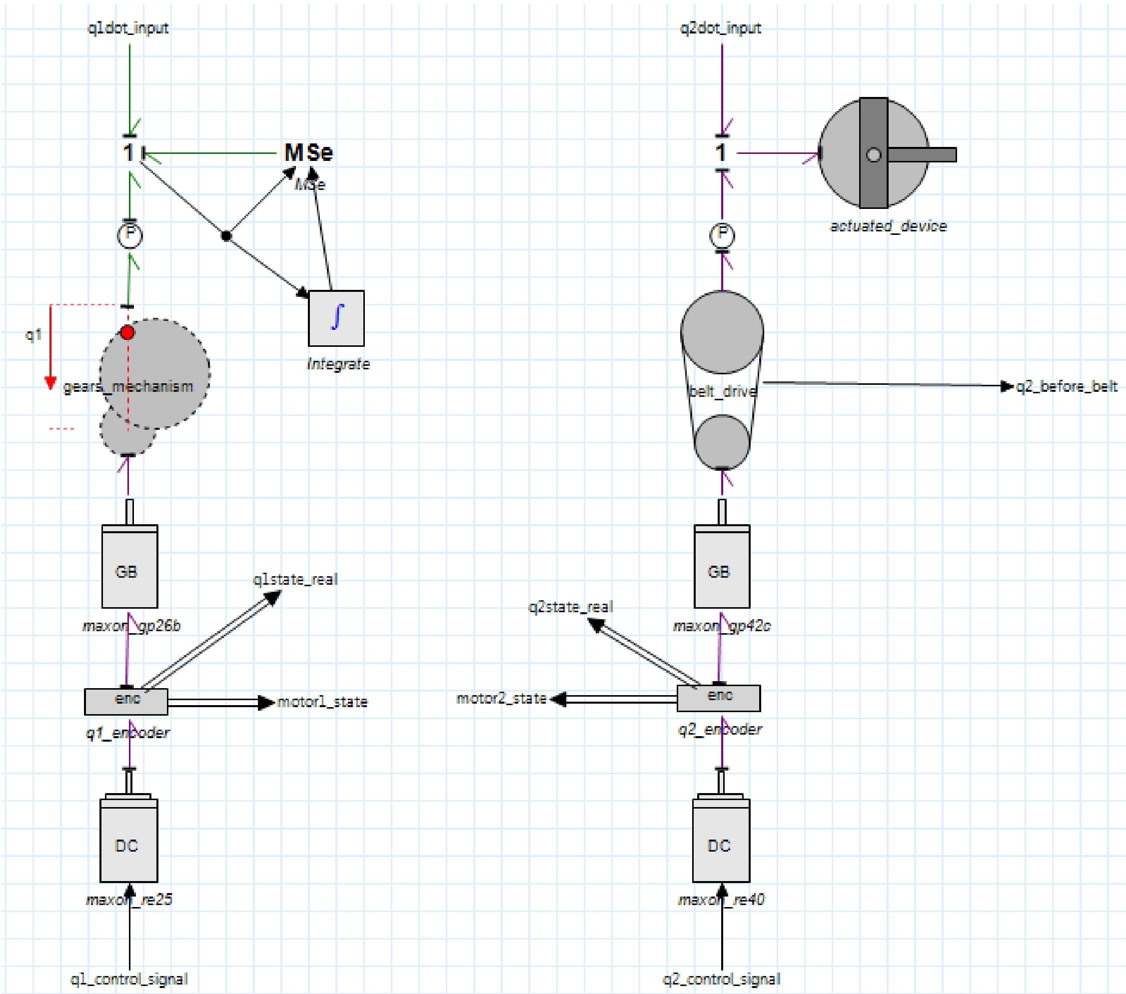


Figure 38: Model of the control port representing the biggest part of the mechanical structure and interconnections.

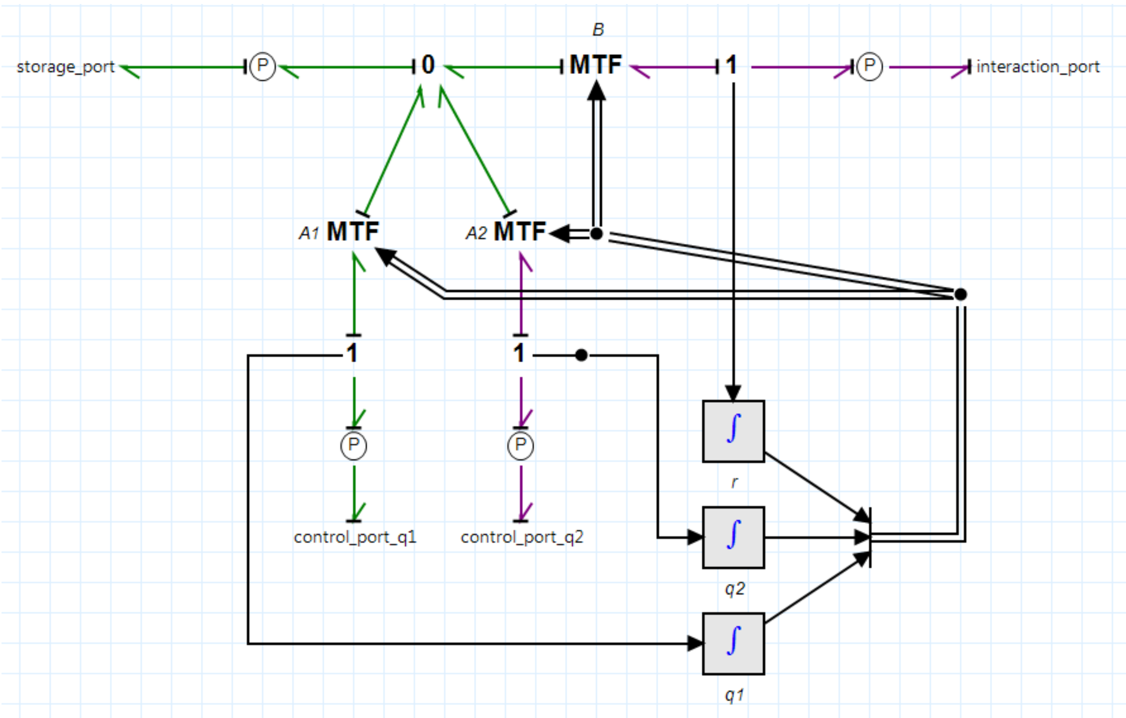


Figure 39: Model of the Dirac structure which connects the separate elements of the VSA and is power continuous.