Modeling and control of a mobile loader crane with an open-center valve

Project report for **PROJECT**H3

Authors:

Anton Dahlqvist
Anton Söderberg
Hjalmar Lundin
Johan Kuylenstierna
Jitin Thomas
Ludvig Michaelsson
Daniel Mofors
Robert Axelsson-Sjöblom
Robert Regula
Yantong Zhang

Supervisors:

Björn Möller
Tahir Qureshi

December 16, 2015

Abstract

This project is conducted by a team comprised of students undertaking their M.Sc. in Mechatronics at the Royal Institute of Technology (KTH). The project is a collaboration between KTH and Hiab wherein a team of ten students work over three study periods (between March and December 2015). Hiab is a company seated in Hudiksvall and one of the world's leading provider of on-road load handling solutions. This project is based on the loader crane, XS-022 which is equipped with an open center valve and a fixed displacement pump. With this setup, interference occurs if multiple functions are run simultaneously. This interference is the reduction or lack of flow in the cylinder with the highest pressure due to all the oil being diverted to cylinders with relatively lesser pressure.

The aim of the project is to deliver a verified model of the Hiab XS 022 considering both the mechanical movements as well as pressures and flows in the hydraulic system. Along with the model, a controller needs to be designed and implemented in order to reduce the interference. The presented model is a continuation on the work done by M.Sc. students from KTH 2014. Their model has been used as a basis for the model development, where additional modules such as static friction and point load, have been added in order to improve the model. Furthermore, a SimScape model has been designed as a complement to the previously mentioned model.

A linear controller has been designed in Simulink and was first tested on the model. It was then implemented on the real system in a controlled hardware-in-the-loop environment at KTH where a Hiab XS 022 is mounted and connected to dSPACE.

As the aim of the project was to develop and verify a model which can be used to develop and implement a velocity controller on the crane, the project has been successful as a whole, since a successful controller has been implemented. However, a verified model of the system could not be delivered due to the high complexity of the system since more information of individual components is needed.

Acknowledgments

We would like to express our gratitude to our support team at Hiab, especially Tahir Qureshi and Daniel Morales for advice regarding software and control implementations and also Amy Rankka and Erik Forzelius for support regarding hydraulic components and crane specifics.

Further we also would like to thank Bengt Eriksson for support with dSPACE and advice regarding control theories, and Vicki Derbyshire for administrating the course and always being willing to help us.

The teaching team, Martin Grimheden, Staffan Qvarnström, Jad El-Khoury, Mohammad Khodabakhshian Khansari and Björn Möller have been supportive and helpful by continuously reviewing our design decisions during the project. We would like to thank them for this support, and especially Björn which also has been our supervisor for the help extended throughout this project.

Contents

Chapter 1 Introduction

This chapter serves as a general introduction to the project. It includes the background, the purpose and problem description, the project scope as well as methods used.

1.1 Background

The project is conducted as a collaboration between the Royal Institute of Technology (KTH) and the Swedish company Hiab. It features a team of M.Sc mechatronic students tasked to improve the operation of a loader crane (specifically, model Hiab XS 022) that is designed by the company.

Loader cranes are also known as knuckle-boom cranes or articulating cranes. These names stem from the fact that there are jointed boom sections which together form an articulated arm. This form factor allows the crane to be mounted on a truck (see fig. 1) and occupies little space when it is in its folded position.

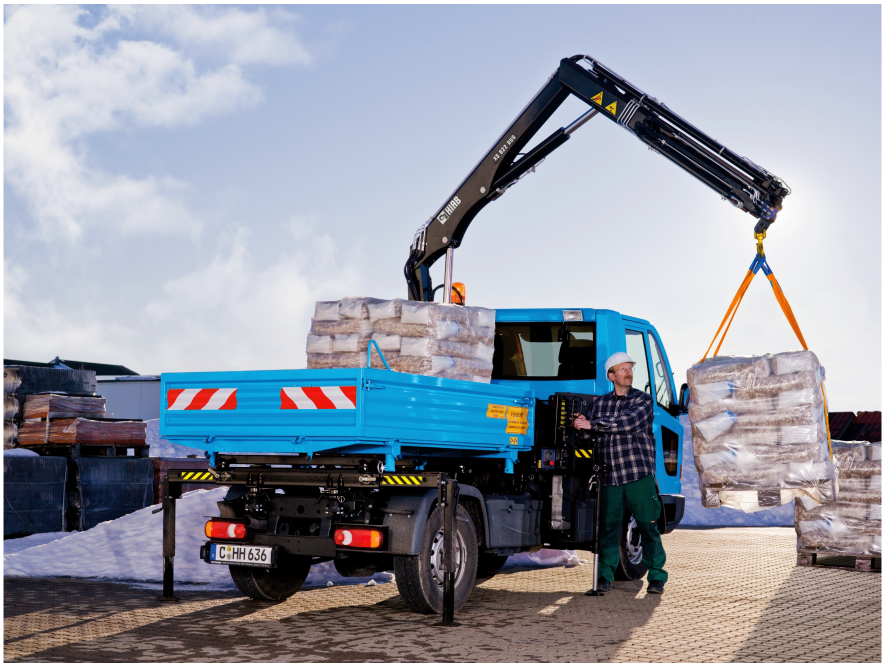


Figure 1. A Hiab XS 022 loader crane mounted on a small truck[1]

The lifting capacity of these types of cranes is measured in tonne-meters (t m). This is because the varying distance from the base to the load produces different stress at the components. The Hiab XS 022 has a capacity of 2.2–2.6 t m. (Cargotec, 2015)

Hiab XS 022 usually consists of four actuators, three in the form of hydraulic cylinders and one rotary actuator for the slew. There are two approaches to supply hydraulic oil to the actuators. One option is to use closed center valves and variable displacement pumps which can adapt the flow from the pump depending

[1] http://www.hiabus.com/models/hiab-cranes/hiab-xs-022 , Accessed on December 16, 2015

of the current load. Option two is using open center valves with fixed displacement pumps, which instead deliver a constant flow of hydraulic oil.

Open center valves are cheaper and more reliable than closed center valves. For this reason, Hiab would like to keep offering their clients this type of solution. However, due to the constant oil flow from the pump, the various functions may interfere with each other. This interference occurs when multiple actuators are enabled at the same time and the oil flows along the path with least resistance, i.e. to a certain actuator. This may result in one or more functions stopping until the remaining ones are terminated. The goal of this project is to investigate the ability to counteract these interferences.

An earlier project was conducted by a team of M.Sc students in mechatronics regarding open center valves during 2014. The project produced non-verified models of the hydraulics and mechanics of one of Hiab cranes. These models serves as a starting point when approaching the tasks concerning this project.

1.2 Project goal

The goal of the project carried out in 2014 was to derive a model of the whole crane, with an open center valve and use that model to design a velocity controller. Since these final goals were not reached, the goals are inherited into this project. In order to reach those goals the model needs verification so that it corresponds to the physical crane. Specifically, the estimated parameters and constants have to be evaluated.

Moreover, the overall goal of this project is to use the model to create a velocity controller which will enable all cylinders to actuate simultaneously. This can be accomplished by controlling the open center valve so that the oil flow is shared by the functions, in order to provide an output that is proportional to the user input.

From the team's aspect, the goal is to carry out a project from start to finish and to gain knowledge and experience while doing so. In the spring period, focus was on planning the project. Verifiable requirements were defined, a state of the art report was prepared and an interface for communication with the crane was established using hardware and software from the previous project i.e. reading sensors and sending reference signals. The goals for the fall semester include verification of the crane model and also the design and implementation a velocity controller. From a learning perspective and for the interest of the project group, another goal is to learn hydraulics and different methods of hydraulic control.

1.3 Method

The project has been divided into a few main objectives (development areas). These objectives include model development, control development, software and interface development, and finally crane maintenance and testing. Additionally, there are other responsibilities such as administration and project planning. This decision was made due to the large team size (10 people involved), which should enable concurrent development suitable for the adapted Scrum methodology used within this project. In table 1, the main development areas are further explained.

Table 1. Explanations of the development areas

Model development	Responsibilities concern verification, validation, and consequent development of the model, i.e. looking through the model for errors, deciding on which parts to be removed or added, examining measurement data, etc.
Control development	Responsibilities concern developing a controller to remedy the interference issues. The development of this controller is based on the model. The control development team is also in charge of implementation of the controller for testing, i.e. the implementation onto the development platforms.
Software and interface development	Responsibilities include managing the interfaces for the development platforms, i.e. how to input references and collect the necessary variables and data. This should be as easy as possible to allow for a higher focus on the other development areas.
Crane maintenance and testing	This final responsibility area considers the maintenance of the crane and sensors installed on it. It also includes making sure that the system is capable of delivering the necessary data at a hardware level, i.e. installing new sensors where necessary, as well as running the test cases.

The development is also iterative, meaning that once a change/addition/removal has been introduced, the result is to be tested and discussed with another team member that has a main responsibility for that particular area. Then the work is continued based on the feedback given. By doing this, the work is monitored continuously and work quality is assured.

1.4 Scope and delimitations

The delimitations chosen for the project are mostly based on the availability of hardware at the premises of KTH. Additionally, the scope will be limited by time restrictions. The main delimitations regarding the project goals are presented below.

- The project will only consider modeling and controlling the Hiab XS 022, it will not focus on making a general model applicable for an arbitrary loader crane. This delimitation comes from time restrictions as well as that there is only this specific model available for testing and verification. This limitation includes the valve model and other hydraulic peripherals mounted on the crane.
- The project will only consider the interference in regards to different loads and three of the four functions available on the test crane. The slew will be disregarded. This delimitation comes from the risks associated with using the slew given the crane is positioned in a lab hall.
- The project will focus on the crane in the lab hall, it will not consider potential differences to the models that may occur when the crane is mounted on e.g. a truck.

Further, delimitations regarding which theories and literature to study will be influenced by the project goals. The main delimitations regarding these aspects are presented below.

- The project will not consider black box models of the crane. Because of the system complexity and multiple input-output characteristics, black-box models of the entire crane will likely not be accurate enough to base control design. This was also one conclusion drawn from the project 2014.
- The project will not consider aspects such as energy consumption, e.g. making a controller that will regulate and optimize the usage of the truck or pump engine.

Chapter 2 State of the art

This chapter describes the state of the art analysis performed in the beginning of the project. An introduction and several control theories are presented, followed by verification procedures and parameter tuning methods.

2.1 Scope of the information retrieval

The main scope of the state of the art (SOTA) is to find the best control strategy applied to hydraulic systems with multiple inputs and outputs. It will cover different control strategies for both model based control as well as for non model based control. The result from the following research should provide the right knowledge to decide which control strategy that best fits this application.

The literature review will not cover hydraulic modeling. The previous Hiab project focused their research mainly about modeling and therefore this material will instead be used if any further improvements have to be done with the model. However, it will touch up on the subject of model verification and parameter tuning.

2.2 Model-based control

In model based control (MBC), control approaches stem from process models. From these, the control signals and variables to be controlled can be identified and studied.

There are several different approaches for MBC. Overall, it's about first designing a model for the plant to be controlled. This model dynamics is mainly described by suitable mathematical equations. The MBC structure relies heavily on the effectiveness of modeling. A controller based on an inadequate model will not work as expected all the time. MBC works well when plants are relatively easy to model but as the plants get too complex these control strategies drop in effectiveness. (Wang, 1997, p. 837)

Using this approach, control methods include proportional-integral-derivative (PID) control, lead-lag control, linear-quadratic-Guassian (LQG) control, $\mathcal{H}_\infty$ control, and model predictive control (MPC). All of these will be described below..

2.2.1 PID control structures

The proportional integral derivative (PID) control structure is one of the most commonly used feedback control structures in the industry today. The ideal version of a PID controller in time domain can be described as

$$u = \underbrace{k_P e(t)}_{Proportional} + \underbrace{k_I \int e(\tau) d\tau}_{Integral} + \underbrace{k_D \frac{d}{dt} e(t)}_{Derivative}, \tag{1}$$

where u is the control signal and e is the error signal defined as the difference between the reference r and the output signal y according to

$$e = r - y. \tag{2}$$

Moreover, the constants k_P, k_I, k_D represent the proportional, integral and derivative parts respectively.

The control loop associated with a PID controller can be designed using either one, two or three degrees of freedom. The one degree of freedom design is shown in fig. 2.

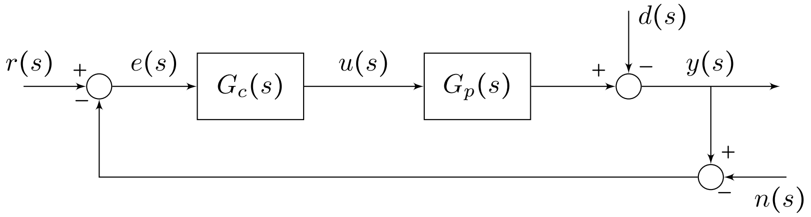


Figure 2. A one-degree-of-freedom PID controller design

The Figure shows the plant model G_p, the control model G_c, the measurement disturbance $n(s)$ and the noise disturbance $d(s)$. It also shows the reference signal r, output signal y and the error signal e. Although this one degree of freedom design is the most commonly used, it results in a control structure where the measurement noise can not be attenuated independently of the reference signal. The closed loop transfer function

$$y(s) = \left[\frac{G_p(s)G_c(s)}{1 + G_p(s)G_c(s)}\right](r(s) - n(s)) + \left[\frac{1}{1 + G_p(s)G_c(s)}\right]d(s) \tag{3}$$

further illustrates the dependence between the reference signal and the measurement signal for the output, y. A two degree of freedom design solves this problem by providing the user with two different control components, one which is related with loop stability and another which can shape the closed-loop response. This design is shown in fig. 3.

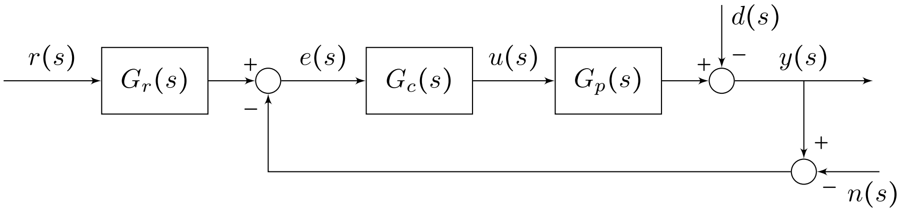


Figure 3. A two-degree-of-freedom PID controller design

The closed loop transfer function

$$y(s) = \left[\frac{G_p(s)G_c(s)G_r(s)}{1 + G_p(s)G_c(s)}\right]r(s) + \left[\frac{G_p(s)G_c(s)}{1 + G_p(s)G_c(s)}\right]n(s) + \left[\frac{1}{1 + G_p(s)G_c(s)}\right]d(s) \tag{4}$$

for this design shows how the added transfer function block $G_r(s)$ affects the closed loop response. Furthermore, a three degree of freedom model is shown in fig. 4.

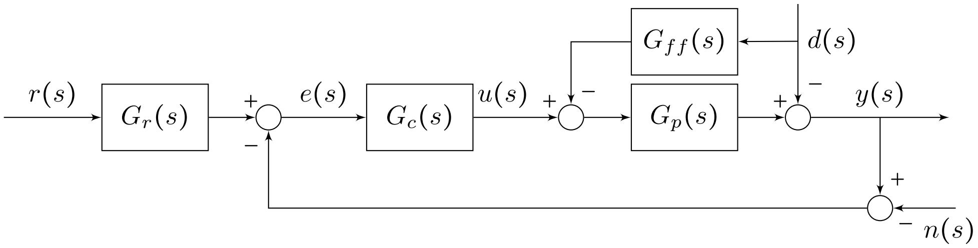


Figure 4. A three-degree-of-freedom PID controller design

This design has introduced yet another transfer function block G_{ff} which enables another degree of freedom. As the closed loop transfer function

$$y(s) = \left[\frac{G_p(s)G_c(s)G_r(s)}{1+G_p(s)G_c(s)}\right] r(s) + \left[\frac{G_p(s)G_c(s)}{1+G_p(s)G_c(s)}\right] n(s) + \left[\frac{1-G_p(s)G_{ff}(s)}{1+G_p(s)G_c(s)}\right] d(s) \tag{5}$$

shows, this block enables a more independent control of the noise disturbance affecting the system (Johnsson and Moradi, 2005, p. 57-59).

2.2.2 Lead-Lag controller

The Lead-Lag compensator is a control structure consisting of a Lead and a Lag term. The structure can be represented in many ways, below is one example (Ljung and Glad, 2006, p. 109).

$$F(s) = K \underbrace{\frac{\tau_D s+1}{\beta\tau_D s+1}}_{Lead} \underbrace{\frac{\tau_I s+1}{\beta\tau_I s+\gamma}}_{Lag}. \tag{6}$$

By using a Lead-Lag compensator a pole-zero is added to the open loop transfer function of the system. The lead compensator gives phase at high frequencies, shifting the poles to the left, leading to a higher stability and responsiveness of the system. The lag compensator introduces phase at low frequencies, resulting in lower steady state error (Ljung and Glad, 2006, p. 105).

The desired pole and zero location depends on the closed loop response as well as on the system being controlled. It should be kept in mind to keep the pole and zero close together, not further than a factor of ten from each other, to prevent poles from shifting to the right, which would cause instability or slow convergence. Ultimately they should be placed near the origin to affect the low frequency behaviour (Levine, 1996, p. 196).

When implemented the Lead-lag controller is expressed as an integral equation, because when differentiating signals the noise gets amplified. Integrators are the most numerically stable when implemented.

The Lead-Lag compensator may be designed in several ways depending on the constraints. Below is an approach where the compensator is designed in two separate steps (Babuska, 2011). The Lead compensator is designed as follows

1. Determine the desired crossover frequency ω_c.
2. Calculate how much extra phase the lead compensator must add at ω_c and then calculate β and τ_D.
3. Compute the overall controller gain K such that ω_c becomes the crossover frequency.
4. Check whether the bandwidth specifications are met, if not, recalculate a new ω_c.

and the Lag compensator part as follows

1. Determine the DC gain $1/\gamma$ of the lag compensator to meet the steady-state requirements.
2. Choose τ_I between $2/\omega_c$ and $10/\omega_c$.
3. Check whether all the control specifications are met, if not, revise.

The advantages with the Lead-Lag compensator is that it gives a low frequency phase and a high frequency roll-off. However, it is slightly more complicated to implement than a PID-controller.

2.2.3 Decentralized control system

An interactive multiple input/multiple output (MIMO) process can be controlled by either a centralized multivariable controller, or if it is possible, a decentralized multi-loop controller which consists of a set of single input/single output (SISO) controllers. The big difficulties with cross-coupled controllers are their complexity and that they are weak in control-system integrity (Xiong et al., 2006). Consider an multi-variable process plant with the transfer function

$$G(s) = \begin{bmatrix} g_{11} & \cdots & g_{n1} \\ \vdots & \ddots & \vdots \\ g_{1n} & \cdots & g_{nn} \end{bmatrix} \tag{7}$$

Decentralization of the system refers to disregarding cross couplings and treat one loop at the time as if the other inputs and outputs didn't exist. A controller that satisfies this condition is

$$G_c(s) = \begin{bmatrix} g_{c1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & g_{cn} \end{bmatrix} \tag{8}$$

This means that the sub-part of the controller G_{ci} is only based on the measured signal y_i and its reference r_i. Decentralized controllers are always square, so if the number of outputs exceed the number of inputs some need to be disregarded. The cross couplings from the off-diagonal elements that are disregarded in the control structure may affect both the stability and the performance of the closed loop system. If no interference between the cross couplings exist, then the system can be seen as different uncorrelated systems (Glad and Ljung, 2000, p. 221).

How to map which input to which output in order to get the least amount of cross coupling is an important part of making sure that the system has the ability to be decentralized. This is called the pairing problem and it also includes the problem of knowing which measurements to use when the number of outputs exceeds the number of control inputs.

The relative gain array (RGA) is an indicator for the stability of a decentralized controller. It is defined by Glad and Ljung (2000, p. 219) as

$$RGA(G) = G \otimes (G^{-1})^T \tag{9}$$

Two conditions that that determine whether the decentralized system is stable or not are as follows

- If a decentralized controller is stable in each single-input-single-output (SISO) loop, and if $RGA(G(i\omega)) = I$, then the closed loop system is also stable. I is the identify matrix with the same size as $G(i\omega)$ (Glad and Ljung, 2000, p. 222)

- If some diagonal element of $RGA(G(0))$ is negative, and a diagonal controller is used, then either the closed loop system is unstable, or it will become unstable if one of the SISO loops are broken. (Glad and Ljung, 2000, p. 222)

A pairing rule according to Glad and Ljung (2000, p. 222) can be derived from the first condition as

- Make the diagonal elements of $RGA(G(i\omega_c))$ as close to the point 1 in the complex plane where ω_c is the intended cross-over frequency of the closed loop system.

If the system is stable then the multi-loop feedback system can always be made stable provided the loop gains are de-tuned to be sufficiently small. De-tuning means adding a common factor to all loops such that the overall system is stable and gives acceptable load disturbance rejection responses. If the conditions above are not met, or if it's difficult to see which input that matches which output, an alternative is to try to make a dynamic decoupling of the system. This means that the controller compensates so that the system can be treated as a decentralized system (Johnsson and Moradi, 2005, p. 107). The decoupling method uses two weighted matrices to transform the transfer function matrix, G, into the matrix

$$G_{diag} = W_2 \begin{bmatrix} g_{11} & \cdots & g_{n1} \\ \vdots & \ddots & \vdots \\ g_{1n} & \cdots & g_{nn} \end{bmatrix} W_1 \tag{10}$$

which is as diagonal as possible. The weighted matrices W_1 and W_2 can be chosen dynamically or real valued. Dynamical matrices suits all frequencies whilst real valued matrices are chosen to compensate at selected frequencies. The matrices are typically related to to the system matrix G and requires a good knowledge of the plant model (Glad and Ljung, 2000, p. 226).

A drawback with using a decentralized controller is that only steady-state information is used in the design, which limits the controller performance (Xiong et al., 2006). The controller is not adaptive to changes in the model parameters, e.g. different loads or even different states of the system. This can be solved by introducing different multiple SISOs for different loads and for different states of the system with some kind of condition when the system changes which controller is used. However, the complexity of the system increases with multiple controllers which contradicts the purpose of making a decentralized controller.

Some advantages with a decentralized control system if implemented are, according to Xiong et al. (2006), that the structure is easier to understand and to implement, it requires fewer parameters to tune, and loop failure tolerance of the resulting control can be evaluated during the design phase.

2.2.4 Model predictive control

The MPC is a control approach that predicts the outcome of a specific model. The idea with MPC is to use the model as a reference point and, in real time, calculate multiple paths that can be taken by assigning different values for the controlled variables and look at the system model response. An algorithm is implemented that goes through the different paths and choses the control action that is closest to the desired outcome within a specified timespan. The approach allows for specifying rules and constraints which the controller takes into consideration when choosing the appropriate control action and can in some cases sacrifice the short term properties to optimize for the long term outcome. (Anderson et al., 2007)

MPC can be used with either a model design based on ordinary differential equation (ODE) or as a fuzzy model, for example neural networks. However, the use of the MPC approach shifts the problem of creating a controller to the modeling part. This due to simpler implementation of a controller, especially when dealing with multiple-input-multiple-output (MIMO) systems, but it needs a good model to correctly predict the different outcomes and take the appropriate path. Apart from MIMO it can also be implemented in single-input-multiple-output (SIMO) systems which will have lower computational time due to fewer possible paths. (Anderson et al., 2007)

The approach can be shown in a few very general and flexible steps that has to be customized to fit the model and controller that is to be created. (Glad and Ljung, 2000, p.383) & (Anderson et al., 2007)

1. Create a very accurate model of the system, either an ODE or fuzzy model. And set desired constraints and rules for the system.
2. Use the current conditions, at time t, as the initial conditions.
3. Compute or predict a set of candidate control actions based on the initial conditions within a finite timespan.
4. Based on these control actions, choose the outcome that is within the error margin and has the shortest path, i.e. optimize that control action with respect to some conditions.
5. Apply the control action to the physical plant.
6. Observe the system and wait for the next sample, at time $t + 1$, and iterate from step 2.

Using the MPC approach come with some advantages compared to other control approaches, such as regular PID controllers etc. It allows the programmer to specify constraints and rules, e.g physical constraints on the plant, that needs to be met. Due to the long term view of the controller it will also go through less undesirable states to get a large return as well as anticipate consequences of a specific control action.

The disadvantages are mostly due to the predictions being based on the model. A very inaccurate model can cause poor control decisions. The algorithm for the prediction can be very computationally demanding for a big MIMO system and poorly written algorithms might result in a path that includes an undesirable state before reaching the desired outcome. (Anderson et al., 2007)

2.2.5 Linear-quadratic-Gaussian controller

The LQG control problem is to find the optimal control u(t) which minimizes the deterministic cost, i.e

$$J_r = \int_0^\infty \left(x(t)^T Q x(t) + u(t)^T R u(t) \right) dt \tag{11}$$

where Q and R are design parameters, x is the system state and u the control signal. This is solved by using two different algebraic Riccati equations which depends on A, B, C, Q and R. (Skogestad and Postlethwaite, 2005, pp. 375 – 385)

Assuming a system

$$\begin{gathered} \dot{x} = Ax + Bu + w \\ y = Cx + v \end{gathered} \tag{12}$$

where w and v are zero mean white Gaussian noise, A, B and C are state space matrices and y is the output signal. If this system is linear and the cost is described by a quadratic function, see eq. (11), this is called a LQ-problem. to control this system the LQG controller is then designed in two steps, which is also called the separation theorem. First, the linear-quadratic regulator (LQR) is used to determine the optimal controller to this LQ-problem without noise. This gives the control signal

$$u(t) = -K_r x(t). \tag{13}$$

Second is to find the optimal estimate $\hat{x}$ of x such that the quadratic error, i.e.

$$E\left\{ [x - \hat{x}]^T [x - \hat{x}] \right\} \tag{14}$$

is minimized using the Kalman filter. This gives a control signal

$$u(t) = -K_r \hat{x}(t). \tag{15}$$

The LQG controller is relatively easy to design and to compute for both SISO and MIMO. Since the LQG controller uses the same methodology independently of the number of state variables and input and output signals, it fits well for this projects purpose. However, some problem with stability and robustness suggests this should be investigated further. (Athans, 1986, pp. 2 – 3)

The quality of the approximated feedback loop depends on the system model. To get an arbitrary good approximation, the system should be minimum-phase (all poles and zeros lies in the unit circle for a discrete-time system for example). If the system is non-minimum-phase, the correctness of the approximation will depend on the location of the specific non-minimum-poles.(Athans, 1986, p. 3)

In general, a more accurate sensor gives a more accurate estimation. However, the increase in control accuracy is not necessarily improved by much with a better sensor. The difference in sensors only affect the last term (out of four) in the evaluated cost function. Moreover, the difference in accuracy for the actuators is slightly more important since they affects the last two terms in the same function. (Athans, 1971, p. 547)

2.2.6 $\mathcal{H}_\infty$-control

The $\mathcal{H}_\infty$ norm of a stable scalar transfer function is simply its peak value as a function of frequency, i.e.

$$\|f(s)\|_\infty \equiv \max_\omega |f(j\omega)| \tag{16}$$

where it's peak value can be defined as

$$\max_\omega |f(j\omega)| = \lim_{p\to\infty} \left(\int_{-\infty}^{\infty} |f(j\omega)|^p \mathrm{d}\omega \right)^{1/p} \tag{17}$$

since raising $|f(j\omega)|$ to an infinite power will result in the peak value. This makes $\mathcal{H}_\infty$ a purely mathematical term. The control design method simply aims to press down the peaks of one or more selected transfer functions. The $\mathcal{H}$ is the Hardy space, a set of stable and proper transfer functions. Replacing infinity with 2 in the eqs. (16) and (17) equals $\mathcal{H}_2$ instead (i.e. the Hardy space with square norm). (Skogestad and Postlethwaite, 2005, p. 60)

Skogestad and Postlethwaite (2005, p. 385) argues that the method is motivated by the shortcomings of LQG, which are mainly its poor robustness properties. LQG is a special case of $\mathcal{H}_2$ optimal control.

Given a standard configuration of the system (see fig. 5),

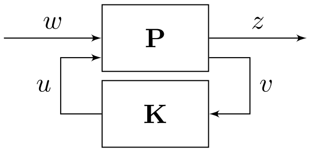


Figure 5. The standard configuration of multivariable systems

the system equations can be set up as

$$\begin{bmatrix} z \\ v \end{bmatrix} = \mathbf{P}(s) \begin{bmatrix} w \\ u \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \tag{18}$$

$$u = \mathbf{K}(s)v$$

where z is the error signals that are to be minimised and the measured variables v that are used to control the system. By expanding the matrix operations, it can be noted

$$z = f(\mathbf{P}, \mathbf{K})w = \left(P_{11} + P_{12}\mathbf{K}\left(I - P_{22}\mathbf{K}\right)^{-1} P_{21}\right)w. \tag{19}$$

Using $\mathcal{H}_\infty$ control, the goal would be to find a controller $\mathbf{K}$ such that $f(\mathbf{P}, \mathbf{K})$ is minimised according to eq. (17). The controller can be found using several different methods, including $\mathcal{H}_\infty$ optimal control, mixed-sensitivity $\mathcal{H}_\infty$ control, signal based $\mathcal{H}_\infty$ control, $\mathcal{H}_\infty$ loop shaping, etc. (Skogestad and Postlethwaite, 2005, pp. 390–400)

The techniques have the advantage over classical control techniques in the sense that they are applicable to problems involving multivariable systems with cross-coupling. However, since the controller will be optimal in the formulated sense a good model of the system to be controlled is needed. Additionally, the mathematics behind the control strategy might be difficult to understand. (Cubillos and de Souza, 2010, p. 4)

2.2.7 Exact Linearisation

Exact Linearisation Non-linearities within systems can be helped by introducing inner loops. A systematic way of introducing inner loops which removes these non-linear behaviours completely is called exact linearisation. There are generally two different control techniques associated with exact linearization, input/output linearization and input/state linearization. In order to describe these two approaches, consider the system

$$\begin{cases} \dot{x} = f(x) + ug(x) \\ \quad y = h(x) \end{cases}, \tag{20}$$

where the control signal, u, and the output signal, y, are scalars. Furthermore, the functions f,g and h are indefinitely differentiable.

Input/output linearization is a control technique where the output y of the non-linear system is differentiated until a physical input u appears in the r^{υ} derivative of y according to

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \\ \vdots \\ y^{\upsilon-1} = \alpha(x) + \beta(x)u \end{bmatrix}, \tag{21}$$

where $\beta(x)$ and $\alpha(x)$ are linear functions of x. The sign υ is the relative degree of the system which is the smallest integer which satisfies the equation

$$\underbrace{\left(g_1\frac{\partial}{\partial x_1} + \cdots + g_n\frac{\partial}{\partial x_n}\right)}_{L_g}\underbrace{\left(f_1\frac{\partial}{\partial x_1} + \cdots + f_n\frac{\partial}{\partial x_n}\right)}_{L_f}^{\upsilon-1} h(x) \not\equiv 0 \tag{22}$$

where L_f and L_g are the lie derivatives. The feedback, u, is then chosen as a transfer function dependent on a synthetic output

$$y^{\upsilon-1} = v \tag{23}$$

which gives

$$u = \frac{1}{\beta(x)}[-\alpha(x) + v]. \tag{24}$$

The closed loop system can then be written on the form

$$\frac{y(s)}{v(s)} = \frac{1}{s^{\upsilon-1}}. \tag{25}$$

This system which is the result from an input/output linearization may contain inner dynamics. This approach is therefore only suitable when the zero dynamics of the resulting system are well behaved. Otherwise, a input/state linearization solution should be explored instead (Glad and Ljung, 2000, p. 401) (Hedrick and Girard, 2005, p. 140). This technique aims to find a state feedback control

$$u = \alpha(x) + \beta(x)v \tag{26}$$

as well as change variables

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} \dot{y} \\ \ddot{y} \\ \vdots \\ y^{n-1} \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{bmatrix} \tag{27}$$

that will transform eq. (20) to

$$\begin{cases} \dot{z} = Ax + Bv \\ y = h(T^{-1}(z)) \end{cases} \tag{28}$$

which is a linear system that can be controlled by appropriate control techniques.

Methods which utilizes feedback linearization has successfully been applied to a number of practical non-linear problems (Lemmen and Bröcker, 2000; Kwon et al., 2006). It can be used for both SISO and MIMO systems for stabilization and tracking control problems. On the negative side, feedback linearization cannot be used in all systems due to singularities, it also requires states to be measures which are not always possible. It also needs an accurate model because parameter uncertainty and not modeled dynamics will cause a non-robust controller (Hedrick and Girard, 2005, p. 136).

2.2.8 Sliding mode control

In sliding mode control (SMC), the main idea is to design a sliding surface, where a chosen parameter $\sigma = 0$, in the state space which divides the space into two disjoint subspaces ($\sigma < 0$ and $\sigma > 0$). Both subspaces then "direct" the system states towards that surface so they can "slide" on it towards desired control output, this is illustrated in fig. 6. Once the system state has joined the surface, it stays there. (Husson, 2009)

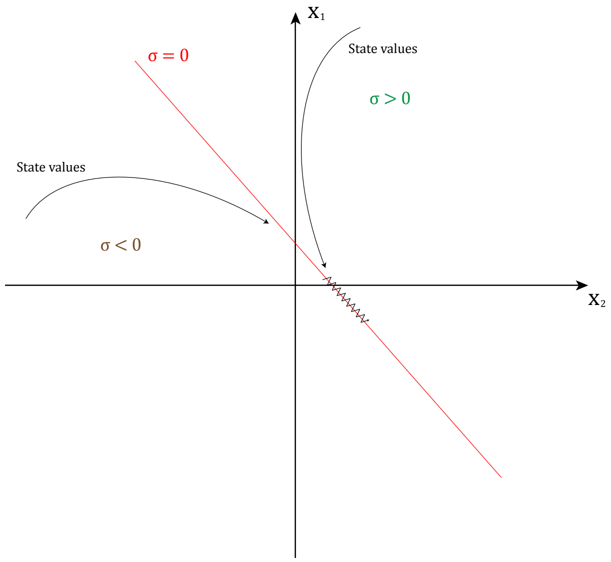


Figure 6. Illustration of the SMC.

The SMC has a variable structure, i.e. it's discontinuous. It was a result of studies to find a controller which was insensitive to variations of the system. This is also why the controller can be designed for both linear and non-linear systems. (Perruquetti and Barbot, 2002)

One condition for a sliding mode to exists (that the controlled system is stable) is that the sliding surface is attractive. This basically means that the system states always should move towards the sliding surface. Another condition for a sliding mode to exist is that system equations should meet the Lipschitz condition which guarantees that a unique solution exists for all states.

2.2.9 Model reference adaptive control

Model reference adaptive control (MRAC) is a control approach that adjust the control parameters to make the physical plant follow a specific output given by a reference model with the same input. This approach consists of three parts, see fig. 7, a reference model to give the desired response of the system, a controller of some sort with adjustable parameters and lastly an adjustment mechanism that is used to update the controller parameters to allow the physical plant to follow the reference model. (Jain and Nigam, 2013)

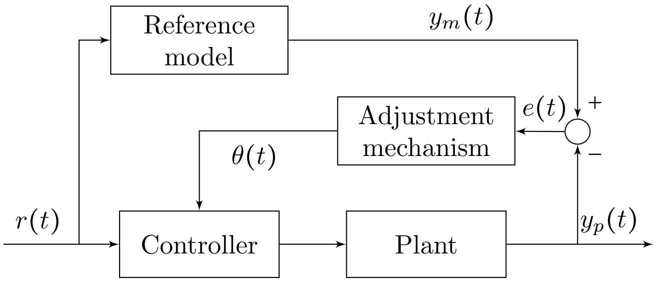


Figure 7. Inner and outer loop for the MRAC approach

There are many different methods of how to adjust the control parameters, one of them is the MIT rule that is built around the assumption that the adjustable parameters change more slowly than other variables within the system. (Vinagre et al., 2002) This MIT rule, by Jain and Nigam (2013), works by introducing a cost function

$$J(\theta) = \frac{e^2}{2} \tag{29}$$

where e is the error between the physical plant and the reference model output and θ is the parameter that is to be adjusted. The idea is to adjust θ in a way that allows the cost function to be minimize to zero. That means that the change of θ will follow the negative gradient of J and can be written as

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} \tag{30}$$

which combined with eq. (29) gives

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta} \tag{31}$$

where γ is the adaptive gain of the controller.

The partial equation $\frac{\partial e}{\partial \theta}$ is called the sensitivity derivative and indicates how the error e change with respect to θ. To implement this theory for a specific system, the error between the plant and the reference model is written as the difference between the two transfer functions.

$$E(s) = KG(s)U(s) - K_0 G(s) U_c(s) \tag{32}$$

where K is an unknown parameter, $G(s)$ is the transfer function for the system, $U(s)$ is the control signal, K_0 is a known parameter and $U_c(s)$ is the reference signal.

To introduce the controlled parameter θ into the equation a control law is defined as

$$u(t) = \theta u_c \tag{33}$$

which gives the following combined with equation eq. (32)

$$E(s) = KG(s)\theta U_c(s) - K_0 G(s) U_c(s). \tag{34}$$

Using the sensitivity derivative and then restate the equation in terms of model output gives the following equation

$$\frac{\partial E(s)}{\partial \theta} = KG(s)U_c(s) = \frac{K}{K_0} Y_m(s) \tag{35}$$

where $Y_m(s)$ is the model output.

From this, the MIT law can be concluded which gives an expression for updating the controlled parameter θ as

$$\frac{\partial \theta}{\partial t} = -\gamma e \frac{K}{K_0} y_m = -\gamma\prime e y_m \tag{36}$$

where $\gamma\prime$ is the adaptation gain.

Choosing the adaptation gain is a crucial part of the control approach and is the basis to get a good system response. Often this value is to be kept relatively low due to experience from some industrial plant has shown that large values might cause instability of the system.

2.3 Non-model-based control

In non-model based control (NMBC) no mathematical models are used for the control design. Instead, the parameters of the controller are mainly determined by using human experience and trial-and-error. NMBC design might be useful when the plant is too complex to model sufficiently, as no dynamic equations or linearizations are needed. As said however, a lot of knowledge and experience are needed to make a suitable controller. (Keshmiri et al., 2012, p. 16)

As NMBC doesn't depend on a model, it's flexible to changes in load or set-point values. The control parameters can be made self-tuning which in some cases makes the NMBC out-perform the MBC, especially in more complex dynamic systems.(Wang, 1997, p. 837)

Some of the different NMBC methods are described further below.

2.3.1 Fuzzy control

In fuzzy control there is a more significant emphasis on the use of heuristics. There is a focus on the use of rules to represent how to control the plant rather than ODE. This approach can offer some advantages in that the representation of knowledge in rules seems more lucid and natural to some people. For others, though, the use of differential equations is more clear and natural. Basically, there is simply a language difference between fuzzy and conventional control: ODEs are the language of conventional control, and rules are the language of fuzzy control. (Passino and Yurkovich, 1998, p. 11)

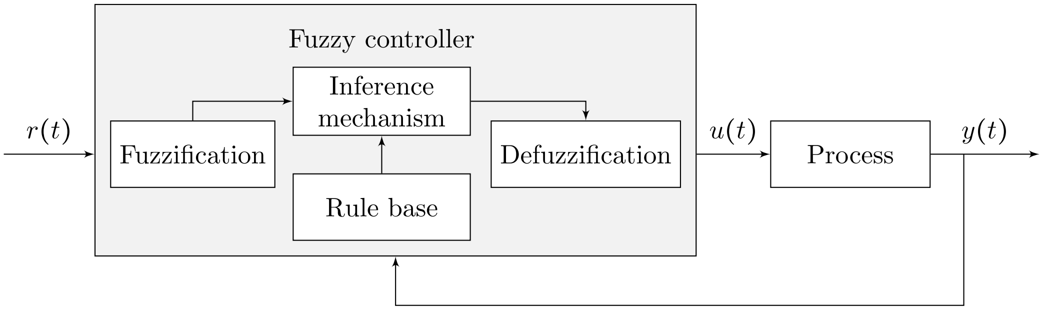


Figure 8. Fuzzy controller architecture

Passino and Yurkovich (1998, p. 11) argue that fuzzy control provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system. The controller has four main components:

1. The "rule-base" holds the knowledge, in the form of a set of rules, of how best to control the system.
2. The inference mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be.
3. The fuzzification interface simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base.
4. The defuzzification interface converts the conclusions reached by the inference mechanism into the inputs to the plant.

To design the fuzzy controller, the control engineer must gather information on how the artificial decision maker should act in the closed-loop system. Sometimes this information can come from a human decision maker who performs the control task, while at other times the control engineer can come to understand the plant dynamics and write down a set of rules about how to control the system without outside help. These "rules" basically say, "If the plant output and reference input are behaving in a certain manner, then the plant input should be some value." A whole set of such "If-Then" rules is loaded into the rule-base, and an inference strategy is chosen, then the system is ready to be tested to see if the closed-loop specifications are met. Generally speaking, if a very detailed rule-base is implemented, the chances of obtaining a better control performance is enhanced. (Passino and Yurkovich, 1998, pp. 12–13)

2.4 Hydraulic control strategies

In Bech et al. (2013), several control structures are studied for tracking of a hydraulic robot arm to find the optimal control strategy. Both linear and non-linear controllers are analyzed. Results show that pressure feedback and state space control increases the performance over classical linear structures, while passive feed-forward (PFF) control gives the best performance of the linear controllers. However, all the non-linear control designs shows superior performance to the linear ones. Findings show that all the non-linear control structures have relatively small root mean square (RMS) errors compared to the linear ones. These include SMC, adaptive inverse dynamics control (AIDC), AIDC combined with a PI-controller and MRAC.

Bonchis et al. (2002) also compares several different control structures on a similar hydraulic system, mainly with position accuracy and robustness in mind. Same as above, findings show that non-linear controllers outperform the linear designed controllers. Best result was given by acceleration feedback control (AFC) using an experimentally identified friction model, MRAC, and variable structure with sliding mode (VSC).

In the analyses, Bonchis et al. (2002) derives three characteristics for an ideal hydraulic controller. These are

- Online self-adjusting controller. Controller changes calculations and parameters on-the-go depending on some adjustments algorithm. Such as adaptive controllers or auto-tuning PID.
- Low number of parameters to adjust. This is to save computational power. For example, pole placement control (PPC) has no parameters to adjust and MRAC has only the adaptation gain.
- Existence of guidelines for parameter settings. Some guidance on how to select the control parameters. In for example MRAC, the initial trajectory parameters is crucial for good control performance.

However, results also show that self-tuning controllers are not suitable when plant parameters changes quickly.

Kalmari et al. (2014) uses a non-linear model predictive control (NMPC) to control a forestry hydraulic crane. Experiments show that NMPC is a feasible controller for that specific purpose. The NMPC needs a lot of computational power since it always estimates the future states and considers them together with the current state to calculate the optimal controller.

Overall, research suggests that the non-linear control is the way to go for complex systems like hydraulics which is the case in this project. Further investigations have to be made before any decision on exactly which approach to go with. However, this investigation implies that both NMPC and MRAC should be considered.

2.5 Model and control validation procedure

A model is a one-out-of-many possible representations of a real world system. Model validation, is the procedure of making sure the model is valid, i.e. correspond to the real system to a reasonable degree which includes critically assessing the model, tweaking it, and adding or removing parts where necessary.

The procedure for follows the process proposed by Ljung and Glad (2004, pp. 389–396) where building and validating the model is iterative. It starts with a proposed model, which may then be approximated or simplified. Parameters that are already specified by data sheets are used, and others are initially estimated. By measuring the real system, parameters can be fitted to the data. After this, the model may be validated and the process is then repeated from any of the previous steps as deemed necessary by the results. An illustration of this process can be found in fig. 9 below.

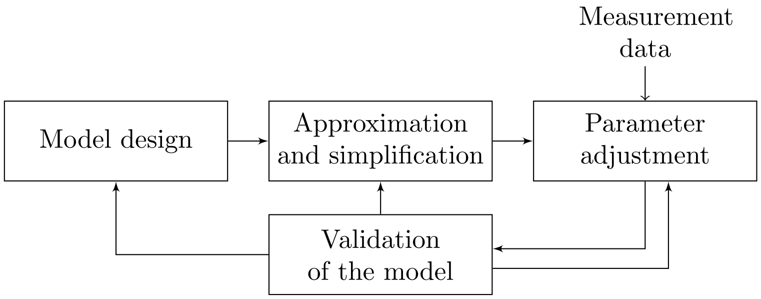


Figure 9. Flow chart depicting the iterative modeling procedure.

Validation of the model, i.e. the block on the bottom of fig. 9, is done simply by using the same input signals to the model as well as the actual system. The various output signals as recorded by the different sensors are then compared to the outputs of the model. If the output residuals (prediction errors) are within the margin specified by the requirements, the model is considered to be valid. This process is illustrated in fig. 10.

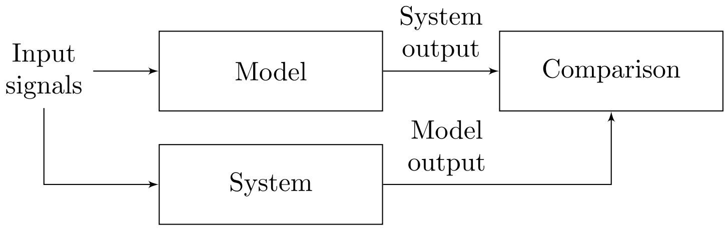


Figure 10. Flow chart depicting the model validation procedure.

Every model has a certain validity region, for instance, around a specific operating point — and should never be used outside of that validity region. For this reason, it may be necessary to have several models for different validity regions, e.g. different load cases. (Ljung and Glad, 2004, pp. 394–396)

There are several ways of actually validating the model and there are several factors that must be taken into account. Generally, the model should be "trained" with one set of input/output signals, and then evaluated with another set of data. Ideally, the model should have residuals that are not dependent on the input signal. If it does depend on the input, the system dynamics is not fully described by the model. (Ljung and Glad, 2004, pp. 363–368)

A process similar to the description above can be used to validate the velocity controller, by comparing the controller output and the behavior of the crane to the specifications. In such a case, the controller parameters and structure are modified until desired behavior has been accomplished.

2.6 Parameter optimization

In order to attempt to estimate the correct parameters of the model, different methods for parameter tuning were considered. As the model was complex and included a large number of unknown parameters that were initially estimated by guess, a trial-and-error approach was infeasible due to the large number of combinations as well as the unintuitive correlation between the parameters and the results. This section focuses on the application of machine learning algorithms to error between the modeled and the measured data. This is generally done by minimizing a cost function which describes said error.

2.6.1 Gradient descent

The gradient descent method (also known as steepest descent) is an algorithm that uses the gradient of a quadratic performance surface $E(\Theta)$ in order to find the parameters that minimizes the error. The quadratic performance surface is often implemented as a mean square error (MSE). In order to minimize this, the algorithms tries to find the global minimum by following the path of the steepest descent. This algorithm requires initial estimated values for all the parameters being tuned as well as a value for the step size α. It uses an iterative process where the parameters, Θ are updated as

$$\Theta_{k+1} = \Theta_k - \alpha \nabla(E(\Theta_k)). \tag{37}$$

The convergence capability of this algorithm depends strongly on the initial values of the parameters as well as the step size. It is possible for the algorithm to get stuck on a local minimum, in which case it is necessary to change the initial guesses. (Monzingo et al., 2011)

2.6.2 Output error method

This methods is based on the assumption that only the measured data contains noise. This noise is also assumed to be zero-mean and normal distributed. The cost function to be minimized in this case is

$$J = \frac{1}{2}\sum_{k=1}^{N}[z(k) - y(k)]^T R^{-1}[z(k) - y(k)] + \frac{N}{2} ln|R| \tag{38}$$

where $z(k)$ and $y(k)$ is represent measured and the modeled data respectively. R is the covariance matrix corresponding to the noise of the measured data estimated as

$$\hat{R} = \frac{1}{N}\sum_{k=1}^{N}[z(k) - y(k)][z(k) - y(k)]^T \tag{39}$$

.

The process is conducted iteratively and the set of parameters Θ are updated as

$$\Theta(i+1) = \Theta(i) + [\nabla^2_\Theta J(\Theta)]^{-1}[\nabla^2_\Theta J(\Theta)] \tag{40}$$

where the first and second gradients, ∇ and ∇^2 are defined by Raol et al. (2004) as

$$\nabla_\Theta J(\Theta) = \sum_{k=1}^{N}[\frac{\delta y}{\delta \Theta}(k)]^T R^{-1}[z(k) - y(k)] \tag{41}$$

$$\nabla^2_\Theta J(\Theta) = \sum_{k=1}^{N}[\frac{\delta y}{\delta \Theta}(k)]^T R^{-1}[\frac{\delta y}{\delta \Theta}(k)] \tag{42}$$

2.6.3 Linear regression

Regression tools can often be used when estimating parameters. The commonly used linear regression aims to fit modeled data to observed data by creating a linear relationship between the response variable and the auxiliar/independent variable. The main idea is to find a set of parameters $(A_1, A_2, \ldots, A_n)$ that will minimize the error between the observed data and the modeled function $\hat{Y}$ defined as

$$\hat{Y} = A\vec{x} \tag{43}$$

where A is a vector containing the parameters and

$$\vec{x} = [x^0, x^1, \dots, x^{n-1}]. \tag{44}$$

The object function, i.e. the function to be minimized in for the parameter optimization is often described with the sum of square error, that is

$$\Phi = \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2 \tag{45}$$

which makes the optimization problem to be solved

$$A = argmin_A \Phi \tag{46}$$

This approach can easily be used to estimate parameters used when the model resembles a linear system, and is very computationally efficient as the parameters can be calculated analytically. (James et al., 2013)

2.6.4 Nonlinear least square regression

Although many systems can be described reasonably well using linear regression, it is sometimes preferable to improve the freedom of design. Nonlinear regression is an extension of the linear regression method that allows the user to fit almost any function on the basic form

$$y = f(\vec{x}, \vec{\beta}) + \varepsilon \tag{47}$$

where y is the response, $\vec{x}$ is a vector containing the control variables and β is a set of unknown parameters. ε is the error, assumed to be independently, identically and normally distributed with a mean zero and constant variance. The estimated parameter $\hat{\beta}$, which is based on least squares, aims to minimize the function

$$S(\beta) = \sum_{j=1}^{n} (y_j - f(x_j; \beta))^2. \tag{48}$$

where n is the number of samples. The estimation of β is done through a method based on gradient descent, that is

$$f(x; \beta) \approx f(x; \hat{\beta}) - \nabla_\beta f(x; \beta)|_{\beta = \hat{\beta}} \tag{49}$$

The main disadvantage of using a nonlinear regression method is that, unlike linear regression, the parameter values can't be calculated analytically. The values are instead calculated numerically through an iterative process which is inherently computationally costly. (Daimon and Goto, 2008).

2.6.5 Nelder-Mead Simplex search

The Nelder-Mead algorithm is a widely used method for optimization that attempts to minimize a scalar valued nonlinear function of n real variables using only function values, i.e. it does not need any derivative information. This makes it a very useful method for estimating complex non-analytic functions. The method consists in maintaining a non-degenerate simplex in R^n of non-zero volume described by $n+1$ vertices.

The algorithm starts with a simplex with n+1 vertices, the first of which is usually a guess from the user and the remaining are calculated from this. The next step is to order the cost function values for the different vertices so that

$$f(\vec{x}_1) \le f(\vec{x}_2) \le ... \le f(\vec{x}_{n+1}) \tag{50}$$

were $f(\vec{x}_n)$ is the cost function value for a vertex x_n. After this, the reflection point $\vec{x_r}$ is calculated as

$$\vec{x_r} = \sum_{i=1}^{n} x_i/n + \rho(\vec{x}_m - \vec{x}_{n+1})) \tag{51}$$

where $\vec{x}_m$ is the centroid of the n best points, that is, simply excluding the worst point. The next step of the process consists in evaluating the cost functions value for $\vec{x}_r$, i.e. f_r. The following step will be dependent on the value of f_r in relation to f_1 and f_n, i.e. the function values for the best and the worst vertices. If

$$f_1 \le f_r \le f_n \tag{52}$$

then the iteration will be terminated. If

$$f_r \le f_1 \tag{53}$$

an expansion point x_e is calculated and its cost function value will be compared with f_r and the best value will be selected. If the function value

$$f_r > f_n \tag{54}$$

the cost function value f_r will be compared to the cost function value of a contracted vertex, f_c and the most appropriate vertex is chosen. After this, all vertices except for x_1 will be replaced as

$$x_i = x_1 + \sigma(x_i - x_1) \tag{55}$$

where σ is the shrink coefficient. (Lagarias et al., 1998)

2.6.6 Pattern search (with global optimization toolbox)

The Hooke and Jeeves' pattern search uses information gained from previous iterations in order to calculate a new one. If a previous iteration was successful, i.e. the cost function value decreased, then the next iteration will be more likely to be successful if a step is taken in the same direction. The algorithm consists of taking exploratory moves which are conducted by changing the value of a single parameter at the time. The following move is called the pattern move which is when the inferred from the exploratory moves in order to find the step that will minimize the cost function the most. This procedure is particularly useful for optimizing complex, non-linear functions as it does not require any derivatives to be calculated. (Torczon, 1997; Hooke and Jeeves, 1961)

Chapter 3 Preparation of the system

One of the project's main tasks was to prepare the crane so testing of controllers and verification of the models could be done as smooth as possible. In this chapter this preparation will be described in terms of hardware improvements and software implementations.

3.1 Hardware overview

In this section, a general overview of the different subsystems that were used during the work with the crane is presented. The crane consists of two larger subgroups: the hydraulic and the electrical system. The crane is controlled and interfaced by either one of the two systems: The IFM system or the dSPACE system. A system overview is described in table 3 and a schematic illustration is also provided in fig. 11.

Table 3. A detailed overview of the components related to the crane

Hydraulic system	
Mobile directional hydraulic valve	The Parker P70CF is an open center directional valve. It is designed to be installed with a constant flow pump. It is a stackable system, where various amounts of spool valves can be chosen, depending on the amount of functions wanted. On the test crane there are four spool valve sections in use. The spools mounted are of ECH-type, meaning they are pressure compensated. The spool valves are of different geometry, due to different cylinder characteristics (e.g. area, load).
Hydraulic actuators (4x)	One hydraulic cylinder for each boom, plus slew actuator.
Counterbalance valves	On each cylinder there is a counterbalance valve. On the first boom there is a Hiab single acting 395-8078. On the second boom and extension there is a Hiab 396-2768 and Hiab 193-7171 which are double acting counterbalance valves.
Hydraulic pump with tank (100 L)	Pump assembly made by Hydronova. Consists of a tank and a Vickers pump (PVB10-RSY-31-C-11) powered by a Nel 160M4 11kW three-phase electric motor.
Electrical system	
Inclination sensors	Two IFM JN2100 inclination sensors mounted on the second boom and extension. They are interfaced via CAN and have a specified accuracy of ±0.5°.
Optical distance sensor	The IFM 01D100 optical distance sensor is mounted on the extension. It measures the elongation of the extension boom.
Pressure sensors	One IFM PT3551 with analogue output, is measuring the gallery pressure. There are three Hiab (Hydac) 395–7888 pressure sensors mounted on both sides of the first boom cylinder and one side of the second boom cylinder.
Spool position sensors	There is a spool position sensor attached to each spool on the hydraulic valve.

Flow sensor	The Hydac Electronic EVS 3100 flow sensor measures the input flow to the crane (only available together with dSPACE).
	The IFM System
IFM ECU	The IFM CR7032 is the main processing unit on the crane. It is a programmable mobile controller, with safety controller functionality, meaning it is designed for safety applications. It executes code in sequence without interrupts making it easier to assure deterministic behavior. The ECU has several different inputs - analogue, digital and CAN. The task of the ECU is to take sensor input data and depending on the program implemented, actuate the spools. CODESYS PC software is used to program the ECU through a USB2CAN-adapter (CANFox).
XS-drive	The XS-drive is a wireless unit for controlling the crane.
Display module	The IFM CR0451 is a CAN-enabled display module. It can be programmed to display specified values broadcasted on the network. The pushbuttons can be enabled to provide input, eg. switch between operation modes.
WiFi CAN bridge	The Electrum 253397-C module is a bridge between the CAN and WiFi, it broadcasts messages on the bus via WiFi and enables logging of the CAN directly to a computer.
IFM power supply	The IFM DN4013 power supply is rated for 10A at 24V and powers the XS-drive and IFM Controller.
	The dSPACE System
MicroAutoBox	The dSPACE MicroAutoBox 1401/1511 is a ECU for performing fast function-prototyping. It has an embedded PC-extension and is also equipped with a WiFi-link (not used). It is connected to a host computer with dSPACE ControlDesk through a Real-time interface.
RapidPro System	The RapidPro box consists of a control unit and a signal conditioning unit. This hardware acts as a subsystem to the MicroAutoBox and handles the analog interface to the sensors and actuators on the crane.
Cinfa power supply	Two Cinfa DRA240-24 power supplies rated for 10A at 24V are used to supply the dSPACE hardware.

3.1.1 Flow sensor installation

A Hydac Electronic EVS 3100 flow sensor, the datasheet can be seen in Appendix A, was acquired to measure the inlet flow to the system in real time. This, to get a better insight about the hydraulic flow dynamics and to verify that the flow is constant. The flow, which was unknown, was observed to have very high impact on system behaviour, therefore installation of a flow sensor was considered vital.

The sensor has a specified flow range of 15–300 L/min and a maximum pressure of 400 bar. It was fitted between the pump and the crane input. The sensor is fed with 24 V from the positive terminal block (supplied from 24 V Cinfa power supply) to input pin 2 on the flow sensor. The sensor then outputs a signal between 4–20 mA from pin 3 which is fed to analog input channel 6 in the dSPACE RapidPro Control unit. The sensor is interfaced directly to dSPACE, therefore currently flow measurements can only be obtained using the dSPACE system.

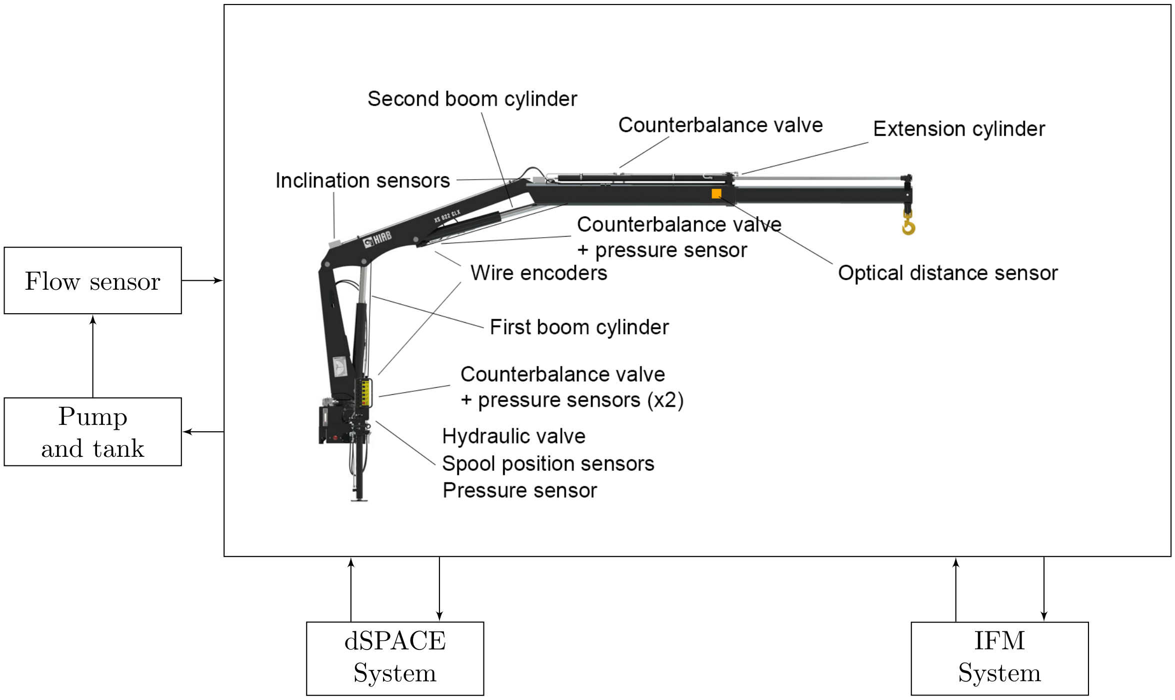


Figure 11. System overview flowchart

3.1.2 Pump installation

The pump first used to supply the crane was a Parker PV016, which is a constant pressure, variable displacement pump. This is not optimal because the crane should be supplied with a constant flow. To get around this problem the pump was set to 250 bar, where it could deliver 94.5 L/min, and equipped with a flow control valve (Hydac SR10-01) between the pump and crane. The valve builds up the pressure and restricts the flow to approximately 30 L/min. This solution is simple but does however constantly waste a lot of power which is transferred to heat at the flow control valve. Also, the flow control valve output range is between 13–38 L/min, which was adjusted with a manual knob, hence very difficult to tune without feedback. When the flow sensor was installed and the flow could be monitored it appeared to fluctuate significantly (30 ± 15) L/min when activating different functions, especially the second boom extension cylinder. A plot of the flow when running the second boom extension with this pump is displayed in fig. 12 below.

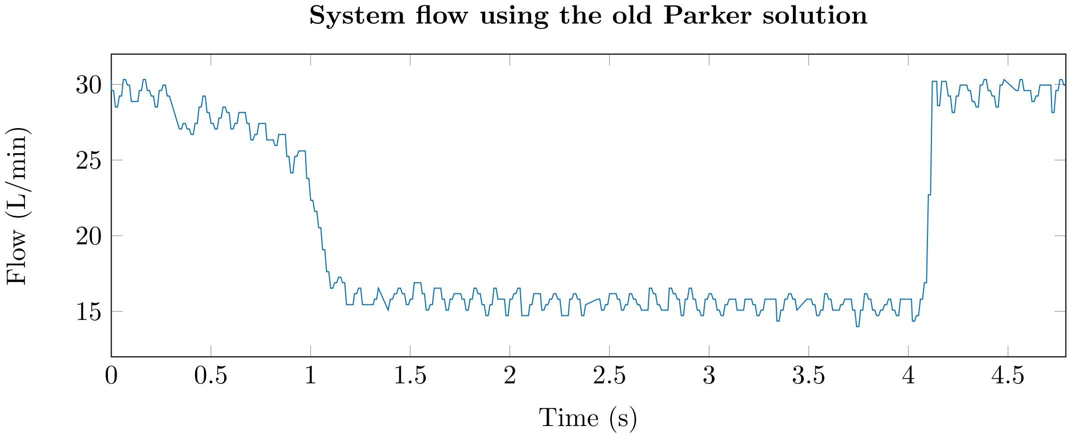


Figure 12. The flow to the crane when running the second boom extension with the Parker PV 016 pump.

The drawn conclusion from this was that the current oil supply solution was way off the expected and required behavior. Therefore a better suited hydraulic pump had to be installed.

The solution was a hydraulic unit from Hydronova, see Appendix B, specified for a maximum 30 L/min of flow and a maximum pressure of 200 bar, but still a constant pressure pump. The pump was installed directly to the crane, bypassing the flow control valve and the pump displacement was adjusted to output it's maximum pressure 200 bar, and the pressure relief valve adjusted to slightly above 200 bar which is the crane specified working pressure. The idea with this pump was to let it feed 30 L/min oil through the valve when not running any functions, while it tries to build up the system pressure. When one is using a function, resistance increases in the system and the pump can then start building pressure. Since the maximum flow of the pump is rated to 30 L/min, this would give approximately the same behavior as a fixed displacement pump with only a slight delay when it has to build pressure. The flow using the new setup can be seen in fig. 13

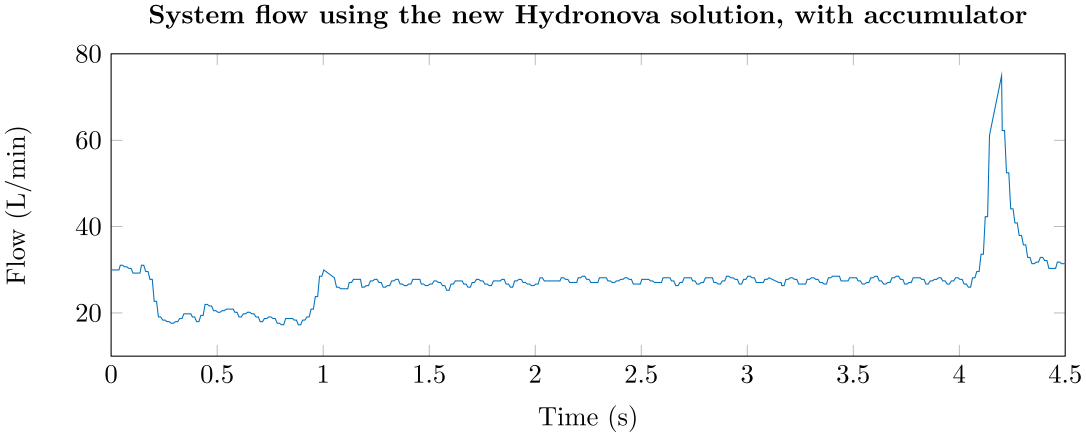


Figure 13. The flow to the crane when running the second boom extension with the new Hydronova solution with the Vickers pump.

As one can see, there is a big delay in the pressure build up phase and also a pressure peak when the spools return to their neutral position. The pump unit was housing a hydraulic accumulator, and it was the cause for the negative impact on system performance. The accumulator induced a delay in the system pressure, resulting in too low flow during valve actuation. When no crane function is active, the fluid is pumped through the system, and the pressure is low, it is low also in the accumulator. When a function is activated the pump has to build up pressure in both the accumulator and the crane lines, thus increasing time for building pressure and inducing large flow oscillations. Since this negative behavior occurred because of the accumulator, it was removed. The resulting improved flow characteristics with the new pump without the accumulator can be seen in fig. 14 below.

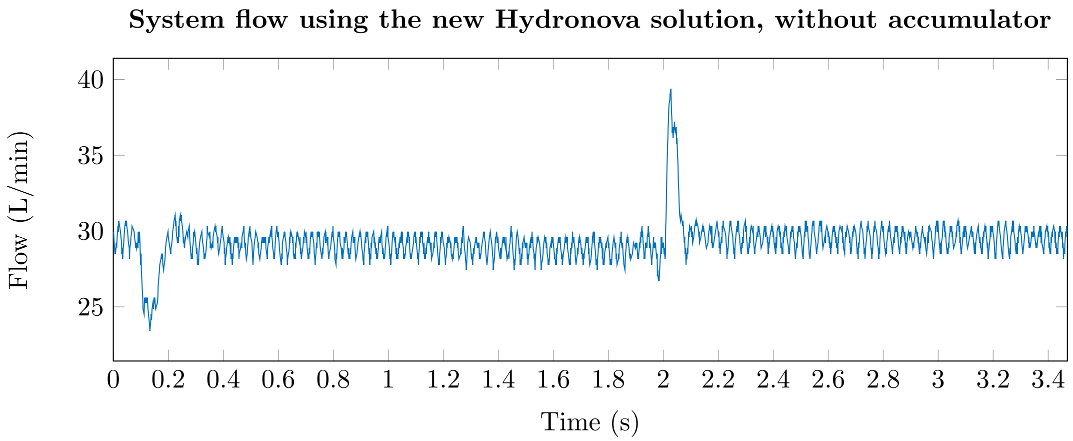
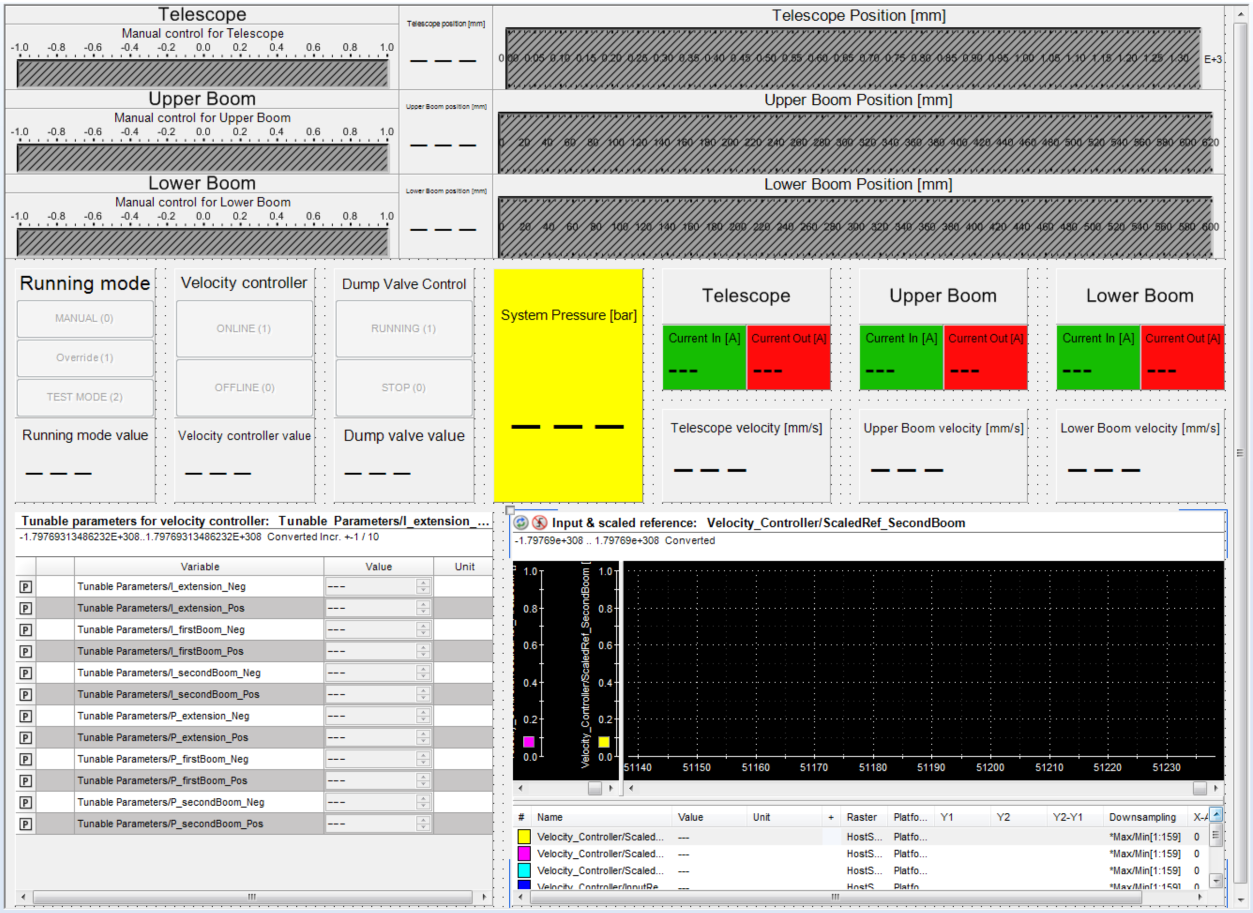


Figure 14. The flow to the crane when running the second boom and extension with the Vickers pump without the accumulator

After bypassing the accumulator tank the flow spikes decreased. The two deviating values one can see after 0.1 s and 2 s are due to spool movement. The first is when the spool is starting to move and the pump builds pressure, and the second when the spool returns to its initial position, inducing a peak. These now much smaller variations were estimated to be of similar order as on a setup in a mobile application.

3.2 dSPACE Control Desk setup

A user interface has been designed using dSPACE Control Desk. As seen in fig. 15, it can be run in three modes, with MANUAL as primary. The other modes available are OVERRIDE and TEST MODE. When the system is started the dump valve has to be set to RUNNING to run the system.



Tunable parameters for velocity controller: Tunable Parameters/I_extension_...

-1.79769313486232E+308..1.79769313486232E+308 Converted Incr. +-1 / 10

		Variable	Value	Unit
P		Tunable Parameters/I_extension_Neg	---	
P		Tunable Parameters/I_extension_Pos	---	
P		Tunable Parameters/I_firstBoom_Neg	---	
P		Tunable Parameters/I_firstBoom_Pos	---	
P		Tunable Parameters/I_secondBoom_Neg	---	
P		Tunable Parameters/I_secondBoom_Pos	---	
P		Tunable Parameters/P_extension_Neg	---	
P		Tunable Parameters/P_extension_Pos	---	
P		Tunable Parameters/P_firstBoom_Neg	---	
P		Tunable Parameters/P_firstBoom_Pos	---	
P		Tunable Parameters/P_secondBoom_Neg	---	
P		Tunable Parameters/P_secondBoom_Pos	---	

Input & scaled reference: Velocity_Controller/ScaledRef_SecondBoom

-1.79769e+308 .. 1.79769e+308 Converted

#	Name	Value	Unit	+	Raster	Platfo...	Y1	Y2	Y2-Y1	Downsampling	X-A
🟨	Velocity_Controller/Scaled...	---			HostS...	Platfo...				*Max/Min[1:159]	0
🟪	Velocity_Controller/Scaled...	---			HostS...	Platfo...				*Max/Min[1:159]	0
🟦	Velocity_Controller/Scaled...	---			HostS...	Platfo...				*Max/Min[1:159]	0
🟦	Velocity_Controller/InputRe...	---			HostS...	Platfo...				*Max/Min[1:159]	0

Figure 15. Graphical user interface in Control Desk

In the manual mode the input signal passes through a velocity controller and a current controller. The velocity controller can be either switched ONLINE or OFFLINE. When the velocity controller is switched ONLINE, the input signal is scaled by the velocity controller. By dragging the sliders between (-) 1 and (+) 1 the operator controls the valve on each cylinder. For the first boom and second boom the value (+) 1 corresponds to full speed in positive direction and (-) 1 to full speed in negative direction. For the extension, a positive input curtails the length of the boom and vice versa.

In the test mode the operator can load a file consisting of predefined inputs for each function. In this mode all inputs go through the current controller in the same manner as described in the previous sub-section. The velocity controller can either be switched ONLINE or OFFLINE according to the purpose of the test. The moment the operator presses the button TEST MODE the system starts to run. Hence, the operator has to run the crane to its initial position before starting the test. When the test case ends, the crane stops.

The override mode allows the operator to operate the crane without any controllers. This correspond to the operator running the crane with the manual levers. In this mode the input signal is feed directly into the actuator block in form of a duty cycle. Therefore, the crane can only be controlled with the slide controls in this mode.

The tunable parameters are used to tune the P and I part of the velocity controller. Each function has two proportional gains and two integral gains, one for each direction of movement, which can be set individually.

Further, several different sensor values are showed in order to monitor the system. This includes current, cylinder position, spool position. These indicators, as well as plots, can easily be exchanged and modified by using standard library components available in dSPACE ControlDesk.

3.2.1 Sensor data conversion

Compared to the commercial setup (where the crane is only equipped with a remote control system connected to electric spool actuators as well as some pressure sensors for safety reasons) the project setup is equipped with a number of extra sensors operating on either the CAN-bus or connected on separate 12 V or 20 mA connections to the dSPACE hardware. These sensors are connected for the purpose of logging relevant data for both verification and as well as development of the controller. Below in table 4 is a list of how the sensors used within the project are interfaced.

Table 4. List of electronic sensors

Sensor	Quantity	Range		Output		Originally mounted
Flow sensor	1	15–300	L/min	4–20	mA	No
Optical distance sensor (IFM 01D100)	1	0.2–10	m	4–20	mA	No
Pressure sensor (Hiab 395-7888)	3	0–42.5	MPa	4–20	mA	Yes
Pressure sensor (IFM PT3551)	1	0–25	MPa	4–20	mA	No
Spool position sensor	8	(-6)–6	mm	1 – 4	V	Yes
Wire encoder (Hiab FSG)	2	0–3	m	CAN		No

All sensors, except the spool position sensors and wire encoders produce analogue signals of 4–20 mA, which then are AD-converted to a value between 0 and 1. The sense resistors in the dSPACE hardware are 240 Ω. All the measured values have been converted to appropriate units and some of them were also filtered to be of practical use. Below are descriptions of relevant characteristics of each sensor as well as other considerations taken into account during interfacing.

Flow sensor (ESV 3100)

The flow sensor has a range of 15–300 L and outputs an analogue signal of 4–20 mA. Hence, $Q_{Measured}$ is converted as

$$Q_{Measured}[\text{L/min}] = \left(\frac{ADC_{Value} \cdot 5[\text{V}] \cdot 1000}{240\Omega} - 4[\text{mA}]\right) \cdot \left(\frac{300 - 15[\text{L/min}]}{(20-4)[\text{mA}]}\right) \tag{56}$$

Optical distance sensor (IFM O1D100)

The sensor has a range of 0.2–10 m and outputs an analog signal of 4–20 mA. Hence, $L_{Measured}$ is converted as

$$L_{Measured}[\text{mm}] = \left(\frac{ADC_{Value} \cdot 5[\text{V}] \cdot 1000}{240\Omega} - 4[\text{mA}]\right) \cdot \left(\frac{1340[\text{mm}]}{(20-4)[\text{mA}]}\right) \tag{57}$$

Since the controller uses the velocity of the extension, this signal also has been derived. This has been done by a derivative block in Simulink. As the resulting signal was observed to be very noisy, a low pass filter was implemented. The obtained noise reduction was unsatisfactory and hence alternate methods of filtering the signal has been looked into. The use of a Kalman Filter with a high value for measurement noise covariance provided a filter that was sufficiently devoid of noise. The process noise covariance and measurement noise covariance constants define the level to which the signal can be trusted and how much change is expected and hence eliminates noise. The high value for the measurement noise covariance implies that the system expects a lot of noise in the signal it receives and hence compensates for the same while filtering. The model used to compare the signal to is of constant velocity, this implies that the signal produced adjusts to a change in input relatively slowly. Changing the values of the above mentioned two constants can help tune the reliance of the system on the model and also predict the level of noise expected in the input signal. The higher the constant, the more the model is trusted, and as a constant velocity model is used, the slower the reflection of changes in input.

Pressure sensors (Hiab 395–7888)

The pressure sensors range between 0–42 MPa and produce an analogue output signal of 4–20 mA. Hence, $p_{Measured}$ is converted as

$$p_{Measured}[\text{bar}] = \left(\frac{ADC_{Value} \cdot 5[\text{V}] \cdot 1000}{240\Omega} - 4[\text{mA}]\right) \cdot \left(\frac{425[\text{bar}]}{(20-4)[\text{mA}]}\right) \tag{58}$$

Pressure sensor (IFM PT3551)

The sensor has a range of 0–25 MPa and produce an analogue output signal of 4–20 mA. Hence, $p_{Measured}$ is converted as

$$p_{Measured}[\text{bar}] = \left(\frac{ADC_{Value} \cdot 5[\text{V}] \cdot 1000}{240\Omega} - 4[\text{mA}]\right) \cdot \left(\frac{250[\text{bar}]}{(20-4)[\text{mA}]}\right) \tag{59}$$

Spool positions sensor

Since the valves are not identical, the values used for conversion have been measured and estimated for each one of them. Below in table 5 are the corresponding ADC-values for each sensor for 6 mm, 0 mm, −6 mm.

Table 5. List of values for spool postion sensors

Sensor Position	$ADC_{Value+6mm}$	$ADC_{Value0mm}$	$ADC_{Value(-)6mm}$	Δ	Offset
Lower Boom	0.40	0.27	0.14	46.67	(-)12.66
Upper Boom	0.39	0.26	0.13	45.45	(-)11.77
Extension	0.39	0.26	0.13	45.45	(-)11.72

The Δ value has for each sensor been calculated as

$$\Delta[\text{mm}] = \frac{12}{ADC_{Value+6mm} - ADC_{Value(-)6mm}} \tag{60}$$

The offset value for each sensor has been calculated as

$$\text{Offset[mm]} = \Delta\text{[mm]} \cdot ADC_{Value+0mm} \tag{61}$$

Hence, the ADC signal is converted as

$$L_{Measured}\text{[mm]} = ADC_{Value0mm} \cdot \Delta\text{[mm]} - \text{Offset[mm]} \tag{62}$$

This approach has been used because the data sheet for the spool positions sensors are missing in the documentation from earlier projects with the crane.

Wire encoders (Hiab FSG)

These sensors are interfaced via CAN and send absolute distance measurements as well as velocities.

3.2.2 Designing current controllers

In order to control the crane in an adequate manner, the solenoids have to be controlled. The solenoids will try to counteract the control signal both by their inductances, spring forces, and masses. By looking at the system signals, the datasheet values did not seem to correspond to the actual system. Moreover, several parameters were missing for the solenoids. In order to optimize the current controllers a good model is necessary. Using such a model, a current controller can be designed more easily and robustly in Simulink.

A solenoid can be modeled as a spool (with a resistance and inductance), and a mass connected to a spring with some viscous friction, i.e. a regular mass-spring-damper system. Any back-EMF was considered to be negligible. This results in a first degree model for the spool,

$$I(s) = \frac{V_{in}(s)}{R + Ls} \implies G_{spool}(s) = \frac{1}{R + Ls} \tag{63}$$

where V_{in} is the voltage input signal as caused by the pulse width modulation (PWM) signal (ranging from 0–1 as per the dSPACE default, yielding 0–24 V), I the resulting current, R is the spool resistance and L is the inductance. The mass-spring-damper system will have a second order model, i.e.

$$X(s) = \frac{K_f \cdot I(s)}{M_s s^2 + d_s s + k_s} \implies G_{msd}(s) = \frac{K_f}{M_s s^2 + d_s s + k_s} \tag{64}$$

where K_f is a constant transforming the current to a force, M_s is the solenoid mass, d_s is the viscous friction and k_s is the spring stiffness constant. The resulting solenoid model for the entirety of the solenoid can therefore be described using the product of the two transfer functions as

$$G_{sol}(s) = G_{spool}(s) G_{msd}(s). \tag{65}$$

It is also assumed that the solenoids are identical, and that any interference caused by the connection to different valves and the alike will have a negligible effect. By running the crane with pseudo-random input in order to cover as many different frequencies as possible, it is possible to approximate these transfer functions using MATLAB's system identification toolbox (`ident`).

A total of six minutes of data was collected for each solenoid and then modeled accordingly, the data set was split into two where the first was used for modeling and the second used for validation. The resulting transfer functions were

$$G_{spool}(s) = \frac{75.8}{s + 75.2} \tag{66}$$

and for the solenoid

$$G_{sol}(s) = \frac{5.6 \cdot 10^4}{s^3 + 11.4s^2 + 1222s + 7130} \tag{67}$$

Figure 16 below shows the result of the modeling of the spool (fitted on the second valve) compared to the validation data. The system appears to be able to include most of the dynamics associated with the spool, and with a fit of 95 % the model was deemed good enough to perform elementary control of the current.

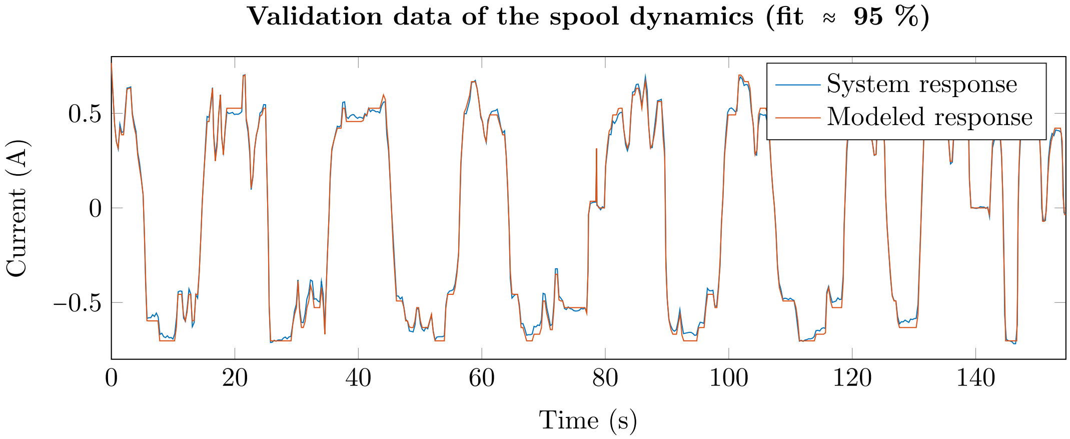


Figure 16. The modeled spool response from voltage to current and the corresponding validation data sets.

Figure 17 below shows the result of the corresponding model of the solenoid compared to the validation data. The fit here is slightly worse at approximately 82 %. This may be due to the fact that there are other external factors or disturbances that are not taken into account, e.g. the flow of the hydraulic oil, or that the input signal saturates the system, etc. However, if a solenoid position feedback controller were to be implemented, this model should capture the essential dynamics necessary to develop such a controller.

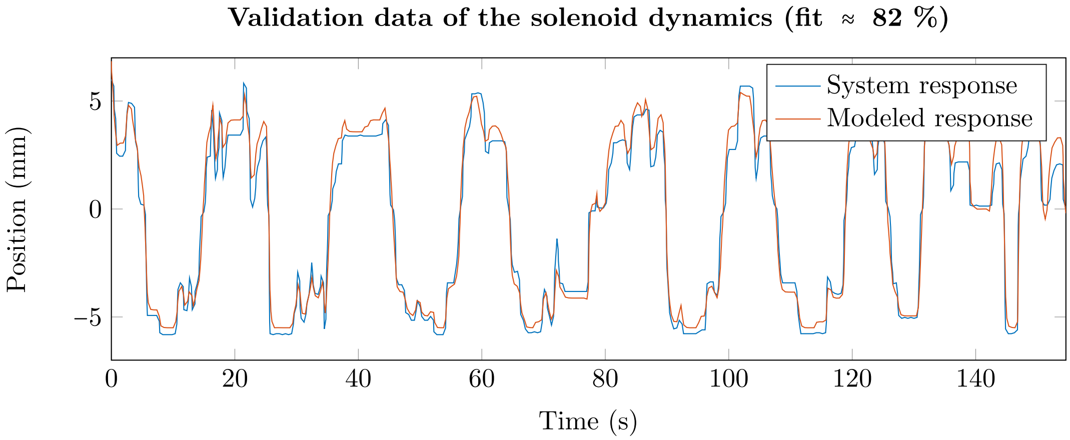


Figure 17. The modeled solenoid response from voltage to position and the corresponding validation data sets.

Additionally, comparing the two plots in fig. 16 and fig. 17 reveals that the relationship seems almost linear. Therefore, by assuming linearity, a current controller should suffice. The remaining solenoids followed the same pattern and the transfer functions were almost identical. Since they are considered very similar or identical, these results yield yet another assurance that the modeling seem to be accurate enough.

Given these models, a PI controller with back-calculation anti-windup was created. This controller follows the structure given by fig. 18, and values for the proportional gain $P = 2.87$, integral gain $I = 202.3$, and back-calculation coefficient $K_b = 0.01$. This resulted in a rise time (from 10 % to 90 % maximum current) of approximately 30 ms given a sample time of 1 ms.

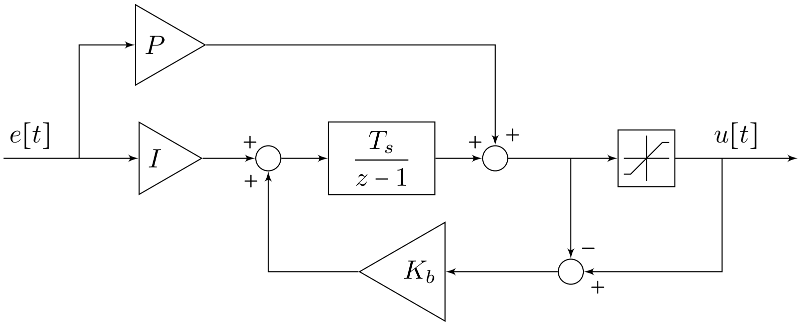


Figure 18. Structure of a discrete PI-controller with back-calculation anti-windup

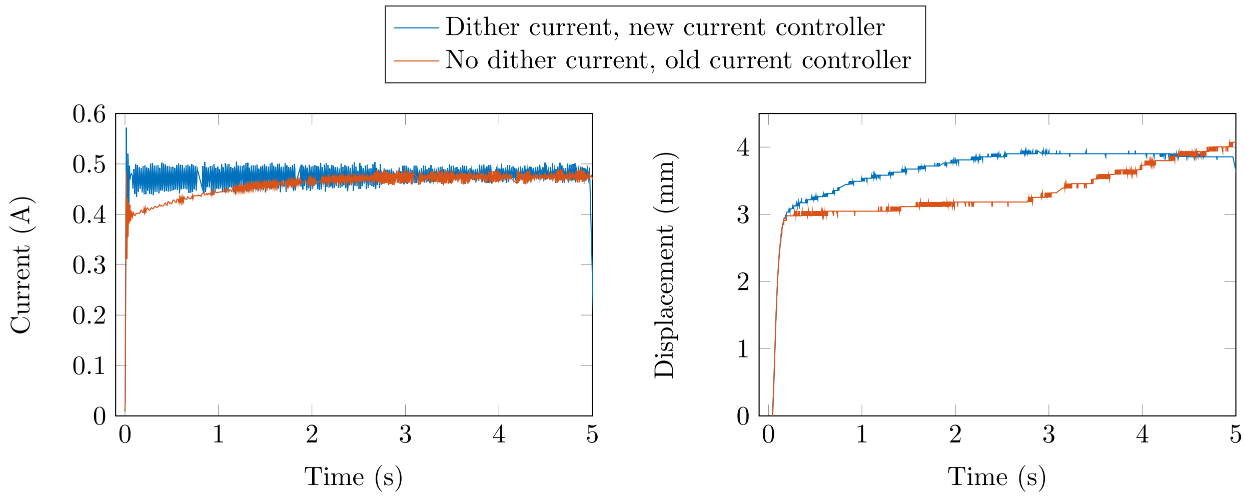
3.2.3 Superimposing dither currents

Since static friction (stiction) and hysteresis can make the behavior of the valve to seem erratic and unpredictable a dither wave was added to minimize these behaviors. Stiction keeps the spool from moving when input signal changes are small. When the signal finally becomes large enough to initiate movement, the spool will tend to overshoot the position required for accurate control. Hysteresis is the tendency for the spool movement to be different depending on whether the input signal is increasing or decreasing, even if the input signal is identical. To satisfy those requirements the amplitude must be set large enough and the frequency slow enough for the spool to respond. However, the signal should be small and fast enough to not create a noticeable pulsation in the hydraulic output of the valve.

The dither waves were first implemented with a specific amplitude and frequency using a square wave which alternated between each of the outputs ("positive" and "negative") to the spool. Practically, an if-statement was used in the Simulink model to determine which side the dither current should be applied to. By doing this, the spool continuously moves back and forth. A problem which occurred with this implementation was that the current controllers regulating the spool current. These were not fast enough to handle currents switching in frequencies as the dither waves required. In a conversation with Hiab about this problem it also appeared that this is not how they are implementing dither waves.

The final implementation is a sinusoidal wave added to each control signal which is saturated to zero for negative values. The Simulink implementation consists of a sine wave block configured with the dither amplitude and frequency. As the control signal is a floating value between 0 and 1 representing 0 % to 100 %, the dither amplitude is set at 0.05 which should correspond to a current around 50 mA.

This implementation is not dependent of the current controllers since it is applied directly to the control signal. By doing so it was possible to keep the dither waves without affecting the current regulators. In fig. 19 two graphs are displayed, one for spool current and one for solenoid displacement. Both of them include plots with the new current controller outlined in the previous section (section 3.2.2) along with the dither current superimposed, and with the old current controller and no dither current. Note that the skips at approximately 0.75 s and 1.85 s are due to sensor errors. A clear difference in response time with the new and old current controller can also be seen, it is also possible to see the impact on the solenoid displacement.



(a) The current response to the step input.

(b) The solenoid displacement response to the step input.

Figure 19. The solenoid current and displacement response to a step input with 60 % reference input, using the new current controller (with dither current) and the old current controller (without dither current) respectively.

3.3 Setting up test cases

In order to determine accurate model parameters and verify the system requirements a series of pre-defined test cases have been created and conducted. The test cases, which are stated in Appendix C, are MATLAB-scripts which are loaded into Simulink by the operator. Each test case contains a set of inputs for the actuators, initial states, and test case duration. The inputs to the actuators are values representing current supplied to the solenoids. As seen in the appendix, there are several different test cases defined to aid in model verification for as many scenarios as possible. In order to prevent any accidents, such as hitting the ceiling, the runtime is limited. Thus, test can be performed in a safe manner.

In the MATLAB script the inputs are stored in a matrix. Each column represents inputs for any given discrete point in time. Depending on the test case runtime and the sampling time, Ts, the number of columns is defined. The rows of the matrix represent inputs to the three cylinders. The first, second and third row represent the first boom, second boom and extension respectively. An example of this MATLAB script can been seen in listing 1.

Listing 1. Example of code used for test case

```
%% Test Case 1A
%Test time -> testTime
%Sampling -> Ts (From ProjectH3_Init_ModelParams)

%1st Row -> 1st Boom
%2and Row -> 2and Boom
%3rd Row -> Extension

clear testCase

testTime = 3;
numElements = testTime/Ts;

testCase(1,:) = linspace(100,100,numElements);
testCase(2,:) = linspace(0,0,numElements);
testCase(3,:) = linspace(0,0,numElements);

%% Initial state
```

```
Boom1 = 200; %[mm]
Boom2 = 360; %[mm]
Extension = 10; %[mm]

initialStates = [Boom1 Boom2 Extension];

disp(' ')
disp('----------------------------------------------')
disp('Test Case 1A')
disp(' ')
disp('Initial states')
disp(['Boom 1: ', num2str(Boom1), ' [mm]'])
disp(['Boom 2: ', num2str(Boom2),' [mm]'])
disp(['Extension: ', num2str(Extension),' [mm]'])
disp(' ')
disp (['Test time: ', num2str(testTime), ' [s]'])
disp (['Sampling: ', num2str(Ts), ' [s]'])
disp('----------------------------------------------')
```

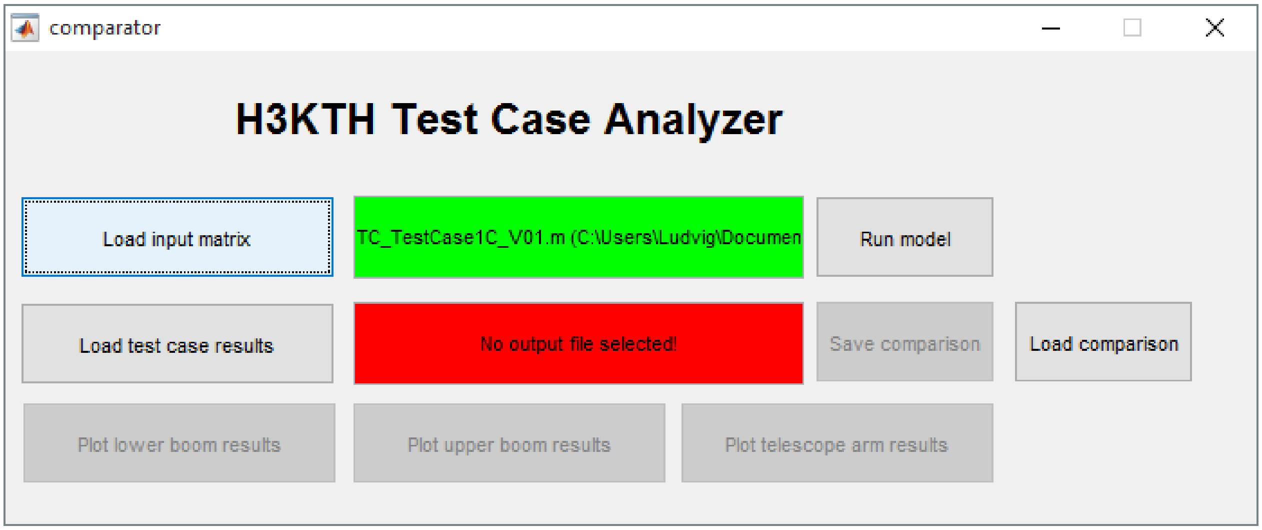
The initial states of each function are printed in the MATLAB command window, so the operator has to manually run the crane to the initial position before the test is executed. The procedure of executing the test cases is as follows:

1. Load the desired test into MATLAB
2. Compile the Simulink model and load the compiled file into dSPACE Workspace
3. Start the system in the initial position and execute the test by pressing the “Test Mode” button in the dSPACE ControlDesk user interface
4. Load the test results and execute the same test case on the model by using the Test Case Analysis software (see subsection 3.4)

When the test is stopped all the recorded data is stored as a struct which can be opened and manipulated in MATLAB. Since all relevant available values are logged and saved, the model can be compared with the real system for both the purpose of tuning parameters and verification. How this is performed is described in the next subsection.

3.4 Test case analysis software

In order to be able to quickly compare files and to save formatted objects with test result (measurement) data, and modeled data, a software interface was created. The main software interface is illustrated in fig. 20 below.



H3KTH Test Case Analyzer

TC_TestCase1C_V01.m (C:\Users\Ludvig\Documen

No output file selected!

Figure 20. Graphical user interface for the test case analyzer

This interface lets the user select and load the files used to generate the test case input. It can also load the measured test case result, as well as previous comparisons of modeled and measured results. Upon loading the data, it is possible for the user to run the model, plot the measurements, model results, and save a properly formatted version of all data for easy access later.

The GUI uses visual feedback to indicate whether an appropriate file has been loaded or not, using red and green colors, and also features error messaging and handling if exceptions are caught. The software is strictly bound to the way the recorder in the dSPACE interface outputs data. It is necessary that the order of the data is always the same in the resulting `.mat` file, since the logging software does not perform any checks whether the input data is in a specific order (only that there's enough fields).

However, should the recorder change the output format, it is easy to modify the code to display the correct data. Similarly, in order to choose which model to run, it is necessary to specify this within the code. In this case, it's simply a matter of changing a variable name. The analysis tool is (crane) model independent, as long as the model outputs its data in a correct manner.

Chapter 4 Modeling the system

This chapter describes the modeling work within this project. This includes a description of all models designed this year and also the improvements done to the model which was received from last year's group. As such this chapter discusses the limitations of the models and presents arguments and motivation for all changes and development work done. This is followed by a description of the implementation of the gradient descent algorithm for optimizing the model.

4.1 Model background and existing material

The design and verification of the system model is one of the main requirements from the stakeholder. With this model one should be able to simulate the dynamic behavior of the crane, and therefore use it for controller development. From the stakeholder requirements (see Appendix D), model requirements found in Appendix E have been derived which have been considered during the model development.

Since this project is a continuation of a similar project from last year, that project's model was inherited and used as a start in this project's modeling. That model was a regular Simulink model representing the hydraulic system, i.e. everything from the hydraulic pump to the actuators, and a Simscape model describing the mechanical components of the crane. This is referred to as the hybrid model. The stakeholders wish is to use that model as a base and try to verify it by comparing tests to the crane and tuning its parameters.

Besides the model acquired from last year's project, another one was also designed in order to get the possibility to compare these models to each other. The model is a pure Simscape model designed with the two tools SimMechanics and SimHydraulics. The reason for designing this model in SimHydraulics was to ensure that the model had all individual hydraulic components modeled properly. This would then give the possibility to verify the Simulink model's components to SimHydraulics' and make sure that no single component is modeled wrong and also that the models behave similarly.

4.2 Hybrid Simulink-Simscape model

This model is designed in two different environments of Simulink, i.e. pure Simulink modeling and Simscape modeling; hence the name "hybrid model". The hydraulic system is designed in Simulink and consists of equations describing the system while the mechanical model is completely designed in the Simscape environment SimMechanics. The model is organized in two main subsystems, the hydraulic systems which covers everything from input references to output force on the actuators, and the mechanical system which translates the force from the cylinders into crane motion. The main layout of the model can be seen in fig. 21.

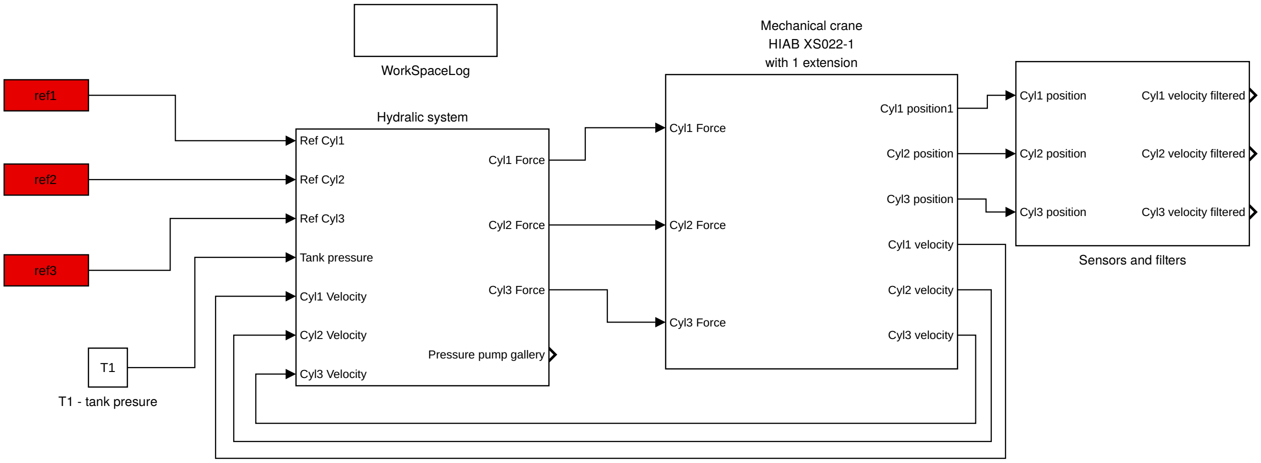


Figure 21. The main layout of the hybrid model

The hydraulic system is further divided into two other subsystems, the valve and the actuators. In each actuator subsystem, one cylinder is modeled and in the valve subsystem one can find the valve and also the counterbalance valves. These systems are shown in fig. 22.

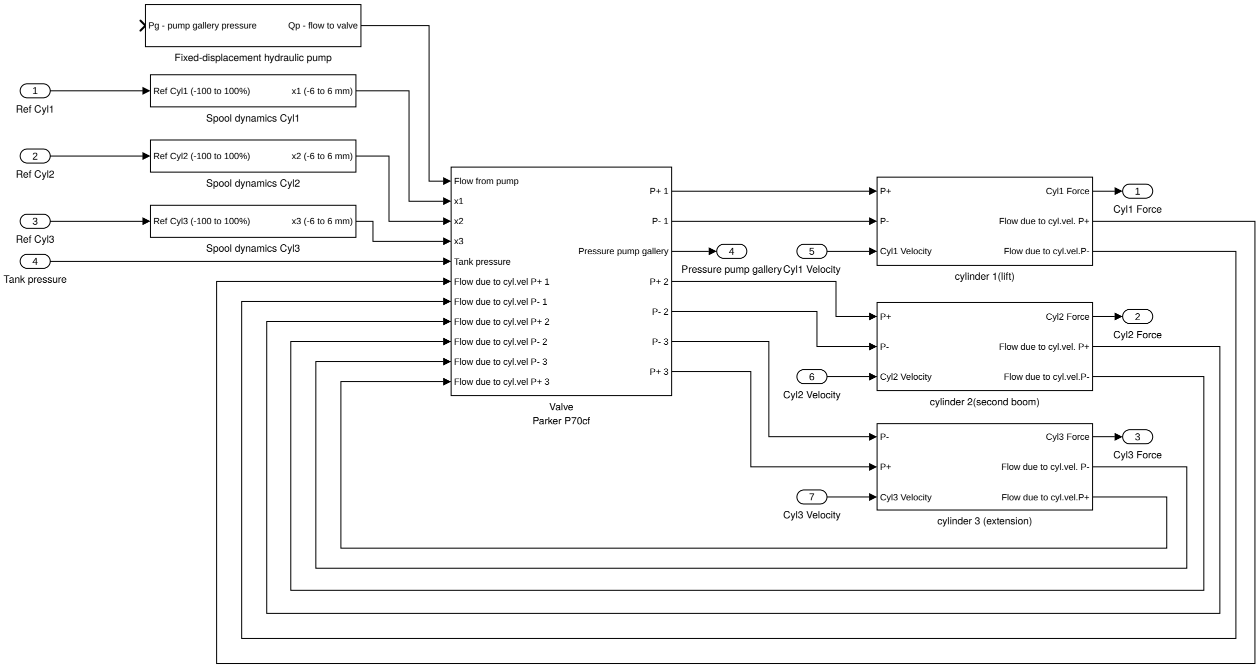


Figure 22. The hydraulic subsystem of the hybrid model

Going further into the left block in fig. 22 one can see the model of the valve and also the counterbalance valves. This subsystem is displayed in fig. 23. The gallery pressure is here modeled with four orifices which are displayed in the left hand side of the picture. The pressure is then acting at the work ports of the cylinders which can bee seen on the right hand side. The counterbalance valves themselves, which are modeled together with the workports on the right hand side, are approximated by using a polynomial, i.e. attempts have been made to mimic the area curves of the valve sections and apply the same principle to the counterbalance valves. By using pressure as input, the flow is determined using these polynomials. This may not be ideal given that there are several other factors that may affect the behavior.

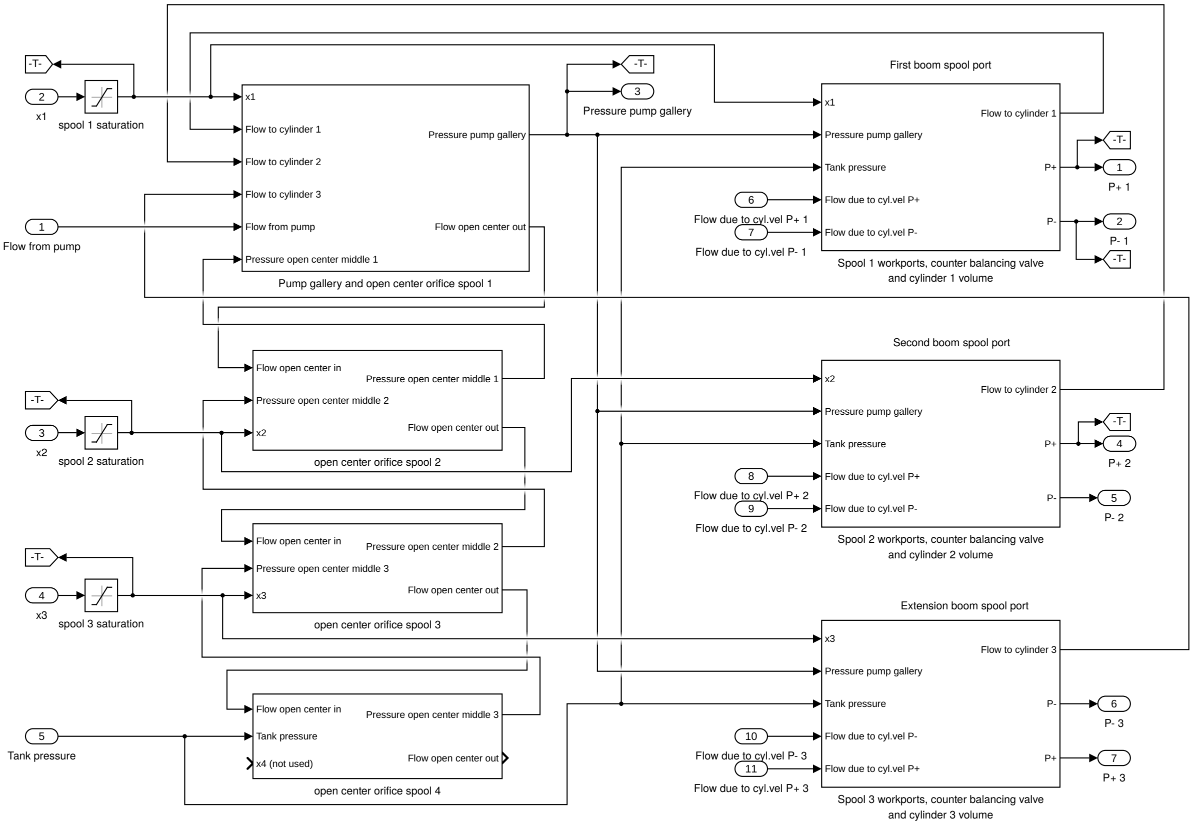


Figure 23. The subsystems corresponding to the various valve systems

4.2.1 Improvements to the hybrid model

Before starting the work with tuning the hybrid model, it was first reviewed for obvious errors and other faults that could be the reason for unexpected or erroneous behavior. By looking at drawings received from Hiab and reading up on hydraulics, some errors were found and corrected. These errors and the adjustments done to the model are presented in the list below.

- The slew function was originally modeled as if it was connected to the last section of the valve, which it not the case. The slew is connected to the first section and the model was corrected according to this. The impact of this error is considered negligible since the slew function is neglected during this project.
- The spool area data retrieved from the valve manufacturer was not used correctly. Each spool has its own relationship between spool displacement and orifice area. To get specific behavior for each function the correct data has to be used at the right place and the model used the wrong data for the extension boom. This has been corrected by changing the look-up tables for the extension boom.
- The counterbalance valve at the second boom was modeled as a single acting counterbalance valve. When looking at the drawings received from Hiab (see Appendix F) one can realize that there is no single acting valve mounted there. In reality, it is a double acting valve which is of the same type as the one mounted on the extension. By the time these drawings were received there was however no time to adjust this in the model.
- The mechanical model was unorganized and hard to follow, mainly due to the process of automatic

generation of the model. However, the model appears to behave correctly, no erroneous behavior could be found. The model is now reorganized with a better layout but no logic changes have been made. It is also possible to expand the mechanical model with a more advanced friction model which is further described later in the report.

When these three errors were fixed, the model had to be modified in order to be compatible with the Test case analyzer. The signal builder which created references for the spools were replaced with blocks that reads from an array in the MATLAB workspace. These arrays are created with the test case scripts.

4.2.2 Limitations to the hybrid model

There are some limitations to the model. These are mainly either implementation-specific limitations, or system-specific limitations. Regarding the implementation, the structure of the model itself is rather complex and to find the various subsystems that compose the crane within the model can be time consuming and tricky. Finding which parameters correspond to what behavior is difficult, especially when one part of the system may affect another part somewhere else due to it's MIMO characteristics.

Additionally, the spool dynamics, i.e. the dynamics from a reference input to solenoid displacement, is approximated with a transfer function. These transfer functions are black box modeled from data collected when the crane had last year's setup. Some parameters, like ones belonging to the current controllers, have been changed since and the transfer functions should be reviewed to better correspond to the crane setup.

Finally, the counterbalance valves are modeled so the oil can flow through a check valve in one direction, and in the reverse direction it must pass an orifice. The orifice opening is modeled to be dependent of the pressure at the load side of the valve and increases along with pressure. One limitation is that this displacement has also been shown during discussions with Hiab to be dependent of the oil flow and pressure in both chambers. The area curves for the orifice opening in the valve also turned out very difficult to get and these two facts makes the counterbalance valves a huge limitation in the model. Not only the behavior but also the physical data are approximations adopted to fit an expected outcome.

4.3 Simscape-based model

The alternative to the hybrid model described in the previous chapter is the Simscape model which is completely made in the two environments SimHydraulics and SimMechanics. The mechanical model is the same as used in the hybrid model, so the difference between them is the hydraulic model. An overview of the model can be seen in fig. 24, where the three bigger blocks represent the mechanical model and the three with pictures of hydraulic actuators is the hydraulic.

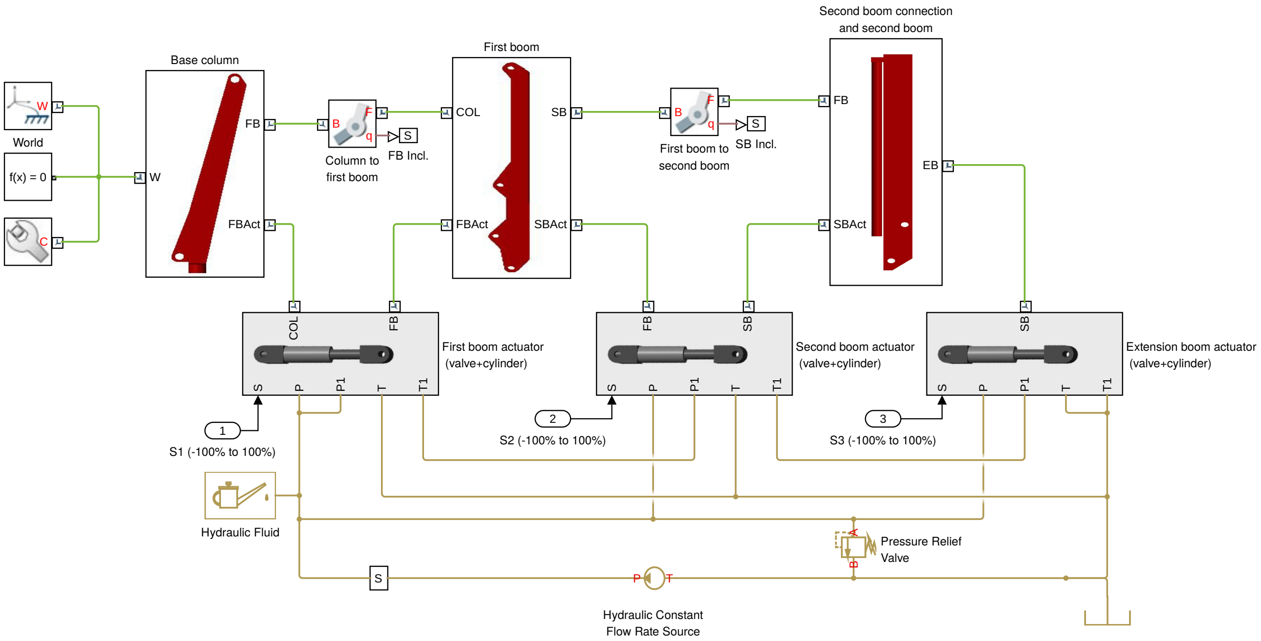


Figure 24. The main layout of the Simscape (SimHydraulics and SimMechanics) model.

Each of the three blocks that represent the hydraulic model contains a section of the valve and the components connected to it, which is the cylinders and possibly also a counter balance valve. A more detailed picture of this is presented in fig. 25 which is presented below.

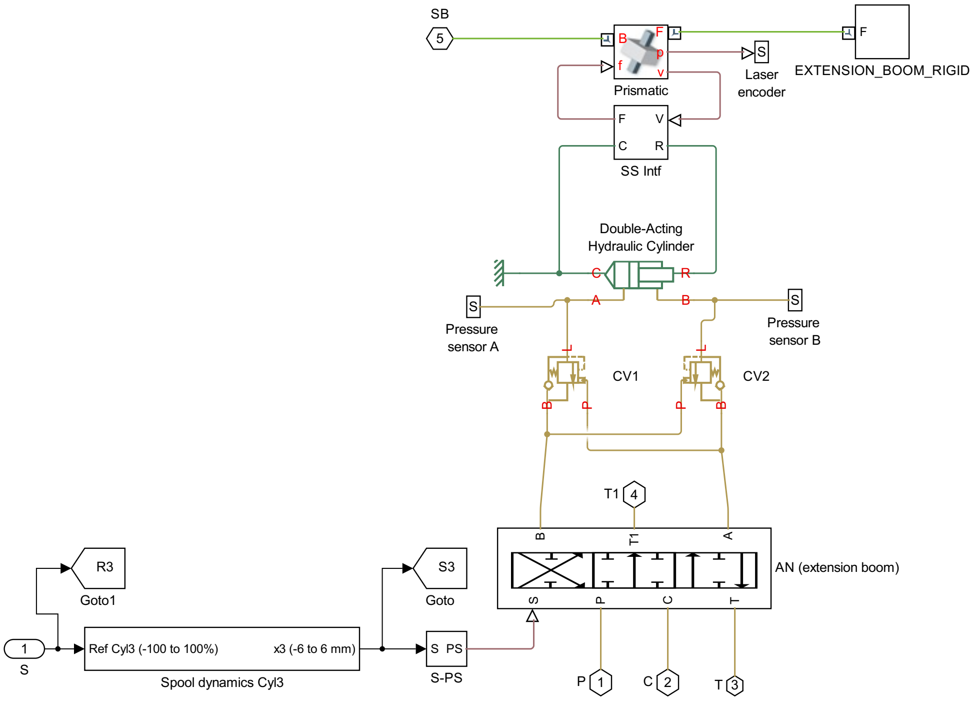


Figure 25. The layout of the hydraulic components and connection to mechanical system.

The valve section in fig. 25 is representing the extension boom, with a cylinder and a double acting counterbalance valve. Since there are no double acting counterbalance valves in the SimHydraulic library, these has to be modeled as two single acting valves connected to each other. This is done by connecting the work port A to one of the valves pilot pressure ports and the B work port to the other. These two valves have the caption CV1 and CV2 but on the crane this is, as mentioned, only one valve.

The spool dynamics is modeled with the same transfer function that's used in the hybrid model. This is still a limitation in the model and a better solution could be modeling these.

Since there aren't any library components in SimHydraulics to model open center valves, these have to be modeled manually. In this model, each valve section of the open center valve is modeled as a subsystem which can be seen in the bottom of fig. 25 as a subsystem with a directional valve symbol. The various inputs and outputs into this valves are

P: The current gallery pressure/pump pressure that has been built up by the system.

C: The input port into the open center. For the first section in a valve, this is the same as P. For all others, this should instead be connected to the T1 port of the valve before it.

T1: The output from the open center for the valve. This should for the last valve be connected to tank and for all other valves be connected to next valves' C port.

T: Return line to tank.

A: Valve A side outlet.

B: Valve B side outlet.

In fig. 24 one can see how the ports should be connected for a open center valve with three sections. A detailed view of how the internal dynamics of each section is modeled is presented below in fig. 26.

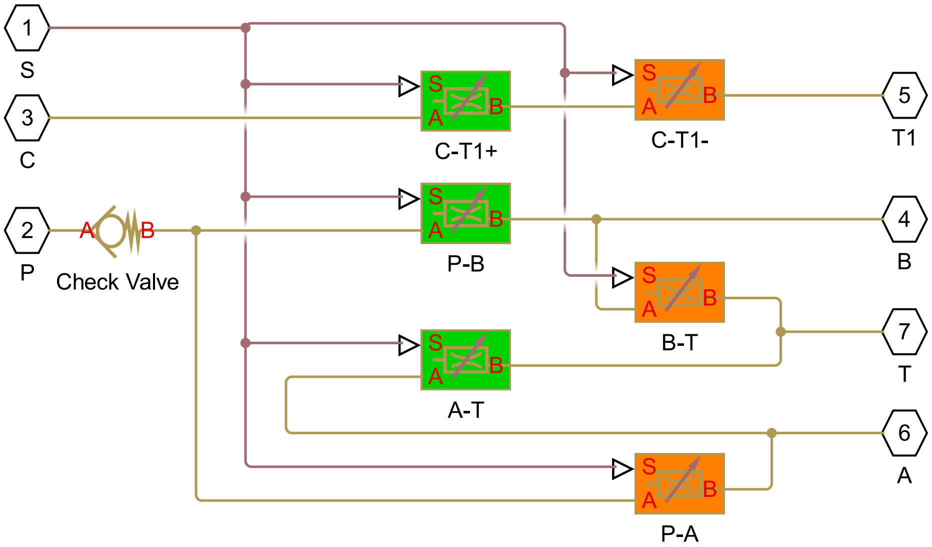


Figure 26. An overview picture showing how each section of the valve is modeled.

All blocks in this subsystem are representing one of the orifices inside the valve. These orifices are all modeled with opening area tables received from the manufacturer. The green blocks are activated when oil is flowing in P-B and A-T direction, meaning the pump is acting on the B side of the cylinder and the A side returns to tank. The red blocks are used in the other direction, when the pump acts on the A side and B returns to tank. The two upper blocks contain data about the orifice that lets oil flow through the open center, one for positive displacement of the spool and one for negative.

4.4 Additional mechanical modules for the models

As an addition to the mechanical model, three additional modules were designed in order to capture the dynamics of th the crane even better. These were a mechanical model designed with simulink blocks, an improved friction model and a load model. All these three models are described in the following section.

4.4.1 Equation-based mechanical model

Due to slow simulation of the SimMechanics model, a model of the system mechanics was developed manually from the equation of dynamics. Another reason was to derive the equations in order to perform model based control. Free body diagrams, equations, assumptions and model architecture are presented here.

The crane is modeled as two interacting mechanical systems, the first boom and the second boom including the extension. Hence, two systems of equilibrium are found, one for each boom. The assumptions and constraints are:

- The extension is integrated in the system for the second boom
- The inputs to the system are the forces from the cylinders
- The hydraulic cylinders are not included in the geometry
- The system consist of stiff bodies
- Lumped parameters are used

The free body diagram of the second boom is depicted in fig. 27 below.

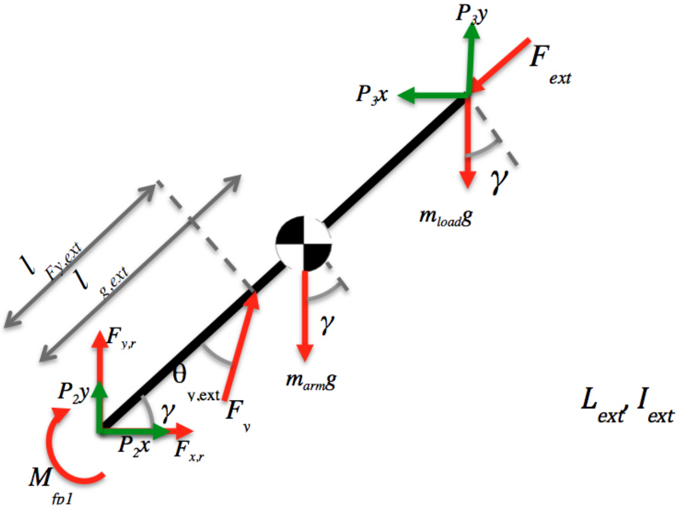


Figure 27. Free body diagram of the second boom

P_{2y}, P_{2x}, P_{3x}, and P_{3y} here denote boom endpoint positions in the cartesian coordinate system. The second boom and extension is lumped into one whole point mass, m_{arm}. Inertia and length of the whole boom are denoted L_{ext} and I_{ext}. Force from the cylinder applied on the extension is denoted F_{ext}. The force from cylinder two is denoted F_y and has $\theta_{y,ext}$ as the angle of attack relative to the centerline of the boom. The forces from the bearing that interconnects the first boom with the second boom are denoted $F_{x,r}$, $F_{y,r}$ and M_{fp1}. The pitch of the boom is γ.

Through this, a system of equations for the motion of the boom can be drived as

$$
\begin{aligned}
\circlearrowright: \quad & M_{fp1} + F_y \sin(\theta_{y,ext}) l_{Fy,ext} - m_{arm} g \cos(\gamma) l_{g,ext} - m_{load} \cos(\gamma) + I_{ext}\ddot{\gamma} + \\
& + m_{load}\ddot{P}_{3x}\sin(\gamma) L_{ext} - \ddot{P}_{3y} m_{ext,tot} L_{g,ext}\cos(\gamma) = 0 \\
\rightarrow: \quad & F_y \cos(\gamma + \theta_{y,ext}) + m_{arm}(-P_{1x} + \ddot{\gamma} l_{g,ext}\sin(\gamma) + l_{g,ext}\dot{\gamma}^2\cos(\gamma)) = 0 \\
\uparrow: \quad & F_y \sin(\gamma + \theta_{y,ext}) + (m_{arm} + m_{load})(-\ddot{P}_{1,y} - \ddot{\gamma} l_{g,ext}\cos(\gamma) + l_{ext,g}\dot{\gamma}^2\sin(\gamma)) + \\
& + F_{y,r} - sin(\gamma) F_{ext} = 0
\end{aligned}
\tag{68}
$$

The equations for the motion of the first boom were carried out as following. A free body diagram of the first boom is shown in fig. 28 below.

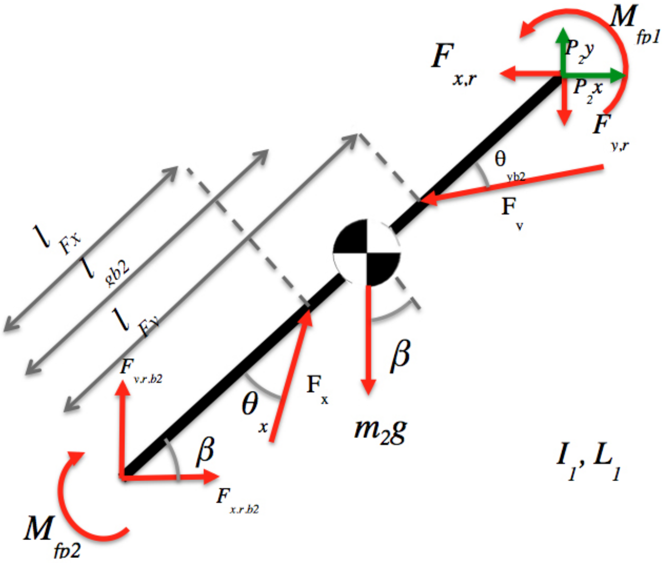


Figure 28. Free body diagram of the first boom

Here, the magnitudes $F_{x,r}$, $F_{y,r}$ and M_{fp1} introduced in the equations of the second boom are applied on the upper part of the first boom. The length and inertia of the boom are denoted L_{ext} and I_{ext}. β is the pitch of the boom. F_y and F_x denotes the force from cylinder two and one. The angles of attack are θ_{yb2} and θ_x.

An equation system for the motion is derived as follows

$$\begin{aligned}\circlearrowright:\quad & \ddot{\beta}I_2 + M_{fp2} - M_{fp1} + m_2\cos(\beta)L_{gb2} - M_{fp2} + \ddot{P}_{2y}m_{b2}L_{gb2}\cos(\beta) - P_{2x}m_{b2}l_{gb2}\sin(\beta)+ \\ & +cos(\beta)F_y l_{Fy} - F_x\sin(\theta_x)l_{Fx} - sin(\beta)F_{x,r}L_1 + \cos(\beta)L_1F_{x,r} = 0 \\ \rightarrow:\quad & F_{xrb2} + F_x\cos(\theta_x+\beta) + m_{b2}(\ddot{\beta}l_{gb2}\sin(\beta) + l_{gb2}\dot{\beta}^2\cos(\beta)) - F_{x,r,ext} - F_y\cos(\beta-\theta_{yb2}) = 0 \qquad (69) \\ \uparrow:\quad & -F_y\sin(\beta-\theta_{yb2}) + F_{y,r,ext} - m_{b2}g + F_{yrb2} + F_x\sin(\theta_x+\beta)+ \\ & +m_{b2}(-\ddot{\beta}l_{gb2}\cos(\beta) + l_{g,b2}\dot{\beta}^2\sin(\beta)) = 0\end{aligned}$$

The two equation systems are transformed into a block system diagram and implemented in Simulink. Here, the inputs are the forces from the three cylinders. The outputs are β and γ including their first and second derivatives. These subsystems and their connections are illustrated schematically in fig. 29 below. The "Equation"-blocks correspond to the equations from the system of equations described above.

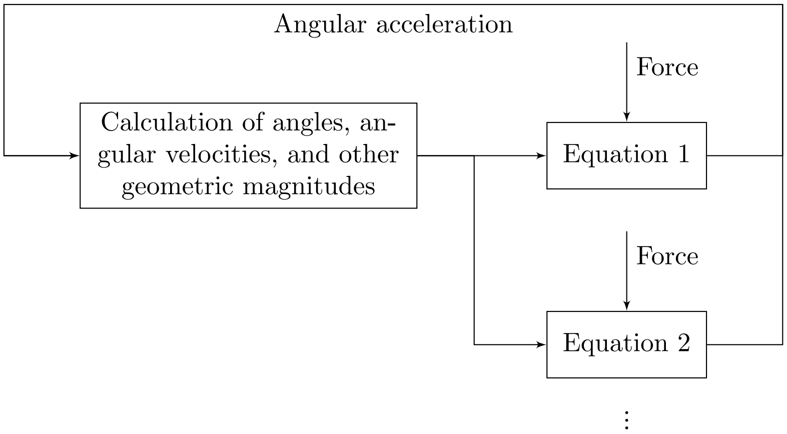

Angular acceleration
Force
Calculation of angles, angular velocities, and other geometric magnitudes
Equation 1
Force
Equation 2
⋮

Figure 29. Schematic overview of the system of equations

The angular acceleration and other magnitudes, such as reaction forces are derived from the equations. Therefrom, the angular magnitudes are fed back and geometrical magnitudes are calculated. Apart from pure mathematical operations, the system also contains logic operators and constrains in order to cope with different states, such as maximum cylinder elongation.

The whole system has been implemented in Simulink and is very complex. When executing the model, it crashes since it contains a number of algebraic loops. Several attempt have been made to resolve this problem with no success. Since the model is so vast, it is judged as too time consuming to verify and debug. Since a functional SimMechanics model exist, the need for another model is not crucial. Therefore the development of this model has been put to a halt.

4.4.2 Friction model

In order to get a more accurate mechanical model, a static friction was created and implemented as a Simulink sub-model on the SimMechanics model. In the friction model that was suggested by Hiab, the friction torque, M_f, in the joints are calculated as

$$M_f = F_r\mu_j D_j/2 \qquad (70)$$

where μ_j is the friction constant for the joints given by Hiab, D_j is the diameter of the joint and F_r is the resulting reaction force in the joint. This was implemented as a Coulomb friction, that is

$$M_f = \sqrt{(f_x^2 + f_y^2 + f_z^2)}\mu_j D_j/2 \tag{71}$$

when one of the following conditions is fulfilled

$$|\dot{\gamma}| > \dot{\gamma}_t \tag{72}$$

or

$$|M_f| < |M_{res}| \tag{73}$$

where the parameters f_x, f_y and f_z are the composite forces measured in the joint and $\dot{\gamma}$ is the measured angular velocity in the joint and $\dot{\gamma}_t$ is the threshold value for the angular velocity set close to zero. The resulting torque, M_f, is calculated from the measured angular acceleration, α_j and the inertia I_j, however, excluding the previously calculated friction, M_{fp}, i.e.

$$M_{res} = \alpha_j I_j - M_{fp}. \tag{74}$$

For all other cases, the friction torque is simply

$$M_f = M_{res} \tag{75}$$

The resulting joint friction model can be seen in fig. 30.

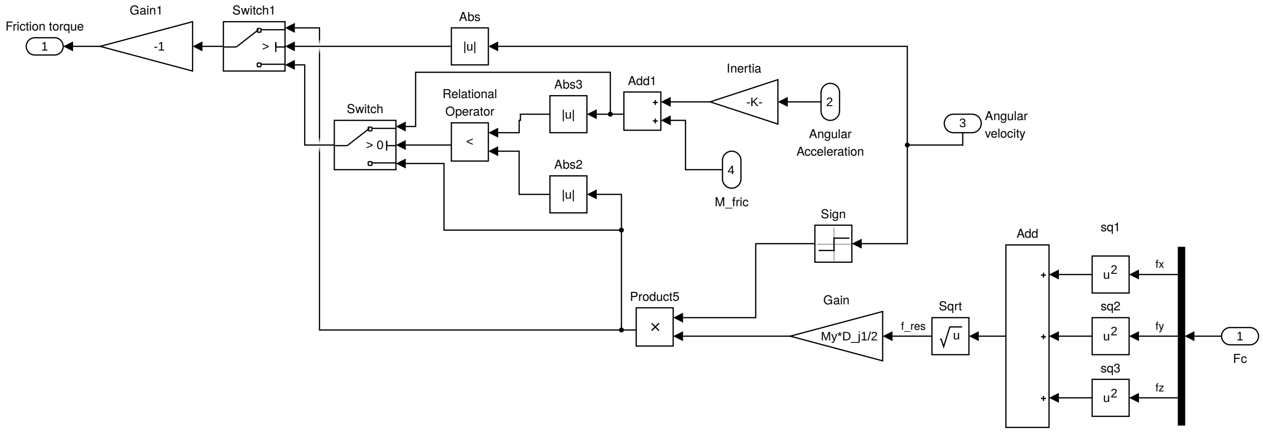


Figure 30. The Simulink module defining the friction

As the extension boom moves in a linear manner, the friction model used in this case was different. The friction on the extension boom was based on the reaction forces on the contact points between the second boom and the extension. These forces are highly dependent on the angle of the boom as well as the position of the extension. This model was, similarly to the joint friction model, developed from the coulomb friction concept. The friction force, $F_{f,\max}$ in the extension was described as

$$F_{f,\max} = \mu_{ext}(f_{r1} + f_{r2}) \tag{76}$$

where the reaction forces f_{r1} and f_{r2} are derived from equilibrium as

$$f_{r1} = \frac{g\cos(\theta)}{z}(m_{ext}l_c + m_l l_{ext}) \tag{77}$$

$$f_{r2} = f_{r1} - g(m_{ext} + m_l)\cos(\theta) \tag{78}$$

where m_{ext} is the mass of the extension, m_l is the mass of the load and l_{ext} and l_c are the distances from the base to the point of attack of the load and the center of gravity, respectively. The length z is the portion of the extension which is inside the second boom as shown in fig. 31. For the purpose of this particular calculation, this value will be limited to

where L_t is a threshold.

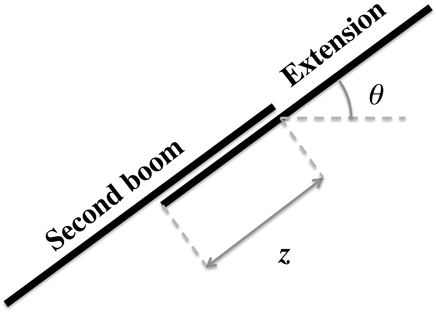


Figure 31. Simplified model of the upper arm and telescope

For the purpose this particular calculation, this value will be limited to

$$L_t < z < l_{ext} \tag{79}$$

Much like the joint friction, the resulting friction force, F_f on the extension depended on the velocity and resulting force such that

$$F_f = F_{f,\max} \tag{80}$$

when one of the following conditions are fulfilled,

$$|\dot{z}| > \dot{z}_t \tag{81}$$

$$|F_f| < |F_{res}| \tag{82}$$

where the parameters $\dot{z}$ and $\dot{z}_t$ are the velocity of the extension boom relative to the second boom as well as the threshold velocity, which is close to zero.

4.4.3 Load model

Further development of the SimMechanics model included a load model. When a load is applied onto the crane the interference problem is amplified and thus easier to observe. The physical characteristics of a physical load are comparable to a pendulum. Initial attempts of modeling the load as a sphere with mass connected to the crane with a thin rod in between, were unsuccessful due to simulations taking too long time (serveral hours).

Instead, a simplified model was therefore developed. An inertia block from the library was attached to the crane tip, which resulted in success. Simulation performance is only marginally affected. However, oscillatory behaviour due to load rocking are lost. During crane operation rocking of the load is undesirable and considered unsafe, it causes stress on the crane and the attachment of the load. Therefore crane operators try to avoid rocking of the load during operation, which is also avoided while running the test cases. The lost characteristics of load rocking are considered to not affect the project.

4.5 Model optimization

As an attempt to systematically tune the parameters, a gradient descent algorithm as seen in 2.6.1, the was implemented in MATLAB. The algorithm used a list of parameters to be tuned and the output to which it should fit. As the model had a very large set of parameters and outputs that should match the measured data, this proved to be a computationally difficult task. In order to improve the processing speed, only a few of the set of parameters were selected for optimization. The algorithm calculated a numerical derivative of the quadratic performance surface (MSE) for the N parameters used by running the model N times with a small step h added to each parameter, one at the time, while all other parameters remained constant.

Due to the time it takes for the model to compute an output, early experiments were performed on the discharge coefficients Cd_oc and Cd_wp that were used to estimate all the discharge coefficients in the model. These two values were initially used in the blocks representing all discharge coefficients, adding up to a total of 20. This alone would constitute a somewhat hard computation to make, especially considering that the nonlinear nature of the model would surely result in a rough quadratic performance surface and therefore, there would be plenty of local minima that the algorithm would get stuck on. The first experiment was to tune the two parameters which from which all discharge coefficients were derived from in the untuned model. This was set to minimize the error of the first boom displacement on a single experiment. This was done to make sure that the algorithm worked as it should and could be applied to the model. The test resulted in a set of values for the parameters Cd_oc and Cd_wp that would fit the model output of the first boom displacement almost perfectly to the measured output of the crane. The results are shown in figure 32. The remaining simulated outputs where, however, worse than before. This lead to the necessity to add more outputs to the weight function, f, creating the quadratic performance surface. This was defined as

$$f = \sum_{j=1}^{m} \sum_{i=1}^{n} (\hat{Y}_{i,j} - Y_{i,j})^2 \tag{83}$$

where n represents the number of data points in the measured and modeled data and m represents the number of outputs that are compared. In order to have more freedom to optimize, all blocks representing a discharge coefficient where changed to an individual parameter that could be tuned independently. This added up to 20 dimensions. The initial test was conducted with parameter values randomly selected in the range [0,1] for the discharge coefficient. The parameter set would also use the cylinder friction parameters in order to increase the freedom of design.

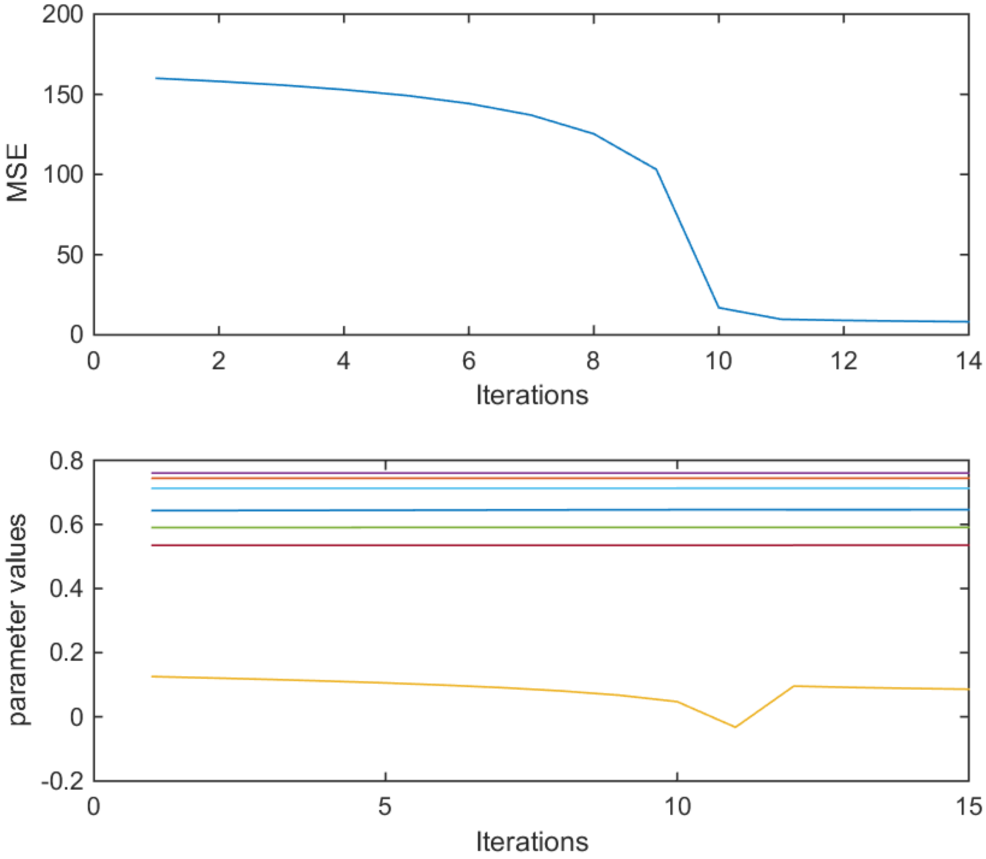


Figure 32. Plot representing the cost function decrease when tuning two discharge coefficients to fit the lower boom position. The lower plot shows the values of different parameters over time.

In an attempt to find the global minimum on the quadratic performance surface, it was necessary to try with many different initial guesses for the tunable parameters as well as different values for the step size α. Due to the high complexity of the model it was hard to achieve values that were satisfactory.

The code used the TCA interface to get all the needed data, i.e. the test case input and the measured results, into the MATLAB workspace. It then remaps the measured output to match that of the model in order to calculate the MSE values efficiently. Other inputs to the code are the parameters to be tuned as well as the outputs values that will be optimized. The gradient descent algorithm also required a step size α value that was acquired trough trial and error until a stable value was found. Another value that was necessary for the algorithm to run was the value h used for numeric differentiation which was also derived through testing. The algorithm can be broken down to a few steps:

1. Compute f_0=$f(\beta)$ for $M(\vec{\beta})$
2. For each β in $\vec{\beta}$: run the model with $\beta = \beta$+h and compute f_h
3. Calculate all partial derivatives $\frac{\delta f}{\delta \beta_i}$ derivatives as (f_h-f_0)/h
4. Update parameter values as β_i=β_i-$\frac{\delta f}{\delta \beta_i}$
5. Repeat for a given number of iterations

Where $M(\vec{\beta})$ is the model run with a set of variables beta and $f(\vec{\beta})$ it's the MSE value for the set of parameters

A safety clause was later implemented in the model in order to avoid unstable steps. This will occur when a step assumed to be in a downward direction on the quadratic surface actually is ill-considered due to the surface being too rough. This was done by lowering the step size and deviating the step if the cost function value of a given iteration was higher than the previous one.

A weight vector was also added to the algorithm in order to give the user a chance to put extra emphasis on the improvement of a certain output of the model. The weight function simply multiplied a scalar value w_j to the difference in the output of the model and the measurements for an output j. This meant that the cost function f was calculated as

$$f = \sum_{j=1}^{m} \sum_{i=1}^{n} ((\hat{Y}_{i,j} - Y_{i,j}) w_j)^2 \tag{84}$$

Further attempts were made to try to equalize the outputs of the model. As the cost function does not take into account the order of magnitude of the outputs, only the absolute difference of these, the algorithm tended to improve only the outputs that had a large expected values. These outputs were in many cases completely uncorrelated and had different units of measurement in some cases. In order to avoid this source of error, a normalized version of the MSE was calculated instead as

$$f = \sum_{j=1}^{m} \sum_{i=1}^{n} ((\hat{Y}_{i,j} - Y_{i,j}) / Y_{i,j})^2 \tag{85}$$

In a later attempt to stabilize the parameter growth, an alternative updating system was implemented. This was done by updating the parameter values with a fixed step instead of a step that is proportional to the derivative. This could be described as

$$\beta_i = \beta_i - sign(\frac{\delta f}{\delta \beta_i}) \tag{86}$$

Chapter 5 Control design and reduction of the interference

This chapter describes the controller structures, which have been designed and evaluated on the system model, as well as the system implementation. A description about the verification and validation of the implemented controllers performance is also included.

5.1 Control strategies

The main objective of the velocity controller is, to reduce the interference between functions when two or more are running at the same time, and make it possible for all cylinders actuate simultaneously. An example of this interference could be seen in fig. 33 where all booms are given a sinusoidal wave as input. The first and second boom are not supplied with enough oil to move when they're trying to move in negative direction. Only the extension is able to move in both directions since it is easier for the oil to run through the extension compared to the other cylinders. In order to make all cylinders actuate simultaneously, a controller which takes care of this interference needed to be developed.

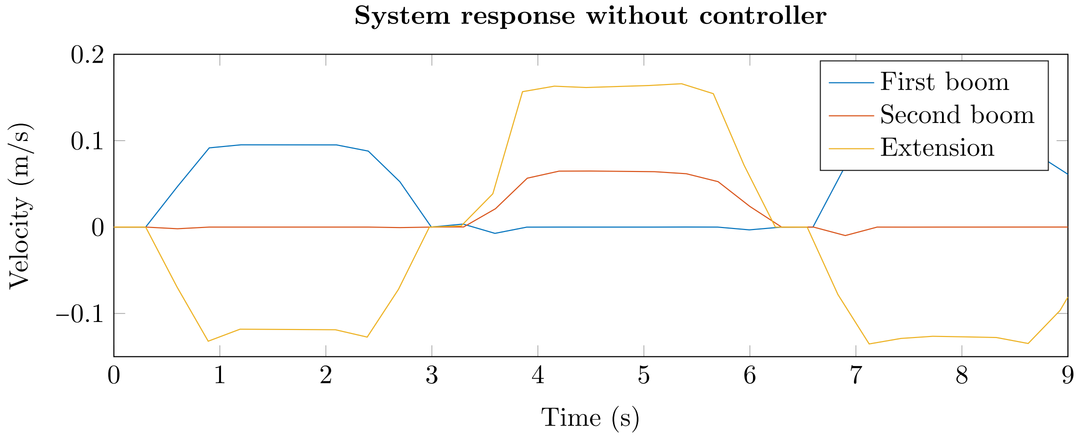


Figure 33. System response with sinusoidal input to all functions

The main idea with this controller is to regulate the open center valve so the oil is shared amongst the different functions, proportional to the user input. Different controller strategies are investigated in chapter 2, and some of those alternatives were considered as feasible solutions on the crane. Experts at KTH and Hiab suggested that it first should be investigated if a linear controller is sufficient to solve the problem. This is because of the relative low complexity and prior knowledge of the area within the project group. Therefore different linear control strategies were first evaluated before advancing to more advanced control theories.

5.1.1 SISO PID structure

An initial strategy is to treat the system as a decentralized control system described in section 2.2.3 and reduce the complexity of the multiple input/multiple output (MIMO) system. This is done by instead of using a MIMO controller, try to use a set of independent controllers with single inputs/outputs as seen in fig. 34.

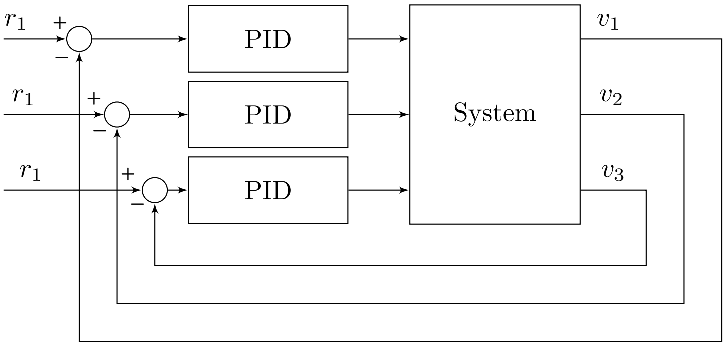


Figure 34. System with decentralized PID controllers

This is a very naive approach, since reducing the interference disturbance between functions is the main objective of the controller, and this interference affects the controllers. The specific problem with this strategy is that even if one of the function is given an input at maximum, the flow to that function could still be reduced if the other functions are running. Since the input of that function already is at maximum, the only strategy to increase its flow is to reduce the other functions' inputs. This can not be achieved with a SISO PID structure since they cannot affect each other's input, only their own.

Another challenge is that the maximum velocity is a function depending on the load, the position of the crane and the number of functions active. Since the complex system varies this maximum velocity, it makes it difficult to know the current maximum velocity when calculating the reference. This could easily make the controller unstable in cases where the velocity reference could not be obtained due to interferences, making the integration part of the PIDs winding up.

5.1.2 Dynamic threshold

In order to make a viable control strategy with a SISO PID structure, despite the challenges described above, a logic function referred as the "velocity planner" was proposed. The velocity planner alters the user's reference before sending the signals to the controllers. This is to ensure the controllers only get references that are in the limit of what is feasible as seen in fig. 35.

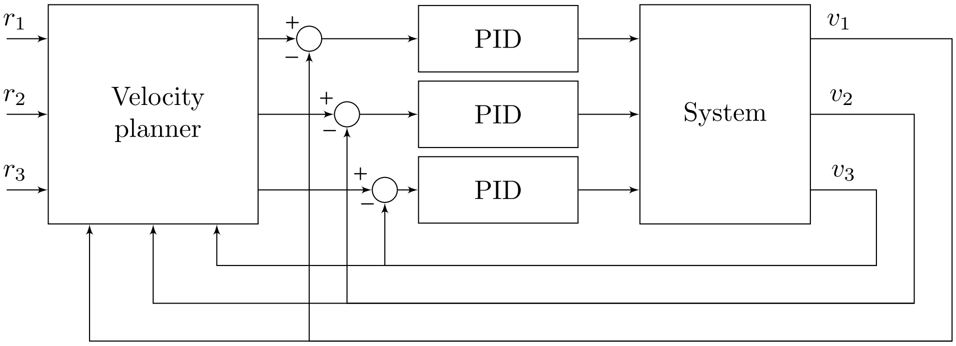


Figure 35. SISO PID structure with a velocity planner

One strategy was to implement the velocity planner with dynamic thresholds. The thresholds should correspond to the highest possible velocity each function could maintain without causing another function to stop. The threshold is dynamically changed since different positions of the crane, number of active functions, and different loads could affect the threshold velocity. This is altered depending on whether the reference velocity is obtained by all functions or not. The measured velocities are compared with the current value of the threshold and the references are scaled with a tunable parameter β. If all functions have a higher velocity than the threshold, the threshold is increased. If one or two functions have a lower velocity than the threshold, the other functions' references are decreased with a parameter α. The algorithm could be described as:

1. If a function's velocity is inside a dead band around the threshold velocity, the reference is unaltered.
2. If all three velocities are greater than the threshold, the value of the threshold is increased by a factor α.
3. If all three velocities are lesser than the threshold, the value of the threshold is decreased by a factor α.
4. If only one or two velocities are greater than the threshold, the reference to those functions are reduced by a factor β.
5. If the value of some velocity is lesser than the threshold, the reference to that function is increased by a factor β.

The simple idea is to reduce interference by increasing the reference to the functions which can not reach a certain threshold, and to reduce the input to the functions which are above the threshold. A drawback with this method is that the sampling time of the velocity planner greatly affects the algorithm. Slow sampling time gives a slow response since the reference is scaled with a constant factor β each sample. Faster sampling time will however make the system unstable since the references are scaled much faster than the response of the dynamical system.

5.1.3 Flow calculated feed-forward

Since the main issue is depletion of flow to the different functions, another approach was to implement a feed-forward function. It maps user input to a certain flow and then transforms it to a control signal as seen in fig. 36.

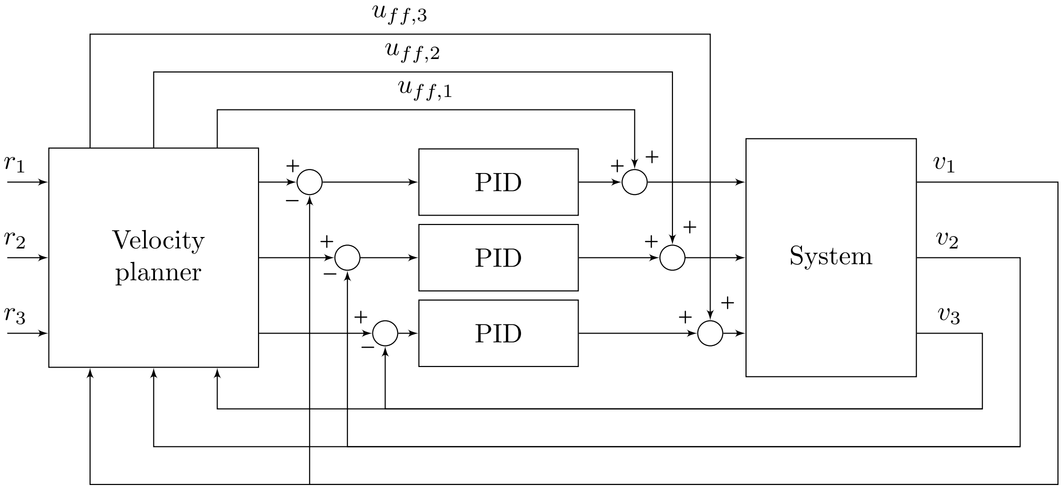


Figure 36. SISO PID structure with feed-forward

This method in essence requires multiple data maps containing mappings of references to flow for the individual functions. These could be collected from the actual system by measuring cylinder velocities, since flow and velocity has a linear relationship. When a user sets a reference, the total amount of flow required would be determined by the data maps. If the sum of flows is greater than what the system can get from the pump (30 L/min), the user inputs are scaled down to get references equal to total available flow.

To get an exact conversion from user reference to flow for every position of the crane, with every possible load with every possible input, the number of recorded data maps would become large and time consuming to record. An algorithm to approximate the load carried would also be necessary, to know which data map to use, as well as to continuously measure the crane position. If system dynamics could be approximated at certain positions, as well as creating data maps only for loads that differ more than a fixed value, the amount of data maps could be significantly reduced, which a microcontroller storage memory could manage. A benefit is that no heavy calculations or extra sensors would be needed. The standard SISO PID structure could then focus on correcting small errors generated by the approximation of data maps while the interference would be corrected by the feed-forward function. This approach resembles a trajectory planner, where the input generates a calculated optimal output and the control loop only corrects differences between calculated optimal output and the true optimal output signal.

5.1.4 Minimizing error difference

A concern when using PID structures in this application is that the maximum velocity for each function is heavily dependent on the number of used functions, lifted weight and crane position. Another strategy proposed was the velocity planner whose aim is to minimize error differences for all active functions. Instead of error between reference and measured velocity as input, as in a traditional PID structure, the difference between error and average error for all active functions was used as input signal, as seen in fig. 37.

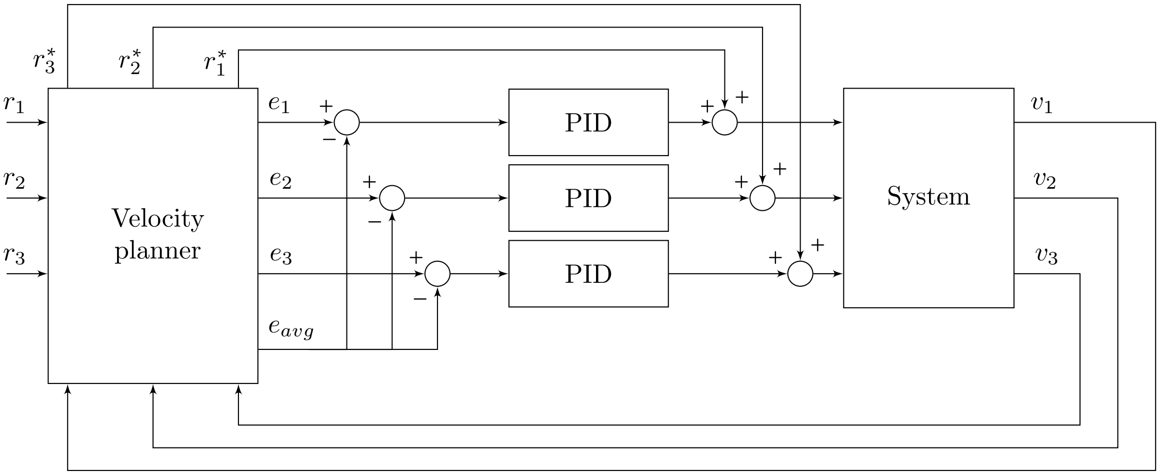


Figure 37. System with minimizing error difference structure

The proposed structure works by first measuring the maximum velocity, $v_{\max,n}$, for a single function active in the system with an open-loop test. This parameter is then used to calculate the relative obtained speed measured $v_{rel,i}$ as seen in eq. (87) when a function is active,

$$v_{rel,i} = 100\frac{v_i}{v_{\max,n}} \tag{87}$$

and the error and the average error for active function i is calculated as in eqs. (88) and (89) where n is the number of active functions.

$$e_i = r_i - v_{rel,i} \tag{88}$$

$$e_{avg} = \frac{1}{n}\sum_{i=1}^{n} e_i \tag{89}$$

When a single function is active, the average error and the error for that function is equal, meaning no control signal is sent to the PID controller. This is because the aim of the controller is to reduce interference when multiple functions are running, thus if only one function is active the controller is inactive. If several functions are running, the error for each function converges to the average error for all active functions. The functions have the same relative velocity difference in relation to their maximum velocity, scaled with their respective input signal. Since the pump cannot deliver the flow needed for all functions to move at their respective reference velocity, the PID scales down all the inputs in a manner so that no function has a bigger relative velocity than the other functions. The number of calculations does not differ much in comparison with traditional PID structures, therefore implementation in an embedded system is considered feasible.

5.2 Controller implementation

The different strategies were first tested on the unverified hybrid model, since the unverified model ought to show some dynamics of the true system. The traditional PID structure did not work, as expected. The PIDs just wound up as they tried to give more input to the system when their respective functions were not moving. The attempted implementation of other control structures are described below.

5.2.1 Dynamic threshold

The dynamic threshold function resulted in better response, but when implemented, the response was to a high degree dependant on the sampling rate. A higher sampling rate made the controller unstable but with sufficient tuning of the parameter β and the sampling rate, the dynamic threshold gave some positive results for some of the test cases. If the same parameter β is used for all functions, the algorithm will force the functions to move with the same velocity which may be undesirable. This could be resolved by using a individual β to individual functions to provide more flexibility. This controller approach was concluded not feasible since its response was unpredictable in different test cases. In some of the test cases, the dynamic threshold converged to zero instead of the highest velocity possible when running all three functions simultaneously.

5.2.2 Flow calculated feed-forward

The calculated feed-forward mapping structure was evaluated in the same manner as above. A collection of data maps was obtained by carrying out open-loop tests for each function, reflecting several different scenarios. The recorded data was then used to produce a control signal which was feed-forwarded into the system. The PIDs could then focus on minimizing disturbances from varying loads, varying positions or that the dynamic friction changes when reducing the velocity. This method was deemed unsuitable in an open center valve system since flow has the possibility of running through the open center channel. When the different inputs were scaled down by the feed-forward map, the open center channel opening became larger. Thus, flow was diverted into the open center channel instead of the function-channels. Therefore it could be concluded that no linear relationship can be obtained when scaling inputs according to a flow map and maximum amount of flow available. The PID structure had to focus on more than just minimizing small disturbances, making this strategy similar to a classical PID structure.

5.2.3 Minimizing error difference

Minimizing the error difference was the only approach that gave positive results in all the test cases used for evaluating the controller on the unverified model. By measuring individual velocity for all functions, their relative difference in speed could be obtained. As an example, the extension moves roughly three times as fast as the other functions when used separately and this relative speed difference is still maintained by the controller. As seen in fig. 33, when comparing the response for a sinusoidal input to all functions, without the controller, the first and second boom could not move in negative directions since the amount of flow is not sufficient. With the controller as seen in figure fig. 38, the output still resembles a sinusoidal input, but the velocities are scaled in order to give the first- and second boom enough flow to move in negative direction. The relative speed difference is still maintained, where the extension moves roughly three times faster than the first boom and the second boom.

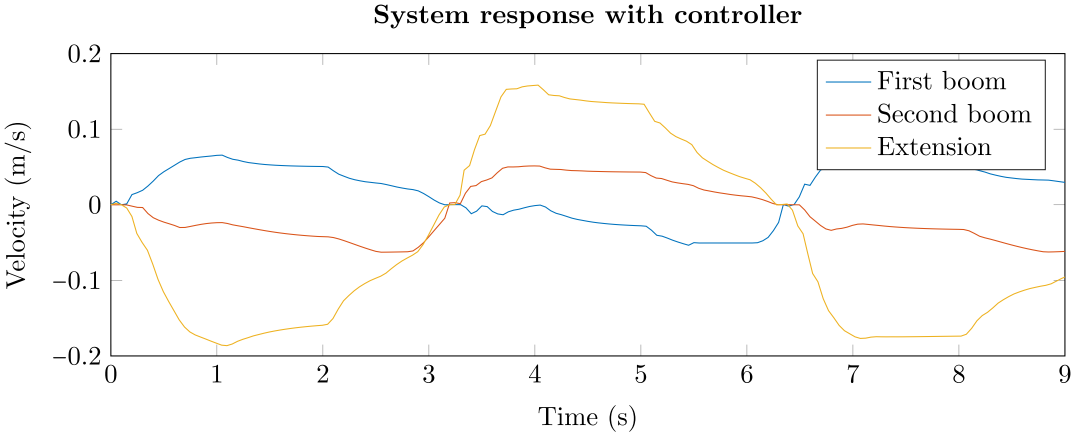


Figure 38. System response with sinusoidal input to all functions

Before the initial tests were made on the actual system, the controller structure was upgraded with some additional features to ensure a faster and safer response. Since the actual system has a large dead band θ, mainly because of non-linearities, the references were scaled to be within the limit where the system gives response. A user input r_i between -100% to 100% is scaled to

$$r_i^* = \frac{100 - \theta}{100} r_i + sgn(r_i)\theta \tag{90}$$

to give a faster response. Since this controller is implemented on a system where there is great risk of potential hazard if an unexpected control signal is produced, a safety layer was added between the controller structure and the actual input to the system. To reduce risk of hazardous events, the safety layer ensures that if the user gives a positive reference, no negative control signal could be produced and vice versa. This is to prevent scenarios where the crane moves unexpectedly in the wrong direction, a situation that may occur on the actual system if a controller produces an unexpected undershoot or becomes highly unstable. The complete controller structure can be seen in figure fig. 39

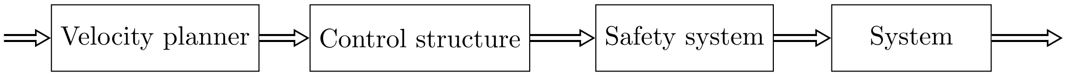


Figure 39. An overview of the controller structure

A digital dead-band was also implemented, to ensure that any unintentional small inputs from the user do not generate any undesired response. The unverified model showed big differences in its dynamics when running a function in a positive or negative direction, due to the counterbalance valves. Because of the big difference, individuals PIDs were implemented for negative and positive direction to give the ability of tuning the two directions independently of each other. Since the controller was developed in Simulink, the controller could be implemented in dSPACE using code-generation and therefore directly tested on the actual system. The chosen sample time for the controller was 20 ms as a result of the sample time of the sensors and the current controller. A rate-limiter structure was also added in the implementation, to ramp input changes and reduce oscillations when the reference is suddenly changed. Results on the actual crane can be found in section 6.2

Chapter 6 Results

This chapter presents the results for model verification and the development of a velocity controller that reduces interferences between functions. Comparisons between the hybrid Simulink-Simscape model and measurements on the real system after manually tuning the model parameters are presented in section 6.1. Additionally, the results of the attempts of using methods to automatically tuning the model are presented. In section 6.2, comparisons of the system when running with and without the proposed controller are presented. This comparison is presented visually in graphs and in a table where the relative error from the average velocity is calculated.

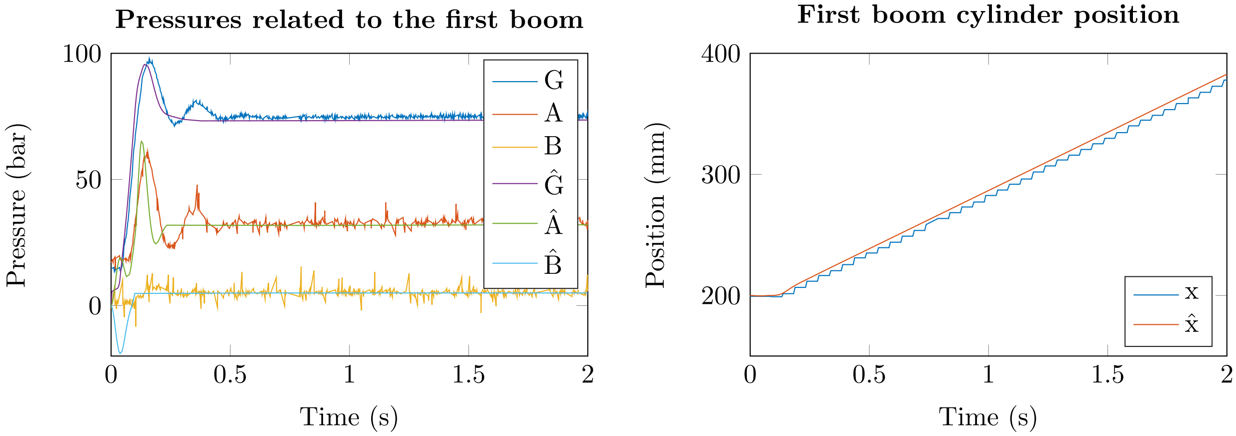
6.1 Modeling and model verification

The modeling and model verification mainly followed the procedures outlined earlier in section 2.5. The first steps were to identify any missing dynamics, obviously erroneous variables or other general issues with the model. As for missing dynamics, a friction model was received from Hiab and introduced to the model. After that point, the main concern was identifying the correct parameter values for the entire system. The mechanical model's CAD drawings were compared to the drawings given by Hiab, the weights and sizes were correct, and therefore inertia and the alike were considered to be correctly calculated by the automagical generation of the SimMechanics block. For this reason, the main tuning of the model was in the hydraulic subsystems.

The figures below display various results, fig. 40 displays the model tuned for the first boom when only the first boom has been run during the testing phase, running test case 1A found in Appendix C. Figure 41 displays the model tuned for the second boom, where only the second boom has been run during the testing phase, running test case 2A. Based on these changes, fig. 42 and fig. 43 displays the result where several functions are run at the same time, running test case 7A. The notation used for the graphs is G, A, B, for the pressures, and x for cylinder position. The hat (e.g. $\hat{G}$) notation is used to indicate the modeled response.

The model for the second boom extension has not been tuned given that there were no pressure sensors mounted on those cylinders. By only being able to look at the cylinder position, it is near impossible to draw any deterministic conclusions. Therefore, no results for that boom are displayed below.

The manual tuning process started with the first boom, where most sensors could be found. First and foremost, the two pressure sensors on each of the cylinder sides. The process started by correcting the flow of the hydraulic oil, since this could now be measured. Based on these results, the position and pressures could be tuned by manipulating lumped parameters with coefficients for the valve section related to the first boom as well as the friction within the cylinder. As can be seen from the graphs in fig. 40a, the main dynamics of the pressure has been captured, but not all of the dynamics. Moreover, the modeled and measured position seem to correspond well (fig. 40b). One anomaly with the model, is that the B side pressure drops to negative values in the beginning. The cause for this has not been found. Since the steady state value matches the measured data, this was deemed not very serious.

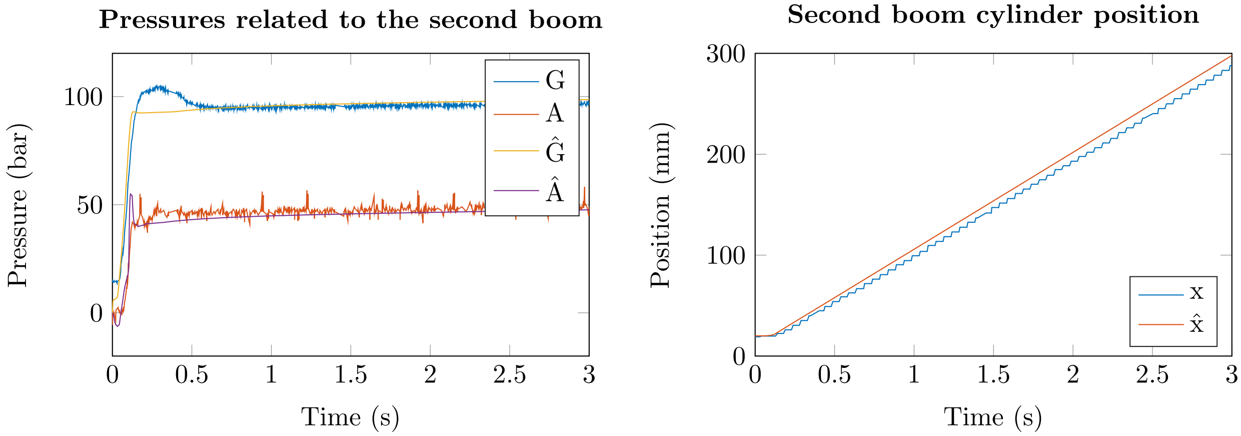


(a) Modeled and measured pressures

(b) Modeled and measured cylinder position

Figure 40. Results of the parameter tuning of the first boom, plotting the response to test case 1A

The same process was applied to the second boom, without changing the pump flow once more, since this is identical. The main tuning points were scaling the lumped parameters with coefficients and friction parameter values. As can be seen in fig. 41 the dynamics are slightly flatter in this case which is also indicated by the model, which does not capture the dynamics fully, but follows the steady state values quite OK for the pressure (fig. 41a). It is more difficult to tune this boom given that the pressure on the B side has to be guessed due to the absence of pressure sensors. The modeled position drifts slightly away from the modeled position (fig. 41b).

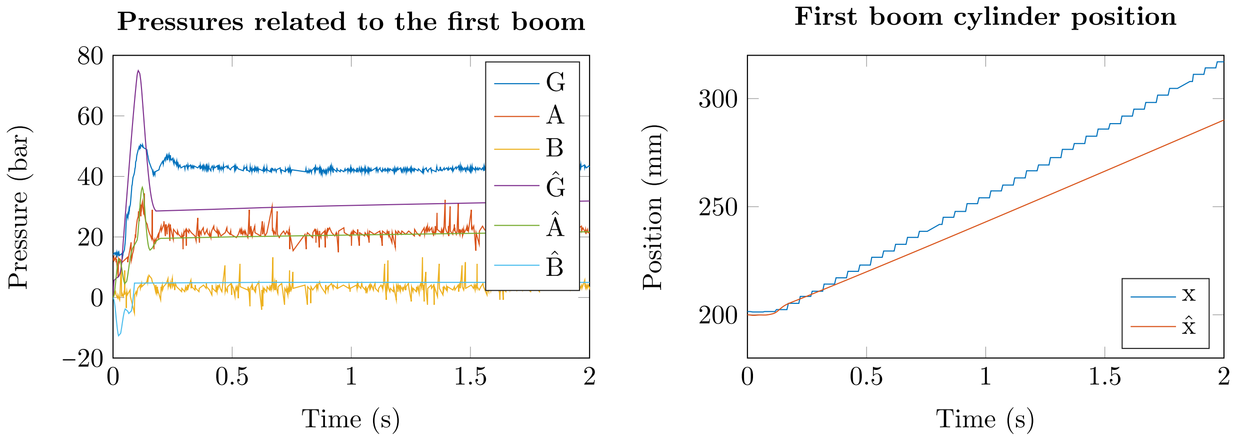


(a) Modeled and measured pressures

(b) Modeled and measured cylinder position

Figure 41. Results of the parameter tuning of the second boom, plotting the response to test case 2A

However, when running test cases where both booms (first and second) are run at the same time, the accuracy of the model drops rapidly. By looking at the first boom, one can see that the gallery pressure buildup is not nearly enough in the model. But the two pressures on either side of the hydraulic cylinder seem to correspond pretty well to the model, see fig. 42a. In turn, this causes the flow to the cylinder to be below that of the real system, causing the modeled position to drift behind the measured position, as can be seen in fig. 42b. Attempts to fix these issues were considered, but doing so without negatively affecting other results was difficult.

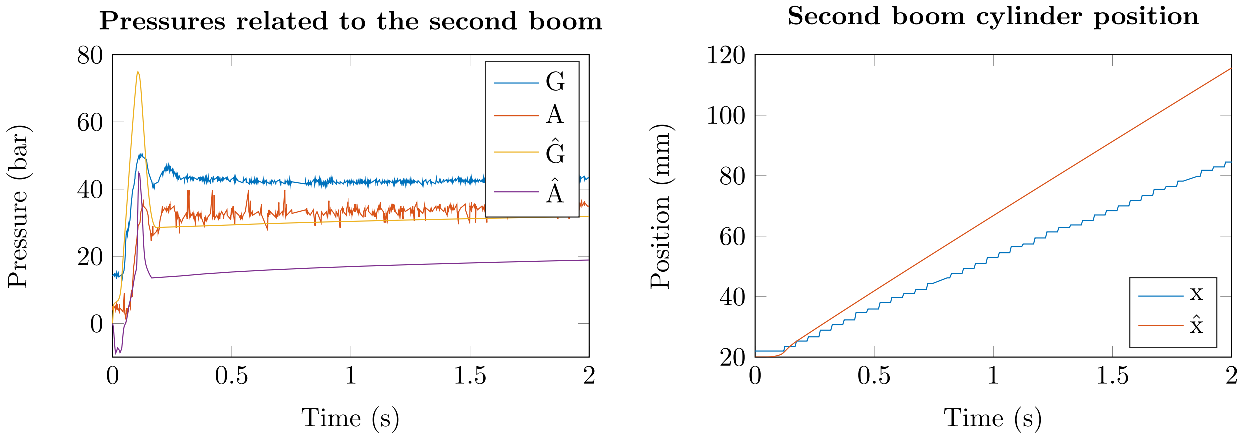


(a) Modeled and measured pressures

(b) Modeled and measured cylinder position

Figure 42. Results of the parameter tuning of the first boom, plotting the response to test case 7A

Similarly, for the second boom the pressure did not correspond well to the system when several actuators were activated at the same time. The gallery pressure here is the same as plotted above in fig. 42a, but here the A side pressure of the cylinder is also too low, as illustrated in fig. 43a. Following the same reasoning, this has led to the flow to the cylinder being above that of the measured one, resulting in the cylinder positioning drifting ahead of the measured position, see fig. 43b.



(a) Modeled and measured pressures

(b) Modeled and measured cylinder position

Figure 43. Results of the parameter tuning of the second boom, plotting the response to test case 7A

The results given by the parameter tuning algorithm did not perform as well as expected. However, there was a significant improvement on the output error of the model in relation to the measured output. The MSE started off with a value near 5000 when using the initia parameters. Although most attempts to run the model were ultimately unsuccessful, due to unstable parameter growth, some attempts did improve on the accuracy of the model.

The initial tests showed convergence of the cost function when tuning the universal parameters *Cd_wc* and *Cd_oc* to the position output of the first boom resulted in convergence close to zero for the cost function. This suggests that the algorithm is functioning as it should and is capable of optimizing the model for lower dimensions. When attempting to tune all the discharge coefficients to all relevant outputs, the algorithm turned out to be unreliable. It showed that it depended highly on the initial guesses in order to be able to find more accurate values for the parameters. The success of the algorithm was also highly dependent of the values of the descent step α and the derivative step h, which were derived through testing.

Most of the 20 values for discharge coefficients in the model were initially set to be the same as the manually tuned parameters. As discharge coefficients should have values between 0 and 1, some were changed to fit this constraint. Some tests were also made by assigning a random value between 0 and 1 as initial values to the parameters.

The test case used for performing the optimization was test case 1A. The optimization only focused on this particular test case in order to have perspective on the different outputs and there was little time to perform optimization on all test cases. Early tests conducted without the safety clause showed some promise in some cases. However, these cases failed to converge over time and the MSE would eventually start to increase at higher iterations. The best set of parameters found during these tests were approximately half of the original value. It is worth noting that it took many attempts with many different initial values to find this set of parameters.

After the safety clause was introduced, the results were considerably better. Although the algorithm didn't always find the best values, it was forced to converge to a local minimum. Through further simulation, using the strategy of only continuing with successful attempts, the MSE decreased significantly. The new parameters improved significantly on the simulated value for the gallery pressure, while the remaining outputs were very similar to the untuned model. Considering that the algorithm was set to optimize the results for the outputs for all cylinder positions as well as all measured pressures, this was unsatisfactory. After adding the weight function to the algorithm, and increasing some weights in order to put emphasis on the outputs that weren't successfully tuned in the previous attempt. The resulting simulations proved to be unstable and the experiment was discontinued and the method was replaced.

After the implementation of the normalized MSE module, the attempted tuning started with the initial parameters used in the untuned model ended up converging to a local minimum close to the initial position. In an effort to try to avoid getting stuck in local minima, the values were given random values between 0 and 1 and tested to see if they were a good fit. This was done by testing each set of random parameters and saving those with a cost function value below a certain threshold. Due to the huge amount of possibilities, this was a time consuming task and it didn't result in any values that wouldn't get stuck in local minima. when changing the algorithm updating procedure to a fixed step, instead of a step proportional to to derivative it was possible to achieve results that were somewhat useful. The cost function decrease in over time was in this case superior to any of the other methods. With this being said, it was not close to zero, which means it was ultimately unsuccessful.

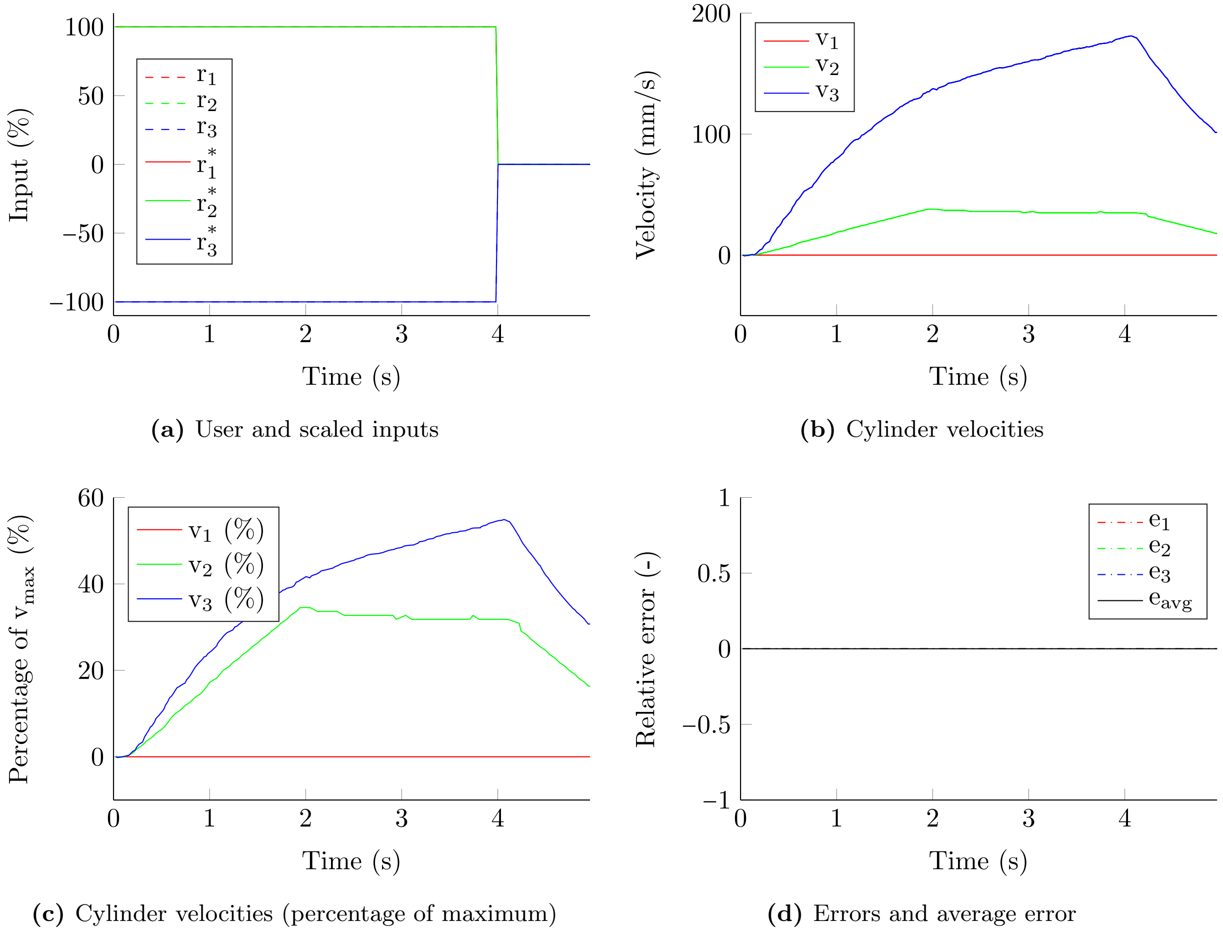
After the cylinder friction parameters were added to the optimization process, it became obvious that the values for these affected the output of the model in a very unpredictable manner. In order to investigate this, a function was created to test the values of the friction in certain intervals. This test gave the optimal values for these parameters. In order to avoid the addition of more parameters to the optimization, these values were chosen as static values for the remainder of the procedure.

6.2 Development of the velocity controller

The results presented in this part is aimed at the data gathered with and without the velocity controller implemented. The idea is to show the effect the controller has on the interference between the cylinders. This is presented in two different ways. First some graphs showing the whole test case running with the corresponding velocities, input and scaled reference, and the error between the requested and measured velocity. Data from two test cases have been plotted, each one with the controller OFF and then the controller switched ON. Then a table containing data from more test cases is introduced to compare the results further. The purpose of the graphs is to show the controller response and how the error converges to one value along with how that impacts the velocities. The table is a more general collection of data and is intended to give an overview of how the controller impacts the dynamics of the movements by taking the average velocity for the cylinders instead of analysing each sample separately.

Test cases containing two or more simultaneous user inputs have been conducted, seen in Appendix H, as the controller is supposed to play an active role only in cases having at least two simultaneous inputs. Each test case has been conducted with and without load (wherever possible), with each iteration conducted with the controller switched ON as well as OFF.

The figures below depict the data collected from test cases running with the controller switched ON and OFF respectively. Each figure contains 4 plots. The first plot contains the input reference signals to the 3 cylinders (r_1, r_2, r_3) as well as the scaled reference signals (r_1*, r_2*, r_3*) outputted from the controller for the corresponding inputs. The second plot contains the velocities sensed for each individual cylinder in mm/s (v_1, v_2, v_3). The third plot shows the velocity as a percentage of the maximum velocity of the same cylinder (v_1, v_2, v_3). In this plot, the controller is expected to make the three values move in the same ratio as the user inputs to the 3 cylinders. The fourth plot is a representation of the individual errors between input reference and sensed velocity of each cylinder (e_1, e_2, e_3) as well as the average error (e_{avg}), all of which are calculated by the controller.



(a) User and scaled inputs

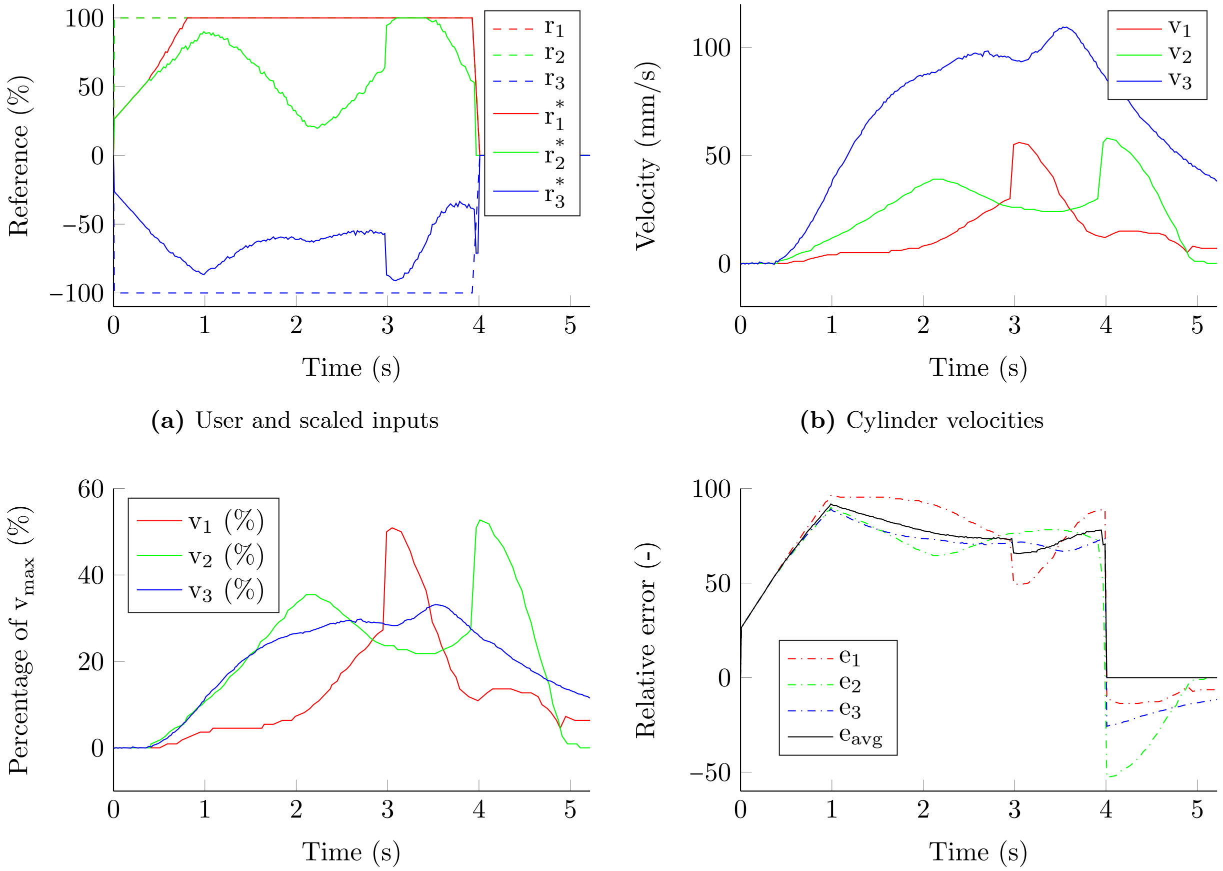
(b) Cylinder velocities

(c) Cylinder velocities (percentage of maximum)

(d) Errors and average error

Figure 44. Test case 7G with the velocity controller disabled and a load attached to increase interference

The data collected from test case 7G, run on a loaded crane, with the controller OFF is depicted in fig. 44. This test case has a step input of 0 to 100 % for the first and second cylinders and 0 to (-100) % for the extension. As the controller is not active, the scaled reference value is the same as user input values in fig. 44a. From fig. 44c and fig. 44d, it is evident that the first cylinder does not move at all. As the controller is not active, there is no error calculation in this case and all values in fig. 44d are plotted zero as expected.



(a) User and scaled inputs

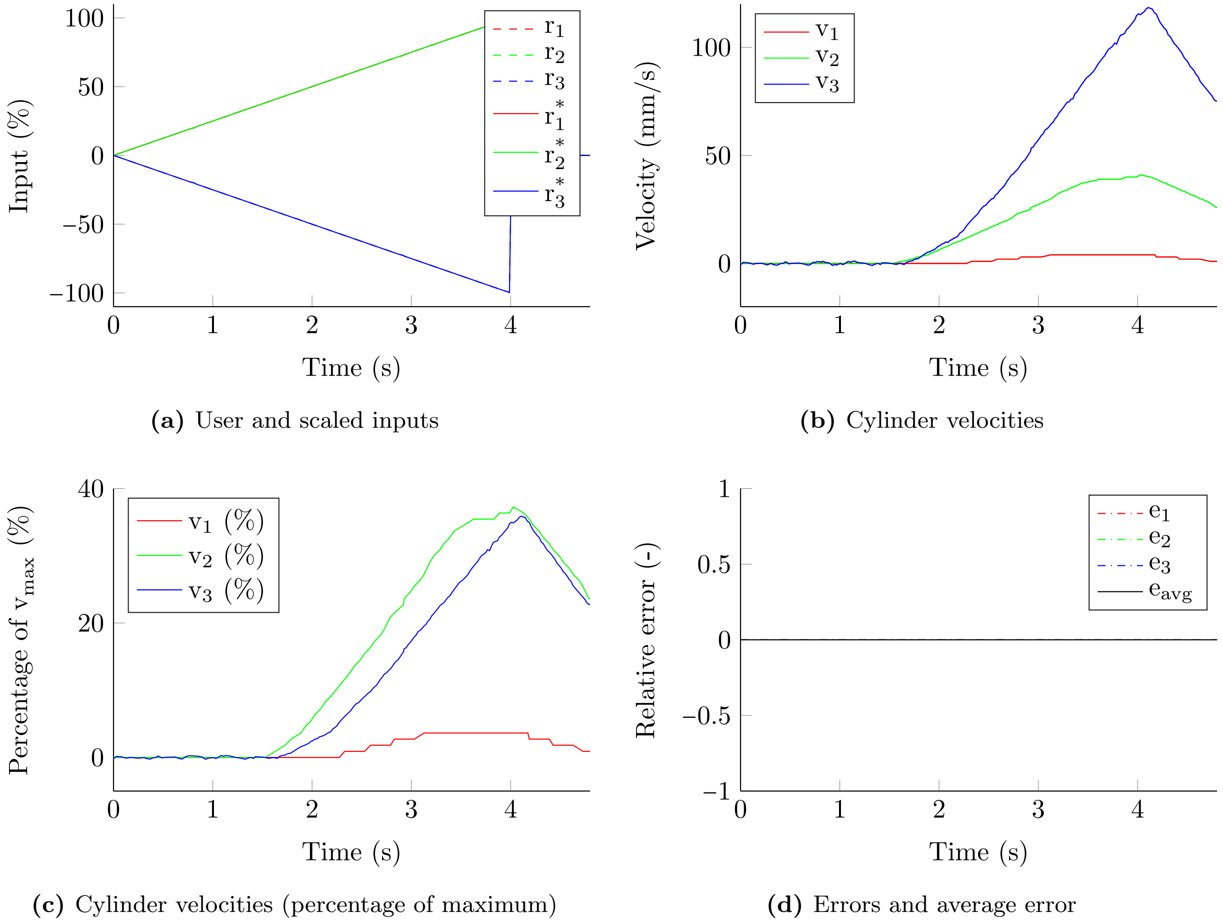
(b) Cylinder velocities

(c) Cylinder velocities (percentage of maximum)

(d) Errors and average error

Figure 45. Test case 7G with the velocity controller enabled and a load attached to increase interference

The data collected for test case 7G on a loaded crane with the controller switched ON is depicted in fig. 45. This test case has a step input of 0 to 100% for the first and second cylinders and 0 to (-100)% for the extension. As the controller is active, the scaled reference values can be observed in fig. 45a (r_1*, r_2*, r_3*). When comparing fig. 45c with fig. 44c, it is evident that the controller plays an active role on modifying the velocities of the cylinders. As all three cylinders have an input of equal magnitude, all values are expected to converge to a similar value in fig. 45d, but the velocity plot could be smoother. The calculated values for the errors (e_1, e_2, e_3)as well as the average error (e_{avg}) can be seen in fig. 45d and this plot seems to be as expected where the errors converge to the average error value.



(a) User and scaled inputs

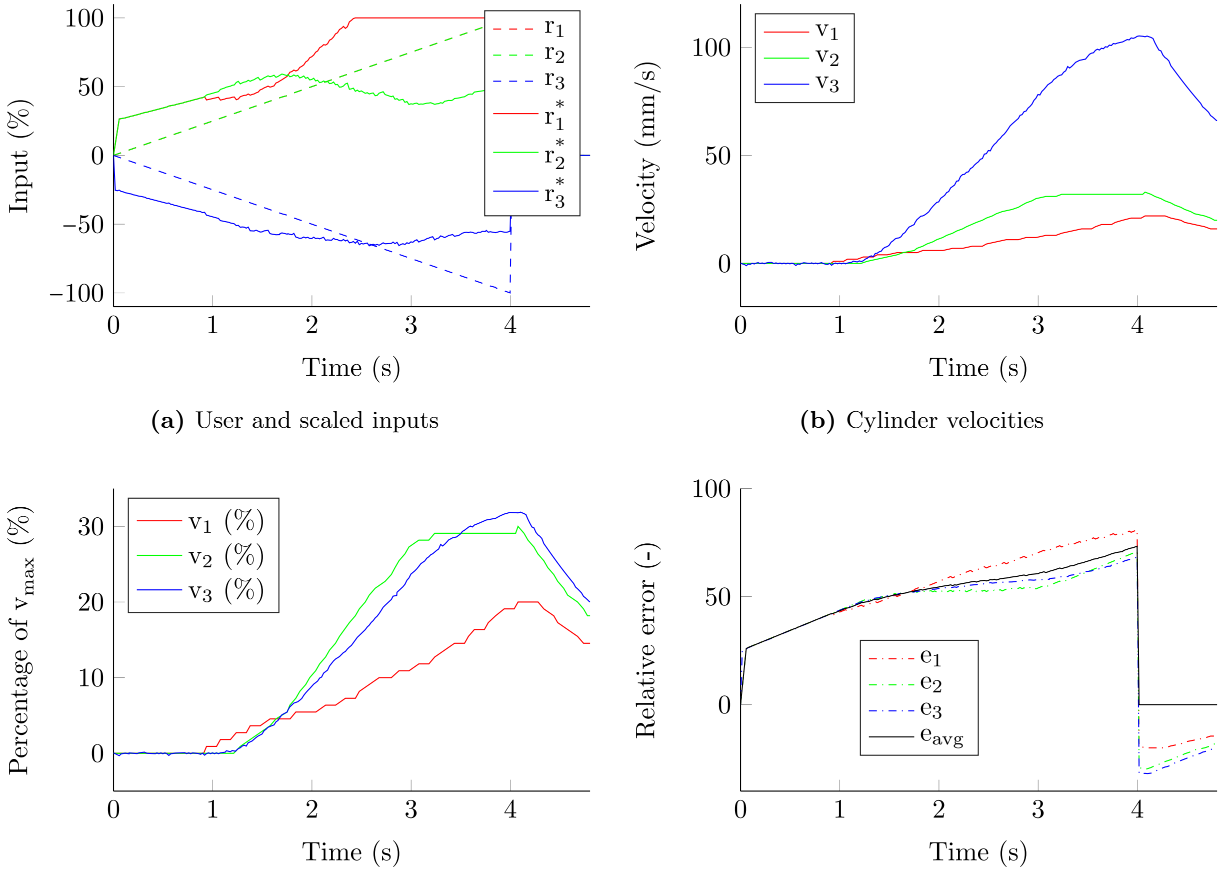
(b) Cylinder velocities

(c) Cylinder velocities (percentage of maximum)

(d) Errors and average error

Figure 46. Test case 8A with the velocity controller disabled and a load attached to increase interference

The data collected for test case 8A on a loaded crane with the controller switched OFF is depicted in fig. 46. This test case has a ramp input over 4 seconds of 0 to 100 % for the first and second cylinders and 0 to (-100) % for the extension. As the controller is not active, the scaled reference value is the same as user input values in fig. 46a. From fig. 46c and fig. 46d, it is evident that the first cylinder does not move at all. As the controller is not active, there is no error calculation in this case and all values in fig. 46d are plotted zero as expected.



(a) User and scaled inputs

(b) Cylinder velocities

(c) Cylinder velocities (percentage of maximum)

(d) Errors and average error

Figure 47. Test case 8A with the velocity controller enabled and a load attached to increase interference

The data collected for test case 8A on a loaded crane with the controller switched ON is depicted in fig. 47. This test case has a ramp input over 4 sec of 0 to 100% for the first and second cylinders and 0 to (-100)% for the extension. As the controller is active, the scaled reference values can be observed in fig. 47a (r_1*, r_2*, r_3*). On comparing fig. 47c with fig. 46c, it is evident that the controller plays an active role on modifying the velocities of the cylinders. As all three cylinders have an input of equal magnitude, all values are expected to converge to a similar value in fig. 47d and it is evident that these values are converging. The calculated values for the errors (e_1, e_2, e_3)as well as the average error (e_{avg}) can be seen in fig. 47d and this plot seems to be as expected where the errors converge to the average error value.

Apart from the ones mentioned for the plots above, more test cases have been conducted and recorded and a comparison between them can help to get an overall picture of the impact of the controller on individual cylinders and the system as a whole. For the comparison of multiple test cases, see table 9 and table 8. The comparison is based on the following three equations.

$$\overline{V_c} = \frac{1}{n}\sum_{i=1}^{n} V_c \cdot \alpha\beta^{-1} \tag{91}$$

Where $\overline{V_c}$ is the average velocity for each cylinder, n is the number of samples, V_c is the velocity at each sample, α is a scaling factor to take the different maximum cylinder velocities into consideration and β is the reference input between zero and one.

$$\Delta\overline{V_c} = \left(\frac{\overline{V_c}}{\frac{1}{m}\sum_{j=1}^{m}\overline{V}_{c,j}} - 1 \right) \cdot 100 \qquad (92)$$

Where $\Delta\overline{V_c}$ is the deviation in percent between the requested and the measured velocity and m is the number of active cylinders.

$$\eta = \frac{\sum_{j=1}^{m} |\Delta\overline{V_c}_{No\,Controller,j} - \sum_{j=1}^{m} |\Delta\overline{V_c}_{Controller,j}}{\sum_{j=1}^{m} \overline{V_c}_{No\,Controller,j}} \cdot 100 \qquad (93)$$

Where η is the improvement with the controller, $\Delta\overline{V_c}_{Controller}$ is the deviations with the controller active and $\Delta\overline{V_c}_{No\,Controller}$ is the deviation without the controller.

Equation (91) gives the average velocity for each cylinder over the whole test case. To be able to compare the different velocities in eq. (92) a scaling factor α is introduced to scale down the extension cylinder velocity to the same velocity as the other two cylinders. Measurements conducted on the crane indicate that the extension has a maximum velocity three times as fast as the other two cylinders. In this case the scaling factor becomes $\alpha = \frac{1}{3}$ and allows the comparison between all the cylinder averages to be done, i.e. it is expected that the extension will move faster than the other cylinders and is taken in consideration. Another scaling factor β is also added to allow for different reference inputs to be used on different cylinders. This factor works as α and rescales the average velocity to be compared with the rest of the cylinders, i.e. a user input of 50 % for example corresponds to a value of $\beta = 0.5$ and doubles the average to compensate for the lesser input.

In eq. (92) the comparison between each cylinder's average velocity and the average of all the cylinders' velocity together is calculated to give a percentage of the error. A positive percentage indicates that the cylinder is moving too fast, while a negative percentage indicates a cylinder moving too slowly. The closer to zero the values are, the closer to the requested velocity the measured values are. To summarize the results for each test case and show the difference between all of the cylinders combined, with and without the controller, eq. (93) is used. The output from this equation gives a percentage of the improvement, meaning the higher the percentage the more improvement shown.

It should be noted that the values obtained using these equations only consider the average velocity of the cylinder over the entire test case. Thus this is a clear indicator of the improvement of individual velocities with respect to input over time and analysis of these values can indicate how interference has been dealt with over the period of a test case. But these values take only the average over a test case cannot indicate how good the controller behaves or how refined or smooth the cylinder movements are at all points of time within the test case. For analysing the behaviour of the crane in depth and understanding the impact of the controller in detail, relevant graphs need to be plotted and studied for each individual test case.

Table 9 and table 8 below shows the results gathered from the predefined test cases where multiple cylinders is supposed to move simultaneously. All the test cases conducted in no load condition are tabulated in table 8 and all the cases conducted with the load attached are tabulated in table 9.

Table 8. Comparison of test case results — No load

		No Controller (% Deviation)			Controller (% Deviation)		
Test Case	Description	First Boom	Second Boom	Extension	First Boom	Second Boom	Extension
7A	Step input 100 % for first and second boom	34	-34	N/A	-4	4	N/A
7B	Step input 100 % for first boom and -100 % for extension	13	N/A	-13	-15	N/A	15
7C	Step input 100 % for second boom and -100 % for extension	N/A	-4	4	N/A	-5	5
7D	Step input -100 % for first and second boom	-6	6	8N/A	-2	2	N/A
7E	Step input -100 % for first boom and 100 % for extension	-18	N/A	18	-14	N/A	14
7F	Step input -100 % for second boom and 100 % for extension	N/A	-17	17	N/A	-19	19
7G	Step input 100 % for first and second boom, step -100 % for extension	42	-38	-4	10	-13	3
7H	Step input -100 % for first and second boom, step 100 % for extension	-18	-7	25	-12	0	12
8A	Ramp input 0 to 100 % for first and second boom, 0 to (-100) % for extension (4 second test)	30	-6	-24	9	-14	5
8B	Ramp input 0 to (-100) % for first and second boom, 0 to 100 % for extension (4 second test)	-24	-23	47	-4	-3	7
8C	Ramp input 0 to 100 % for first and second boom, 0 to (-100) % for extension (2 second test)	6	17	-23	26	-6	-20
8D	Ramp input 0 to (-100) % for first and second boom, 0 to 100 % for extension (2 second test)	-6	-34	40	-9	-3	12

It can be seen from table 8 that in some cases where there are only two inputs, some a slight deterioration can be observed in performance, but this is only marginal. This could be accounted for by considering the dynamics of the system or maybe even improved upon by tuning the parameters in the controller for an optimal behaviour. It can also be observed that the controller improves the performance in all cases which involve simultaneous inputs to all 3 cylinders. Using the tabulated data, the improvement of the controller for all tests in no load condition was calculated to be η_{NoLoad} = 37%.

This value was calculated by using eq. (93) for each of the test cases mentioned in table 8 and taking an average of these.

Table 9. Comparison of test case results - Load

		No Controller (% Deviation)			Controller (% Deviation)		
Test Case	Description	First Boom	Second Boom	Extension	First Boom	Second Boom	Extension
7A	Step input 100 % for first and second boom	-52	52	N/A	4	-4	N/A
7B	Step input 100 % for first boom and -100 % for extension	-59	N/A	59	-12	N/A	12
7C	Step input 100 % for second boom and -100 % for extension	N/A	-22	22	N/A	-9	9
7G	Step input 100 % for first and second boom, step -100 % for extension	-100	20	80	-29	18	11
8A	Ramp input 0 to 100 % for first and second boom, 0 to (-100) % for extension (4 second test)	-86	58	28	-32	17	15
8C	Ramp input 0 to 100 % for first and second boom, 0 to (-100) % for extension (2 second test)	-52	40	12	-35	29	6
8D	Ramp input 0 to (-100) % for first and second boom, 0 to 100 % for extension (2 second test)	10	-6	-4	7	-7	0

It can be observed in table 9 that for all test cases, the controller produces a better performance with respect to the velocity of individual cylinders and the system as a whole. Using the data in table 9, the improvement of using a crane for testing with load has been calculated to be η_{Load} = 61%.

This value was calculated by using eq. (93) for each of the test cases mentioned in table 9 and taking an average of these.

The impact of interference is much more evident for all the cases with load than those without load. Hence, the impact the controller has on the crane is more evident when referring table 9 than referring table 8. A comparison of the values obtained for η_{Load} and η_{NoLoad} shows that interference is much more evident in loaded tests and that the controller has a much higher visible impact when running tests on a loaded crane when compared to an unloaded crane.

As can be seen in the tables above all the cylinders converge to a value of zero when the controller is implemented compared to no controller. The difference can be seen clearer for the test where a step input is introduced as the convergence tends to be faster compared to a ramp input. The test cases vary in running time between two and four seconds. Looking at test 8A and 8C the difference in the end results is very small and indicate that after the initial change the impact of the controller slowly decreases and needs a longer test to completely remove the interference. No data regarding single cylinder movement is presented due to no interference problem occurring. The controller will disregard scaling the input signal in that case and the result will not differ between having the controller implemented or not.

Chapter 7 Discussion and conclusions

This chapter contains a discussion of the results obtained in this project and also conclusions reached from analysis of progress and results. There are sections which discuss specifically about the overall project, the system and prerequisites, modeling and model verification and controller development.

7.1 Project overview

This section comprises of the reflections on the project as a whole and a discussion about the overall project, including the project plan and the execution of the project based on this plan. The risk analysis and its impact on the project will also be discussed in this section, along with a perspective on potential changes that could have caused a significant improvement in efficiency or the project result as a whole.

The project was approached using Scrum with bi-weekly sprints. This worked quite well with the help of visual representation using a white board and sticky notes. The idea of using the organization structure to assign responsibilities was aimed at having individuals who directly follow up on specific tasks within the project and ensure that the work planned on that task is carried out as per the project plan.

This has been used effectively to an extent but the implementation has scope for improvement which would lead to much more efficiency within the team. A risk analysis was performed at the start of this project and in retrospect, quite a few of the identified risks were observed at some point of time in the project. The project plan included a Gantt chart with a detailed time plan and it can now be stated that some of the set deadlines were not met. Requirements were drafted by the team and they seemed quite reasonable, but the ability to realize them was not considered by someone with experience in the field at the start of the project. A document management convention was prepared at the start of the project, with an emphasis on maintaining a regularly updated backup of files on SVN. The document naming convention was followed for a considerable part of the project but there is still room for improvement.

A few suggested changes which could be incorporated during the planning phase of the project are

- Checking all hardware and material handed down from other projects before preparing the time plan as hardware or software that does not work as expected causes delays that put the project off track. A suggested approach would involve first checking handed down material, then preparing requirements based on a dynamic active discussion with all stakeholders and then preparing the project plan and time schedule only once this is completed.
- The time plan could have some buffer time incorporated at major milestones within the project, or even a floating buffer time which could be used whenever the project gets delayed due to unavoidable circumstances which cannot be accounted for while preparing the time plan. Identifying dependencies in the plan would also highlight how some tasks are more critical than other and help a team identify priorities during the project.
- The focus during risk analysis should be on steps that can be taken to avoid, mitigate or manage all possible risks rather than just identifying all risks. Perhaps the risk analysis could be referred to around every milestone, so as to refresh the teams approach to any risk faced.
- Investigating available document management techniques which work across operating systems and platforms suitable to the team before selecting one.

- Imbibing a sense of ownership for tasks that each member is responsible for from the start of the project and maintaining the same responsibilities till the end would be a good change which could lead to a higher overall efficiency.
- Defining the set of requirements in discussion with all stakeholders (teaching team and company) during a meeting so as to get input on the difference between what is desirable and what exactly is achievable within the existing boundaries and constraints. It is very important to formulate precise requirements which can be realized and preparing test cases which directly relate to the fulfillment of these requirements if needed.
- The visual representation for Scrum could also be updated on an online platform, wherein a more dynamic work environment can be achieved. A Google spreadsheet, which was used for the final sprint of this project, could be a good solution for more effective and dynamic resource and time allocation if used right from the onset.
- Having two team members in charge of all communication external to the team as well as having two persons who can be contacted directly at the company whenever needed, in order to ensure that redundancy is obtained in terms of ideas and knowledge when there is an interaction between the team and stakeholders.

7.2 System and prerequisites

There are two different systems connected to the test rig at KTH from which the crane can be run, dSPACE and IFM electronics. In this project it was intended to start implementing and testing the controller in dSPACE to further tune and verify it while later transfer the results to the IFM controller. The system used at Hiab is IFM electronics, however the choice to start the implementation in dSPACE were done based on some of the pros and cons of the systems.

At the start of the second cycle of this project the dSPACE system had been implemented on the crane by KTH staff. Along with the installation of the system a control desk interface and a corresponding Simulink model were also created. However, this interface and model was not designed for this project. Because of this some changes had to be done in order to fit our needs.

For the students undertaking this project, prior knowledge about dSPACE weighed a heavy role in the decision to start the using this system. This also allowed the controller to be tested earlier in the project. dSPACE also supports changing parameters while running the crane which facilitated the work of tuning the controller. dSPACE can also generate code from Simulink models. This introduces an easy-to-use common interface that simplifies testing of models and controllers.

A lot of changes were done to the interface within the time span of the project. Different interface options were added along the way when they were needed. The most obvious change is the addition of the slider interface to run the crane directly from Control desk. These sliders were added and connected to all the running modes except for the test cases. This allowed the testing to be done faster by letting the user run the crane from the computer station instead of using the leavers on the physical crane. Along with these sliders, tunable parameters for the velocity controller were also added in order to have the possibility to change them between tests without recompiling, expediting the tuning process. Besides from actual implementing additional features to the interface, a face lift were done to the overall interface removing unnecessary parts and adding graphs to see the sensor values in real time. In short, the changes done to the interface was done too increase the user-friendliness of the interface but also make running the crane less time-consuming.

As mentioned, the controller is to be implemented on the IFM controller. This controller together with the development suite CODESYS is currently used at Hiab. This suite contains libraries specific for crane operation. Along with prebuilt functions supplied by IFM electronics, Hiab also provides a safety library designed particularly for some malfunctioning that might occur during crane operation. These type of safety measures are not implemented in the dSPACE system, which is something that has been considered. During operation, caution has to be taken and there has always been two operators with hardware based emergency stops in the case of malfunctions.

During the project, there was a number of delays due to the hardware setup itself. For instance the sufficient sensors were not in place. This could likely have been redeemed earlier, should the project group have realized this at an earlier stage. For instance, setting up dSPACE and its corresponding sensor value conversions took several iterations, adding dither currents and current controllers, adding a flow sensor, changing the pump, etc., took longer than expected and probably longer than necessary. However, these were the conditions that the group had to work around. It would have been favorable if these conditions could have been noticed in an early project stage, preferably during the spring, and discovered at the same time. As it was now, the project ran into several road blocks after one another, causing substantial delays.

The flow sensor should have been ordered at the beginning of the project. Then at early stages of the project we could have realized the large flow variations, and acquired a pump more suitable for this task. A lot of time analyzing the system has thus been squandered because of an assumed parameter, that has had high impact on model behavior, has been incorrect. Likewise, we have realized that there has been an insufficient amount of pressure sensors mounted on the crane. Without being able to see the pressures, on B side of second boom cylinder and on both sides of the extension cylinder, it has not been possible to overview how the whole system responds to different parameter changes. By eliminating as many unknown pressures as possible, the model analysis process could be facilitated.

Tests conducted on the crane showed that the response from the current controllers were too slow which in return impacted the results for the velocity controller. The current controllers were redesigned to be as fast as possible, but still without any overshoot, the result was the controller presented in section 3.2.2. Up until this was solved any modeling or control development was unfruitful. The dither currents and the new current controller were necessary to ensure that the correct spool position would be reached at all times. The model that was given to us by the previous group included a transfer function for the spool behavior. By getting the spool position as close as possible to the reference with a quick response time, it may be possible to disregard from that transfer function when doing the model verification, e.g. if a reference input directly corresponds to a certain spool displacement. The current controller is also used as a part of a cascaded control structure when it comes to controlling the piston velocities, requiring a fast response time.

It is also worth mentioning that the use of wire encoders and the optical distance sensor currently mounted on the crane is not a feasible solution on a commercial crane, which often is operated in harsh environments. Those sensors are currently only mounted out of convenience; they are easy to interface and have low noise. In real world scenarios, the exposed wires of the wire encoders would be prone to getting caught by lifted objects or falling debris which is frequently occurring during operation. Additionally, the optical distance sensor relies on both its lens and the reflective panel being clean, which is difficult to maintain during operation. Such incidents may cause the sensors to output undetermined values, causing unexpected system behavior. Use of these sensors on the commercial crane would lead to more frequent maintenance, unreliability and also, safety concerns. A more suitable solution could be to use the accelerometers mounted on the first and second boom to extract angular velocity and position. Since an accelerometer wont be sufficient for the extension boom, another solution has to be considered there. One could consider using a wire encoder there, if it is mounted at the inside in the hollow space the booms have. At this position it will be better protected from external disturbance.

The test cases have been designed to mimic realistic usage of the crane in as many situations as possible. In commercial use of the crane the remote controller is assumed to be operated with smooth finger movements. The required response is in most situations slow because the average crane movements are slow due to involved large forces and heavy loads. Heavy objects are usually lifted with slings, thus horizontal acceleration of the crane tip is preferably low to avoid rocking. This is also to avoid stress on the booms and joints caused of dynamic torque. The interference problem is more evident during cylinder extraction because of the gravity force. While conducting tests, it was observed that the test cases which involve the crane moving downwards seemed unsafe when carried out with the load connected due to the considerable oscillations observed. Hence, it was decided by the team to conduct loaded tests only on the cases where the crane cylinders were extracted, i.e. input to the first two booms being positive. All the defined test cases must be run to complete the verification of the controller. It is suggested that the omitted test cases be run in a closed controlled environment wherein there are no safety concerns to bystanders, operator or risk of material damage.

7.3 Modeling and model verification

The outcome of the modeling and model verification was not successful. By manually tuning the model, it was possible to achieve decent results for individual booms and cases where only a single actuator was used. However, when using more than one actuator at the same time, it has not been possible to tune the model such that all functions in the model displays the same behavior as the measured system. Additionally, the tuning process was to change parameters that are lumped, e.g. multiplied with other parameters. This essentially leads to introducing a scaling factor for the entire set of parameters in that specific equation. Therefore, the verification of the model is not deterministic and the true parameter values are still unknown. For control purposes, this is of no importance, but if the goal is to have each parameter individually determined then another procedure has to be followed.

Although some improvements were shown on the results of the model when gradient descent algorithm was implemented in respect to the measured output of the crane, it can be seen that the algorithm wasn't the most suitable for this application. As it requires partial derivatives of all parameters, the process becomes tediously slow which is unacceptable as the model needs to be rerun many times with different initial values in order to have a chance of finding the global minimum. The fact that the model was to be tuned to fit several outputs also made it difficult to create a suitable cost function that would consider all outputs in an accurate manner. The implementation of weight vectors and relative MSE calculation could have made it possible to optimize the model if further development and testing was made. This was not possible due to lack of time.

Another possible explanation to the unsatisfactory results would be that the model subject to optimization was ill-defined. The gradient descent methods expects the model to be correct and can only refine the parameters of the system. Due to the complexity of the system it is very likely that this could be a problem. The complexity of the model also made it difficult to select parameters for tuning. As the model contains a large amount of parameters, many of which were guessed or estimated, it became challenging to select the appropriate parameters as there was no obvious correlation with each parameter and the output of the model.

It is also possible that the noisy outputs given by the crane could deteriorate the performance of the algorithm. As the cost function operates under mean square error, outlying points will heavily affect its value. This could be solved by filtering the measured data.

The gradient descent algorithm did help to improve the results, however it would require more tuning and analysis for it to be efficient. If the model is actually well defined, it should be possible to tune it using the algorithm implemented, however it would require some further work and a decrease in the level of uncertainty about the parameter values. It will also be necessary to run the algorithm on several of the test cases, and not only on one in order to validate the model.

The main obstacles when it comes to verifying the models developed have been identified to be of either of two categories; unmeasurable parameters and system scale. For instance, one of the major assumptions of the system was that the hydraulic oil flow was constant, but unknown. When a flow sensor was introduced to determine the flow, it was discovered that the flow was not constant. The attempts to verify the model before this flow sensor was acquired were meaningless since the flow has a very large impact on both pressure and consequently piston velocities. Similarly, the absence of pressure sensors on the second boom and second boom extension makes it difficult, if not impossible, to tune the model response to be the same of the measured system. Additional pressure sensors were delivered by Hiab, but there was not enough time to safely install and use these on the system setup.

As for the system scale, the sheer number of parameters that can be modified is large, but more importantly; the variables are interconnected and one parameter change affects several parts of the dynamics. This raises the question whether another approach should be used for the verification. For example, individual testing of the various components somehow could make it possible to determine the unknown parameters without the risk of other parameters influencing the results. Ideally, it would be desired to test the components individually in a test rig to determine e.g. the viscous friction within the cylinders. Attempting to do this when the cylinders are put into the context of the system is unlikely possible.

Also, the dynamics of the counterbalance valves were considered to have a great impact on why it was so difficult to verify a model of the system. A test was proposed as seen in Appendix G to investigate the dynamics of the three counterbalance valves separately in a hydraulic test rig to reduce the uncertainty of the system parameters. Unfortunately for the outcome of the project, the test was not specified in time for Hiab to conduct the tests in their hydraulic test rig. If the possibility to verify the parameters for the different counterbalance valves is possible in future work, it is strongly suggested. To recieve the data from the suppliers of these valves would be even better if possible.

This project considered two separate models, the aforementioned “hybrid model” which was inherited from previous year’s work, and the newly developed “Simscape model”. Some work was done to the hybrid model to include missing dynamics as well as rearranging the system to be more easily overviewed. The two models have their respective benefits and drawbacks related to modeling, verification, and application. The hybrid model allows for a high degree of freedom in choice of complexity. This is due to the fact that the (hydraulic) model is composed of equations where each parameter introduced is a conscious act by the developer. On the other hand, the Simscape model includes libraries for physical modeling of systems including mechanical and hydraulic systems. These libraries have a predetermined set of parameters that have to be numerically set. Due to this, it is not uncommon that the number of parameters gets too large for e.g. auto-tuning methods such as gradient descent. However, the actual modeling of the system (disregarding the parameter values) is much more straightforward since the system is simply modeled in direct comparison to the real system. If parts were to be verified separately, Simscape may be preferred since then the number of parameters can be kept sufficiently low and the modeling itself is simply a task of drag-and-drop.

Both the hybrid model and the Simscape model includes the same mechanical model (based on Simscape). Since this subsystem is automagically generated from a CAD model, we cannot be sure that it is in fact correct. The system behavior appears correct, and the weight and sizes calculated by the CAD programs correspond to the drawing received by Hiab whereas resulting inertia and the alike are calculated by the program. An attempt to create a model based on equations for the mechanical system was done; mainly because of these uncertainties but also because it is preferred to use the equations when a control strategy is developed. However, this proved difficult and the simulations were not successful due to geometric loops. More time could be invested in developing this type of an equation based model in order to have full control of the entire system as a whole.

Given the current stakeholder requirements, specifically that the model should describe the dynamics of the system with enough precision so that a velocity controller can be developed, the models presented above may be sufficient. But if the idea is to identify the system precisely, and that controller performance is a very crucial part of the requirements, then a lot of work is still required. Clarifying the stakeholder requirements further might be necessary.

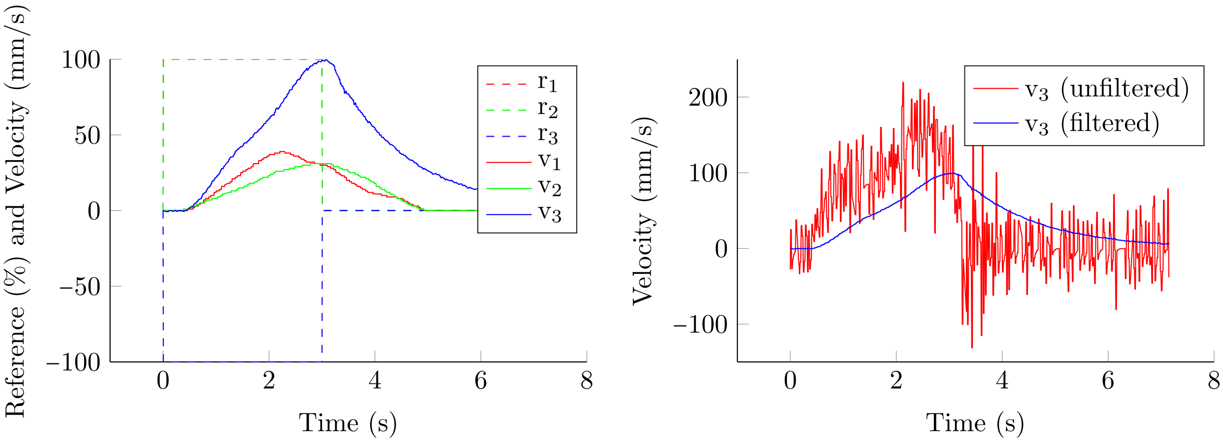
7.4 Development of the velocity controller

Our proposed control structure was the error difference minimizing structure which rendered positive results in some test cases in reducing interference between the functions. The structure is easy to implement in another control system and only relies on fixed values of maximum velocities of each function. By using dual PIDs, one for negative movement and one for positive per function, the tuning is more extensive but a greater performance can be achieved. Since the direction of movement has such a great impact on the system dynamics, it was considered better to have the ability to have individual parameters for different directions. One known problem with the controller is that the maximum measured velocity for a single function should never be achieved when multiple functions are running since this produces a negative error. Increasing the preset maximum velocity value could help minimize this risk, but this would also cause the controller to try to force the system to go faster than what the user demands. The slew has not been tested due to safety concerns mentioned before, but implementing this or another hydraulic function in the existing control structure is easy since the controller can easily be extended to include more functions. Since the controller should reduce the interference between functions, the controller is offline when only one function is active since there is no interference to regulate. The controller does not use any mathematical operations that is not available in any programming language, hence implementation in CODESYS on the IFM controller is possible.

Since unexpected movements of the crane can prove quite hazardous, the safety of the user and the surrounding environment has been considered. The implementation of a safety layer ensures that the crane can never move in the direction opposite to the user's input, thus the possibility of a hazard due to movement in unexpected directions is reduced. The rate limiter structure filters the user's input to be smoother, making the response a bit slower but also limits the possibility of sudden movements.

The problems concerning hardware discussed above in section 7.2, and the choice to wait for a verified model before testing the controller on the real system made the testing phase shorter then desired. Since implementation in dSPACE is a matter of pressing a button, there was still enough time to run all intended test cases but the controller could not be tuned for the best performance in terms of speed and robustness. Due to time constraints, the crane was tested only under two conditions, namely no load testing and loaded testing. Loaded testing was conducted for only one load, a sand bag which weighs approximately 250 kg. It is not certain how the controller reacts to a heavier load than the one used, except in simulation on an unverified model.

When implementing the controller, it was noticed that the sensors' accuracy and speed had a detrimental effect on the performance of the controller. The laser encoder on the extension boom produces a very noisy signal which needs to be filtered heavily in order to a get a good measurement, as can be seen in figure fig. 48b. The wire encoders for the first and second booms have a long delay which makes the controller regulate the input based on previous values instead of the current velocity as can be seen in figure fig. 48a. Since the sensors are connected to the CAN network, the sampling time for the sensors has been limited to 20 ms. This sampling time and the cascaded control structure with the current controller as an inner-loop makes the lowest possible sampling time for the velocity controller also 20 ms. These limitations in the hardware setup, the sensor delay and the sampling time make the controller perform worse than expected. A possibility would been to implement these limitations in the model so that the controller is built on these prerequisites.



(a) Showing the lag between user input and measured values

(b) Showing the measured velocity for extension

Figure 48. Depiction of the issues related to the implementation of the controller

The structure using the error difference between the average error and the function's error resembles the structure proposed in [chapter adaptive reference model]. Since the error for each function converges to the average, there is some resemblance to the non-linear sliding mode controller mentioned in chapter 2. The difference is that our proposed structure is a linear structure with individual PIDS instead of having a MIMO system built on equations. It was decided to first try linear control structures to the extent that none showed positive results since more advanced non-linear structures were considered too time consuming to learn. After the minimizing the error difference structure gave positive indications on the unverified model, more effort on fixing the problems found on the actual system were spent and the investigation for other control theories was put on hold.

When comparing the results to the system requirements found in Appendix E, almost none of the stated requirements in the beginning of the project were achieved. These were on the other hand derived with no information or experience in making a controller for reducing the interference between functions on a hydraulic crane. When instead comparing to the stakeholder requirements Appendix D, many of the requirements are achieved to some extent. The crane follows the different user references better when compared to the absence of a controller. As seen in test case 7G (as described in table 9), the first boom does not move when the load is lifted but with the controller all functions move. The only used sensors are the sensors already installed on the crane used, these sensors however are not the same sensors used on a commercial crane and in a future work more investigation which sensors that could replace the wire encoders needs to be done.

Chapter 8 Future work and recommendations

This chapter contains recommendations for future tasks related to this project specifically, and generally for other projects where hydraulic and mechanical components are to be modeled and controlled. The ideas presented here are a logical response to what the group working with this project consider necessary to complete the task. Moreover, it contains ideas for the potential of completely new tasks as a direct result of the conclusions drawn in this project.

8.1 System and prerequisites

The load used during testing should have its weight determined. Since no force sensor is available, acquiring such a sensor should be considered. The scale could be a load cell, preferably of S-type, which is designed for measuring axial forces. The sandbag currently used is estimated to weigh 250 kg, therefore a force sensor rated 10 kN would be suited. Most force sensors also need a transceiver module for 4–20 mA interfacing.

More robust sensors that are not as prone to crane vibration as inclinometers (which have considerable delay due to noise filtering) may be resolvers or "encapsulated potentiometers" used on for example boat stern drives which are submersed in water. Those could be used instead of inclinometers or wire encoders by mounting them directly to the boom joints either on the outside or on the inside of the booms for more protection. This might be a more robust solution, also with the advantage of higher refresh rates. A drawback is that they do not account for the vehicle orientation relative gravity, which would require at least one inclinometer for vehicle orientation. This should be investigated if future research concludes that using only pressure sensors or using inclinometers in collaboration with pressure sensors is impracticable. As mentioned in the discussion, in order to validate the model, installing additional pressure sensors on both sides of the second boom extension as well as on the B side of the second boom is necessary.

8.2 Modeling and model verification

The existing models calculate the dynamics of the open center valve which could likely be simplified substantially without major drawbacks to the system behavior as a whole. One way of simplification could be to lump the valve sections together and consider them as one. This could be something to consider if control strategies such as model predictive control is used which calculates the system response before deciding the control signal. In this case, the calculations have to be done in real-time.

Moreover, the newly developed friction model and load model may also need some refinements. Optimization has been done in order to facilitate simulation and reduce simulation time. If the load rocking is to be captured by the model, then the load model has to be revised. These dynamics are not included today, but are a substantial interference on real systems.

The existing models also assume that the counterbalance valves only depend on the pressure in the cylinders, while in reality these types of valves are both pressure and flow dependent. Since no data exists for the counterbalance valves, the parameters values for these will have to be determined by testing. The aforementioned testing procedure that is presented in Appendix G needs to be revised since the initial conditions of the tests are not possible to create. A constant flow that is sufficiently low to keep the valve closed until the pressure $V1$ is high enough, is not possible. Since a closed valve doesn't let any flow through, maintaining a constant flow during the entire test is not possible. Another indicator than a constant pressure $V2$ to show that the valve is fully open is also needed. If $V2$ is connected to the tank, the pressure will be constant during the entire measurement period. A strategy how the different work ports should be connected needs to be done in order to make a successful test. Additional tests for the extension and second boom counterbalance valve needs to be added since they are double acting.

Since the large number of unknown parameters create difficulties in verifying a model of the system, another strategy could be breaking down the system into smaller subsystems. Removing different cylinders from the system and test them in a hydraulic lab is tedious and time consuming, but if given the possibility this should be done in order to get a successful outcome of model verification. If it is desired to use auto-tuning methods to find parameter values of the system as a whole, then the amount of parameters have to be substantially reduced somehow. Specifically, the current SimMechanics model for the mechanical subsystem of the crane contains many variables which pose a problem for such methods. The alternative is assuming that the mechanical model is correct and verified, which would not be very conclusive, and then ignore these parameters in the tuning process.

As the attempt to tune the parameters using a gradient descent algorithm was inconclusive, a different type of method should be considered in order to optimize the model. As stated in the SOTA, there are plenty of other alternatives, some of which would be more suitable to the model at hand. From the fact that a small change in a single parameter would change the outcome of the model quite drastically, one can infer that the quadratic optimization surface created by the MSE was not smooth enough to effectively be used in a derivative-based algorithm. An attempt to optimize the model using a different approach could be suitable. This should be done with one of the algorithms that don't require any derivatives, such as the Nelder-Mead simplex search or a direct search approach, such as pattern search or another method mentioned by Rios and Sahinidis (2013).

As the current model includes a large set of inaccurate or guessed variables, much consideration should be made when selecting which parameters should be tuned. A large set of dimensions will lead to convergence-issues on most algorithms, especially when using such an unpredictable model and noisy measured data. Considering this, the tuning of this particular model will require a lot of work and it is far from guaranteed that the finalized model will fulfill the criteria that was initially set in the requirements phase. With this being said, it is likely that a more thorough and well thought out algorithm could at least improve on the result significantly. As the method currently used (gradient descent) is a derivative-based algorithm, it is not optimized for the current model, as the derivative of the cost function can't be analytically calculated something that compromises both the accuracy and the efficiency of the model.

Finally, in order to make the different models more user-friendly, the notations should be altered to conform to a general standard. This would preferably be the notation used at Hiab since all drawings and documents from the company has a standard notation. The hybrid model is also somewhat inconsistent in where equations are introduced compared to the actual physical layout of the system. An example is that the subsystems concerning the counterbalance valves are inside the open-center valve subsystem in the model, which could be confusing for someone not experienced with the model since the actual counterbalance valves are mounted directly on the cylinders.

8.3 Development of the velocity controller

After implementing the controller and testing it on the crane some concerns and questions arose which lay as a basis for some of the suggested actions in this future work section. Along with these suggestions some intended work and implementations that were planned but not completed is also added as a possible continuation to the project.

Starting with the controller logic, by introducing more tunable parameters could further increase the controllability . One proposition is to add different proportional values whether the measured velocity is higher or lower than the requested velocity. By implementing this logic, tuning of the controller would allow the control signal to not saturate as fast. Resulting in a faster and smoother response. Besides from adding this logic, further tuning of the controller would also enhance performance.

Considering the implementation of the controller, the rate limiter used to create a smoother input signal, impacts the movement of the crane in both a negative and positive way. Further tuning of the rate limiter regarding the slope might increase the performance of the controller, and a better balance between speed and smoothness could be achieved. Another solution is also looking into the possibility of removing the rate limiter and tune the controller itself to have a smoother reaction to a sudden change in the reference signal.

When testing the controller on the real system, a testing facility without physical limitations would allow for longer test cases as well as larger movements. This should result in more test data which could be take into consideration when tuning the controller, resulting in better fine-tuned parameters.

The implementation of the controller is in Simulink and runs using dSPACE and CAN communication. Hiab uses a different system for running their cranes that is built around the software program CODESYS. The control logic along with all sensor readings and corresponding communication should because of that be implemented in CODESYS. This can be done by either utilizing MATLABs code generating application to convert the Simulink model directly to CODESYS or by manually writing the control logic in the CODESYS software.

Bibliography

Cargotec, “Hiab XS 022 Brochure,” 2015, accessed April, 2015. [Online]. Available: http://www.hiab.com/en/global/products1/loader-cranes-100/hiab-10/products/-51/selected-brand-hiab/

J. Wang, “A non-model-based controller used in process control,” in *Proceedings of the 1997 36th IEEE Conference on Decision and Control*, vol. 1. IEEE, 1997, pp. 832–837.

M. Johnsson and M. Moradi, *PID Control - New identification and Design Methods.* Springer, 2005.

L. Ljung and T. Glad, *Reglerteknik : grundläggande teori.* Studentlitteratur AB, 2006.

W. S. Levine, *The Control Handbook.* CRC Press LCC, 1996.

R. Babuska, “Lead-lag position control of a hydraulic actuator,” 2011, accessed April, 2015. [Online]. Available: http://www.dcsc.tudelft.nl/~wb2207/transp/examples/hydraulic/hydraulic_lead_lag.html

Q. Xiong, W. Cai, M. He, and M. He, “Decentralized Control System Design for Multivariable Processes – A novel method based on Effective Relative gain array,” *Industrial & engineering chemistry research*, vol. 45, no. 8, pp. 2769–2776, 2006.

T. Glad and L. Ljung, *Multivariable and Nonlinear Methods.* CRC Press, 2000.

B. Anderson, A. Bennick, and M. Salciccioli, “Model Predictive Control,” 2007, accessed May, 2015. [Online]. Available: https://controls.engin.umich.edu/wiki/index.php/MPC

S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design.* John Wiley & Sons, 2005.

M. Athans, “A Tutorial on the LQG/LTR Method.” Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 1986.

——, “The role and use of the stochastic linear-quadratic-Gaussian problem in control system design,” *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 529–552, 1971.

X. C. M. Cubillos and L. C. G. de Souza, “Using of H-infinity control method in attitude control system of rigid-flexible satellite,” *Mathematical Problems in Engineering*, vol. 2009, 2010.

J. K. Hedrick and A. Girard, “Control of Nonlinear Dynamic Systems,” University of California, Tech. Rep., 2005.

M. Lemmen and M. Bröcker, “Different Nonlinear Controllers for Hydraulic Synchronizing Cylinders,” University of Duisburg, Department of Measurement and Control, Tech. Rep., 2000.

J. H. Kwon, T. hyeong Kim, J.-S. Jang, and I. yeong Lee, “Feedback linearization control of a hydraulic servo system,” in *SICE-ICASE, 2006. International Joint Conference.* IEEE, 2006, pp. 455–460.

R. Husson, Ed., *Control Methods for Electronic Machines.* ISTE Ltd, 2009.

W. Perruquetti and J. P. Barbot, *Sliding Mode Control in Engineering.* CRC Press, 2002.

P. Jain and D. M. Nigam, “Design of a model reference adaptive controller using modified MIT rule for second order system,” *Advanced in electronic and electric engineering*, vol. 3, no. 4, pp. 477–484, 2013.

B. Vinagre, I. Petráš, I. Podlubny, and Y. Chen, “Using fractional order adjustment rules and fractional order reference models in model-reference adaptive control,” *Nonlinear Dynamics*, vol. 29, no. 1-4, pp. 269–279, 2002.

M. Keshmiri, A. F. Jahromi, A. Mohebbi, M. H. Amoozgar, and W.-F. Xie, “Modeling and control of ball and beam system using model based and non-model based control approaches,” *International Journal on smart sensing and intelligent systems*, vol. 5, no. 1, pp. 14–35, 2012.

K. M. Passino and S. Yurkovich, *Fuzzy Control.* Addison-Wesley Longman, Inc., 1998.

M. M. Bech, T. O. Andersen, H. C. Pedersen, and L. Schmidt, “Experimental Evaluation of Control Strategies for Hydraulic Servo Robot,” *Mechatronics and Automation*, pp. 342–347, 2013.

A. Bonchis, P. I. Corke, and D. C. Rye, “Experimental Evaluation of position Control Methods for Hydraulic Systems,” *Control Systens Technology*, vol. 10, no. 6, pp. 876–882, 2002.

J. Kalmari, J. Backman, and A. Visala, “Nonlinear model predictive control of hydraulic forestry crane with automatic sway damping,” *Computers and Electronics in Agriculture*, vol. 109, pp. 36–45, 2014.

L. Ljung and T. Glad, *Modellbygge och simulering.* Studentlitteratur AB, 2004.

R. A. Monzingo, R. L. Haupt, and T. W. Miller, *Introduction to Adaptive Arrays, 2nd edition.* SciTech Publishing, 2011.

J. R. Raol, G. Girija, and J. Singh, *Modeling and Parameter Estimation of Dynamic Systems.* IET, 2004.

G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning.* Springer, 2013.

T. Daimon and M. Goto, “The mean squared error optimum design criterion for parameter estimation in nonlinear regression models,” *Communications in Statistics - Theory and Methods*, vol. 37, no. 4, pp. 508–519, 2008.

J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, “Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions,” *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

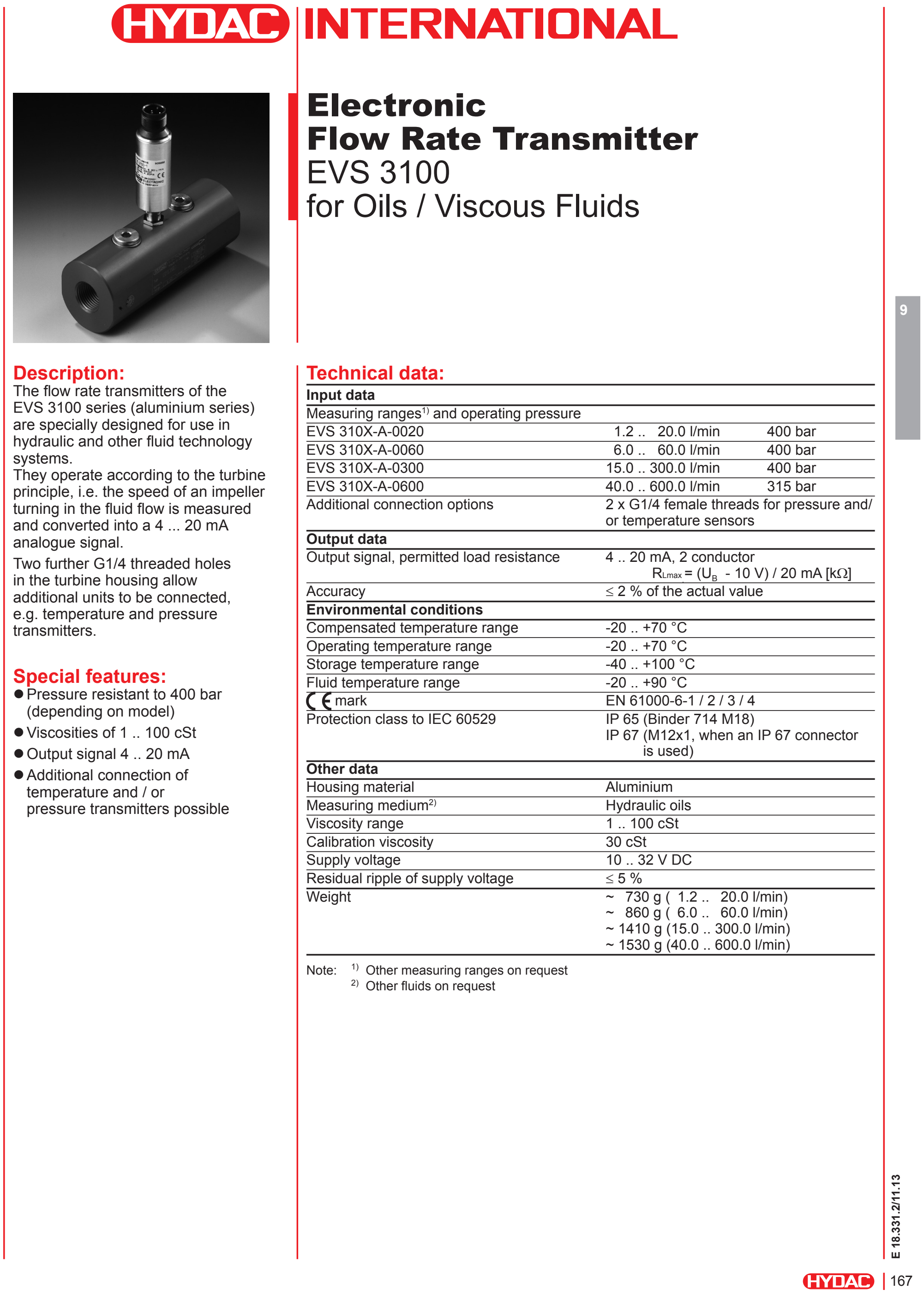
V. Torczon, “On the convergence of pattern search algorithms,” *SIAM Journal on Optimization*, vol. 7, no. 1, pp. 1–25, 1997.

R. Hooke and T. A. Jeeves, “Direct Search, Solution of Numerical and Statistical Problems,” *Journal of the ACM (JACM)*, vol. 8, no. 2, pp. 212–229, 1961.

L. M. Rios and N. V. Sahinidis, “Derivative-free optimization: a review of algorithms and comparison of software implementations,” *Journal of Global Optimization*, vol. 56, no. 3, pp. 1247–1293, 2013.

Appendices

Appendix A EVS3100 Datasheet



HYDAC INTERNATIONAL

Electronic Flow Rate Transmitter

EVS 3100
for Oils / Viscous Fluids

9

Description:

The flow rate transmitters of the EVS 3100 series (aluminium series) are specially designed for use in hydraulic and other fluid technology systems.
They operate according to the turbine principle, i.e. the speed of an impeller turning in the fluid flow is measured and converted into a 4 ... 20 mA analogue signal.

Two further G1/4 threaded holes in the turbine housing allow additional units to be connected, e.g. temperature and pressure transmitters.

Special features:

- Pressure resistant to 400 bar (depending on model)
- Viscosities of 1 .. 100 cSt
- Output signal 4 .. 20 mA
- Additional connection of temperature and / or pressure transmitters possible

Technical data:

Input data		
Measuring ranges[1] and operating pressure		
EVS 310X-A-0020	1.2 .. 20.0 l/min	400 bar
EVS 310X-A-0060	6.0 .. 60.0 l/min	400 bar
EVS 310X-A-0300	15.0 .. 300.0 l/min	400 bar
EVS 310X-A-0600	40.0 .. 600.0 l/min	315 bar
Additional connection options	2 x G1/4 female threads for pressure and/ or temperature sensors	
Output data		
Output signal, permitted load resistance	4 .. 20 mA, 2 conductor $R_{Lmax} = (U_B - 10\ V) / 20\ mA\ [k\Omega]$	
Accuracy	≤ 2 % of the actual value	
Environmental conditions		
Compensated temperature range	-20 .. +70 °C	
Operating temperature range	-20 .. +70 °C	
Storage temperature range	-40 .. +100 °C	
Fluid temperature range	-20 .. +90 °C	
CE mark	EN 61000-6-1 / 2 / 3 / 4	
Protection class to IEC 60529	IP 65 (Binder 714 M18) IP 67 (M12x1, when an IP 67 connector is used)	
Other data		
Housing material	Aluminium	
Measuring medium[2]	Hydraulic oils	
Viscosity range	1 .. 100 cSt	
Calibration viscosity	30 cSt	
Supply voltage	10 .. 32 V DC	
Residual ripple of supply voltage	≤ 5 %	
Weight	~ 730 g (1.2 .. 20.0 l/min) ~ 860 g (6.0 .. 60.0 l/min) ~ 1410 g (15.0 .. 300.0 l/min) ~ 1530 g (40.0 .. 600.0 l/min)	

Note: [1] Other measuring ranges on request
[2] Other fluids on request

E 18.331.2/11.13

HYDAC | 167

Model code:

EVS 3 1 0 X – A – XXXX – 000

Housing material
0 = Aluminium

Electrical connection
4 = Male 4 pole Binder series 714 M18
(connector not supplied)
6 = Male M12x1, 4 pole
(connector not supplied)

Signal
A = 4 .. 20 mA, 2 conductor

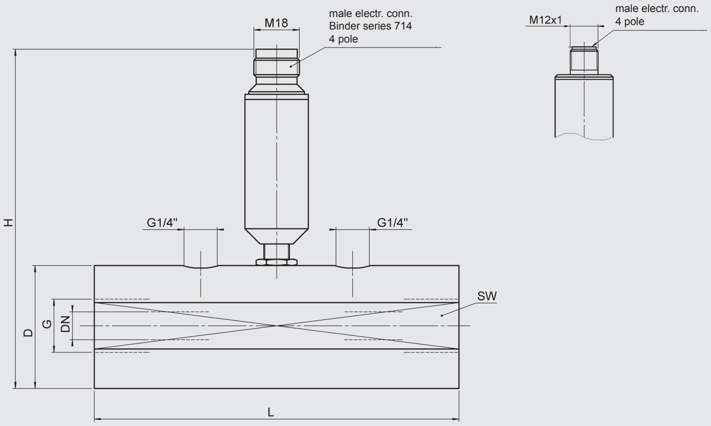
Measuring range
0020 = 1.2 .. 20 l/min
0060 = 6.0 .. 60 l/min
0300 = 15.0 .. 300 l/min
0600 = 40.0 .. 600 l/min

Modification number
000 = Standard

Note:
On instruments with a different modification number, please read the label or the technical amendment details supplied with the instrument.

Accessories:
Appropriate accessories, such as electrical connectors, can be found in the Accessories brochure.

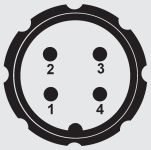
Dimensions:



Model	Meas. range [l/min]	L [mm]	H [mm]	D / SW [mm]	G	Torque value [Nm]	DN [mm]
EVS 310X-A-0020	1.2 .. 20	117	135	47 / 46	G¼"	60	7
EVS 310X-A-0060	6 .. 60	144	135	48.5 / 46	G½"	130	11
EVS 310X-A-0300	15 .. 300	155	150	63.5 / 60	G1¼"	500	22
EVS 310X-A-0600	40 .. 600	181	150	63.5 / 60	G1½"	600	30

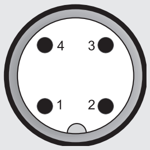
Pin connections:

Binder series 714 M18



Pin	EVS 3104-A
1	reserved
2	Signal +
3	Signal -
4	reserved

M12x1



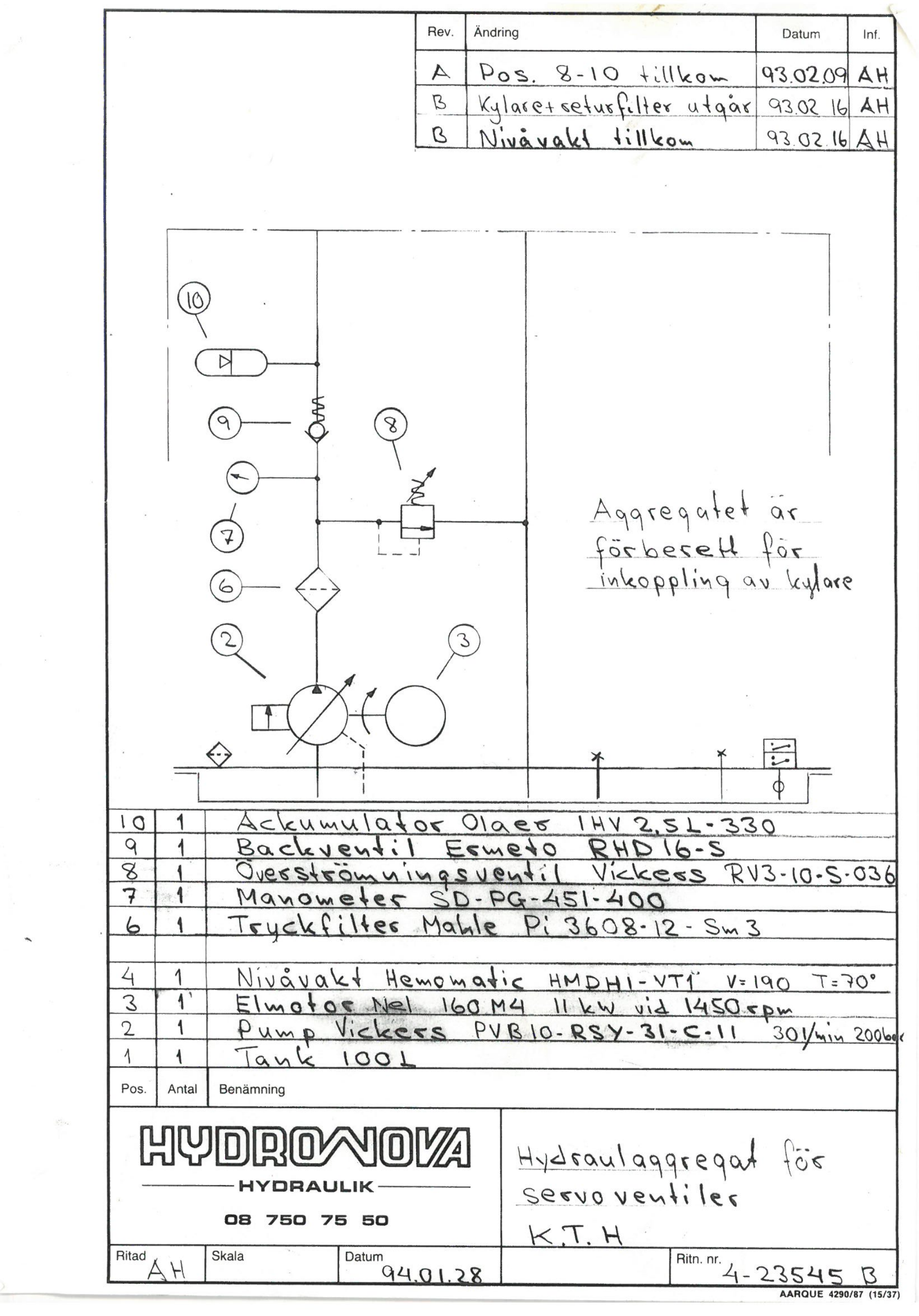
Pin	EVS 3106-A
1	Signal +
2	reserved
3	Signal -
4	reserved

Note:

The information in this brochure relates to the operating conditions and applications described.
For applications or operating conditions not described, please contact the relevant technical department.
Subject to technical modifications.

HYDAC ELECTRONIC GMBH
Hauptstraße 27, D-66128 Saarbrücken
Telephone +49 (0)6897 509-01
Fax +49 (0)6897 509-1726
E-mail: electronic@hydac.com
Internet: www.hydac.com

Appendix B Hydronova solution datasheet



Rev.	Ändring	Datum	Inf.
A	Pos. 8-10 tillkom	93.02.09	AH
B	Kylare+returfilter utgår	93.02.16	AH
B	Nivåvakt tillkom	93.02.16	AH

Aggregatet är förberett för inkoppling av kylare

Pos.	Antal	Benämning
10	1	Ackumulator Olaer IHV 2,5L-330
9	1	Backventil Ermeto RHD16-S
8	1	Överströmningsventil Vickers RV3-10-S-036
7	1	Manometer SD-PG-451-400
6	1	Tryckfilter Mahle Pi 3608-12-Sm3
4	1	Nivåvakt Hemomatic HMDH1-VT1 V=190 T=70°
3	1	Elmotor Nel 160 M4 11 kw vid 1450 rpm
2	1	Pump Vickers PVB10-RSY-31-C-11 30l/min 200ba
1	1	Tank 100L

HYDRONOVA
HYDRAULIK
08 750 75 50

Hydraulaggregat för servo ventiler
K.T.H

Ritad	Skala	Datum	Ritn. nr.
AH		94.01.28	4-23545 B

AARQUE 4290/87 (15/37)

Appendix C Test cases for controller validation

Each case in the input matrix should be performed once for each case in the initial state matrix, i.e. a test case which contains 4 input cases and 2 initial state cases needs 8 different test instances.

The velocities stated refer to the velocity relative to the previous fixed point, i.e. the angular velocity of the second boom is relative to the angular velocity of the first boom. All velocities are also relative to the scaled values, i.e. the velocity calculated in order to avoid interference.

Table C.1. Nomenclature

ω	Angular velocity
V	Linear velocity
$f1$	First boom
$f2$	Second boom
$f3$	Extension boom
θ	Angle
X	Distance
m	Modeled system
r	Real system
t	Rise time
p	Pressure
g	Gallery
u	upper
l	lower

1. Requirement: The simulated steady state angular velocity of the first boom shall be within a 10% margin of the real system during 90% of the time.

Table C.2. Test case 1 initial state

State (position)	Case a	Case b	Case c	Case d
Function 1	N/A	N/A	N/A	N/A
Function 2	$\theta_{f2,max}$	$\theta_{f2,min}$	$\theta_{f2,max}$	$\theta_{f2,min}$
Function 3	$X_{f3,max}$	$X_{f3,min}$	$X_{f3,min}$	$X_{f3,max}$

Table C.3. Test case 1 Expected outcome

State/input	Case 1	Case 2
Case a	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$
Case b	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$
Case c	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$
Case d	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$

Table C.4. Test case 1 Expected outcome

State/input	Case 3	Case 4
Case a	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$
Case b	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$
Case c	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$
Case d	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$	$\omega_{f1,m} = \omega_{f1,r} \pm 10\%$

2. Requirement: The simulated steady state angular velocity of the second boom shall be within a 10% margin of the real system during 90% of the time

Table C.5. Test case 2 input values

Input (angular velocity)	Case 1	Case 2	Case 3	Case 4
Function 1	0	0	0	0
Function 2	$\omega_{f2,max}$	$\omega_{f2,max}/2$	$-\omega_{f2,max}/2$	$-\omega_{f2,max}$
Function 3	0	0	0	0

Table C.6. Test case 2 initial state

Input (angular velocity)	Case 1	Case 2	Case 3	Case 4
Function 1	$\theta_{f1,max}$	$\theta_{f1,min}$	$\theta_{f1,max}$	$\theta_{f1,min}$
Function 2	N/A	N/A	N/A	N/A
Function 3	$X_{f3,max}$	$X_{f3,min}$	$X_{f3,min}$	$X_{f3,max}$

Table C.7. Test case 2 expected outcome

State/input	Case 1	Case 2
Case a	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$
Case b	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$
Case c	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$
Case d	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$

Table C.8. Test case 2 expected outcome

State/input	Case 3	Case 4
Case a	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$
Case b	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$
Case c	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$
Case d	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$	$\omega_{f2,m} = \omega_{f2,r} \pm 10\%$

3. Requirement: The simulated steady state velocity of the third boom shall be within a 10% margin of the real system during 90% of the time.

Table C.9. Test case 3 input values

Input (angular velocity)	Case 1	Case 2	Case 3	Case 4
Function 1	0	0	0	0
Function 2	0	0	0	0
Function 3	$V_{f3,max}$	$V_{f3,max}/2$	$-V_{f3,max}/2$	$-V_{f3,max}$

Table C.10. Test case 3 initial state

State (position)	Case a	Case b	Case c	Case d
Function 1	$\theta_{f1,max}$	$\theta_{f1,min}$	$\theta_{f1,max}$	$\theta_{f1,min}$
Function 2	$\theta_{f2,max}$	$\theta_{f2,min}$	$\theta_{f2,min}$	$\theta_{f2,max}$
Function 3	N/A	N/A	N/A	N/A

Table C.11. Test case 3 expected outcome

State/input	Case 1	Case 2
Case a	$V_{f3,m} = V_{f3,r} \pm 10\%$	$V_{f3,m} = V_{f3,r} \pm 10\%$
Case b	$V_{f3,m} = V_{f3,r} \pm 10\%$	$V_{f3,m} = V_{f3,r} \pm 10\%$
Case c	$V_{f3}, m = V_{f3,r} \pm 10\%$	$V_{f3,m} = V_{f3,r} \pm 10\%$
Case d	$V_{f3,m} = V_{f3,r} \pm 10\%$	$V_{f3,m} = V_{f3,r} \pm 10\%$

Table C.12. Test case 3 expected outcome

State/input	Case 3	Case 4
Case a	$V_{f3,m} = -V_{f3,r} \pm 10\%$	$V_{f3,m} = -V_{f3,r} \pm 10\%$
Case b	$V_{f3,m} = V_{f3,r} \pm 10\%$	$V_{f3,m} = V_{f3,r} \pm 10\%$
Case c	$V_{f3}, m = V_{f3}, r, \pm 10\%$	$V_{f3}, m = V_{f3,r} \pm 10\%$
Case d	$V_{f3,m} = V_{f3,r} \pm 10\%$	$V_{f3,m} = V_{f3,r} \pm 10\%$

4. Requirement: The rise time of the model with regard to the velocity of first boom shall be within a 10% margin of the real system for a step input.

Table C.13. Test case 4 input values

Input (angular velocity)	Case 1	Case 2
Function 1	$-\omega_{f1,max}$ to $\omega_{f1,max}$	$\omega_{f1,max}$ to $-\omega_{f1,max}$
Function 2	0	0
Function 3	0	0

Table C.14. Test case 4 input values

Input (angular velocity)	Case 3	Case 4
Function 1	0 to $-\omega_{f1,max}/2$	0 to $\omega_{f1,max}/2$
Function 2	0	0
Function 3	0	0

Table C.15. Test case 4 input values

Input (angular velocity)	Case 5	Case 6
Function 1	$-\omega_{f1,max}$ to $\omega_{f1,max}$	$\omega_{f1,max}$ to $-\omega_{f1,max}$
Function 2	$\omega_{f2,max}$	$\omega_{f2,max}$
Function 3	$-V_{f3,max}$	$V_{f3,max}$

Table C.16. Test case 4 input values

Input (angular velocity)	Case 7	Case 8
Function 1	0 to $-\omega_{f1,max}/2$	0 to $\omega_{f1,max}/2$
Function 2	$-\omega_{f2,max}$	$-\omega_{f2,max}$
Function 3	$V_{f3,max}$	$-V_{f3,max}$

Table C.17. Test case 4 initial state

State (position)	Case a
Function 1	N/A
Function 2	N/A
Function 3	N/A

Table C.18. Test case 4 expected outcome

State/input	Case 1	Case 2
Case a	$t_{f1,m} = t_{f1}, r \pm 10\%$	$t_{f1,m} = t_{f1}, r \pm 10\%$

Table C.19. test case 4 expected outcome

State/input	Case 3	Case 4
Case a	$t_{f1,m} = t_{f1}, r \pm 10\%$	$t_{f1,m} = t_{f1}, r \pm 10\%$

Table C.20. test case 4 expected outcome

State/input	Case 5	Case 6
Case a	$t_{f1,m} = t_{f1}, r \pm 10\%$	$t_{f1,m} = t_{f1}, r \pm 10\%$

Table C.21. Test case 4 expected outcome

State/input	Case 7	Case 8
Case a	$t_{f1,m} = t_{f1,r} \pm 10\%$	$t_{f1,m} = t_{f1,r} \pm 10\%$

5. Requirement: The rise time of the model with regard to the velocity of second boom shall be within a 10% margin of the real system for a step input.

Table C.22. Test case 5 input values

Input (angular velocity)	Case 1	Case 2
Function 1	0	0
Function 2	$-\omega_{f2,max}$ to $\omega_{f2,max}$	$\omega_{f2,max}$ to $-\omega_{f2,max}$
Function 3	0	0

Table C.23. Test case 5 input values

Input (angular velocity)	Case 3	Case 4
Function 1	0	0
Function 2	0 to $-\omega_{f2,max}/2$	0 to $\omega_{f2,max}/2$
Function 3	0	0

Table C.24. Test case 5 input values

Input (angular velocity)	Case 5	Case 6
Function 1	$\omega_{f1,max}$	$\omega_{f1,max}$
Function 2	$-\omega_{f2,max}$ to $\omega_{f2,max}$	$\omega_{f2,max}$ to $-\omega_{f2,max}$
Function 3	$-V_{f3,max}$	$V_{f3,max}$

Table C.25. Test case 5 input values

Input (angular velocity)	Case 7	Case 8
Function 1	$-\omega_{f1,max}$	$-\omega_{f1,max}$
Function 2	0 to $-\omega_{f2,max}/2$	0 to $\omega_{f2,max}/2$
Function 3	$V_{f3,max}$	$-V_{f3,max}$

Table C.26. Test case 5 initial state

State (position)	Case a
Function 1	N/A
Function 2	N/A
Function 3	N/A

Table C.27. Test case 5 Expected outcome

State/input	Case 1	Case 2
Case a	$t_{f2,m} = t_{f2,r} \pm 10\%$	$t_{f2,m} = t_{f1}, r \pm 10\%$

Table C.28. Test case 5 Expected outcome

State/input	Case 3	Case 4
Case a	$t_{f2,m} = t_{f1,r} \pm 10\%$	$t_{f2,m} = t_{f1,r} \pm 10\%$

Table C.29. Test case 5 Expected outcome

State/input	Case 5	Case 6
Case a	$t_{f2,m} = t_{f1,r} \pm 10\%$	$t_{f2,m} = t_{f1,r} \pm 10\%$

Table C.30. Test case 5 Expected outcome

State/input	Case 7	Case 8
Case a	$t_{f2,m} = t_{f1,r} \pm 10\%$	$t_{f2,m} = t_{f1,r} \pm 10\%$

6. Requirement: The rise time of the model with regard to the velocity of extension boom shall be within a 10% margin of the real system for a step input.

Table C.31. Test case 6 input values

Input (angular velocity)	Case 1	Case 2
Function 1	0	0
Function 2	0	0
Function 3	$-V_{f3,max}$ to $V_{f3,max}$	$V_{f3,max}$ to $-V_{f3,max}$

Table C.32. Test case 6 input values

Input (angular velocity)	Case 3	Case 4
Function 1	0	0
Function 2	0	0
Function 3	0 to $-V_{f3,max}/2$	0 to $V_{f3,max}/2$

Table C.33. Test case 6 input values

Input (angular velocity)	Case 5	Case 6
Function 1	$\omega_{f1,max}$	$-\omega_{f1,max}$
Function 2	$-\omega_{f2,max}$	$\omega_{f2,max}$
Function 3	$-V_{f3,max}$ to $V_{f3,max}$	$V_{f3,max}$ to $-V_{f3,max}$

Table C.34. Test case 6 input values

Input (angular velocity)	Case 7	Case 8
Function 1	$-\omega_{f1,max}$	$-\omega_{f1,max}$
Function 2	$\omega_{f2,max}$	$-\omega_{f2,max}$
Function 3	0 to $-V_{f3,max}/2$	0 to $V_{f3,max}/2$

Table C.35. Test case 6 initial state

State (position)	Case a
Function 1	N/A
Function 2	N/A
Function 3	N/A

Table C.36. Test case 6 expected outcome

State/input	Case 1	Case 2
Case a	$t_{f3,m} = t_{f3,r} \pm 10\%$	$t_{f3,m} = t_{f3,r} \pm 10\%$

Table C.37. Test case 6 expected outcome

State/input	Case 3	Case 4
Case a	$t_{f3,m} = t_{f3,r} \pm 10\%$	$t_{f3,m} = t_{f3,r} \pm 10\%$

Table C.38. Test case 6 expected outcome

State/input	Case 5	Case 6
Case a	$t_{f3,m} = t_{f3,r} \pm 10\%$	$t_{f3,m} = t_{f3,r} \pm 10\%$

Table C.39. Test case 6 expected outcome

State/input	Case 7	Case 8
Case a	$t_{f3,m} = t_{f3,r} \pm 10\%$	$t_{f3,m} = t_{f3,r} \pm 10\%$

7. The gallery pressure during steady state shall be within a 10% margin of the real system in all cases during 90% of the time.

Table C.40. Test case 7 input values

State/input	Case 1	Case 2
Function 1	$\omega_{f1,max}$	0
Function 2	0	$\omega_{f2,max}$
Function 3	0	0

Table C.41. Test case 7 input values

State/input	Case 3	Case 4
Function 1	0	0
Function 2	0	$-\omega_{f2,max}$
Function 3	$V_{f3,max}$	$-V_{f3,max}$

Table C.42. Test case 7 input values

State/input	Case 5	Case 6
Function 1	$-\omega_{f1,max}$	$-\omega_{f1,max}$
Function 2	0	$-\omega_{f2,max}$
Function 3	$-V_{f3,max}$	0

Table C.43. Test case 7 input values

State/input	Case 7	Case 8
Function 1	$\omega_{f1,max}$	$-\omega_{f1,max}$
Function 2	$\omega_{f2,max}$	$-\omega_{f2,max}$
Function 3	$V_{f3,max}$	$V_{f3,max}/2$

Table C.44. Test case 7 input values

State/input	Case 9	Case 10
Function 1	$-\omega_{f1,max}$	$\omega_{f1,max}/2$
Function 2	$\omega_{f2,max}/2$	$-\omega_{f2,max}$
Function 3	$-V_{f3,max}$	$-V_{f3,max}$

Table C.45. Test case 7 initial state

State (position)	Case a
Function 1	N/A
Function 2	N/A
Function 3	N/A

Table C.46. Test case 7 expected outcome

State/input	Case 1	Case 2
Case a	$p_{g,m} = p_{g,r} \pm 10\%$	$p_{g,m} = p_{g,r} \pm 10\%$

Table C.47. Test case 7 expected outcome

State/input	Case 3	Case 4
Case a	$p_{g,m} = p_{g,r} \pm 10\%$	$p_{g,m} = p_{g,r} \pm 10\%$

Table C.48. Test case 7 expected outcome

State/input	Case 5	Case 6
Case a	$p_{g,m} = p_{g,r} \pm 10\%$	$p_{g,m} = p_{g,r} \pm 10\%$

Table C.49. Test case 7 expected outcome

State/input Case 7	Case 8	
Case a	$p_{g,m} = p_{g,r} \pm 10\%$	$p_{g,m} = p_{g,r} \pm 10\%$

Table C.50. Test case 7 expected outcome

State/input	Case 9	Case 10
Case a	$p_{g,m} = p_{g,r} \pm 10\%$	$p_{g,m} = p_{g,r} \pm 10\%$

8. The upper side pressure for the first boom during steady state shall be within a 10% margin of the real system in all cases during 90% of the time.

Table C.51. Test case 8 input values

Input (angular velocity)	Case 1	Case 2
Function 1	$\omega_{f1,max}$	$-\omega_{f1,max}$
Function 2	0	0
Function 3	0	0

Table C.52. Test case 8 input values

Input (angular velocity)	Case 3	Case 4
Function 1	$\omega_{f1,max}$	$\omega_{f1,max}/2$
Function 2	$\omega_{f2,max}$	$-\omega_{f2,max}$
Function 3	$V_{f3,max}$	$-V_{f3,max}$

Table C.53. Test case 8 input values

Input (angular velocity)	Case 5	Case 6
Function 1	$-\omega_{f1,max}$	$-\omega_{f1,max}/2$
Function 2	$\omega_{f2,max}$	$-\omega_{f2,max}$
Function 3	$V_{f3,max}$	$-V_{f3,max}$

Table C.54. Test case 8 initial state

State (position)	Case a
Function 1	N/A
Function 2	N/A
Function 3	N/A

Table C.55. Test case 8 expected outcome

State/input	Case 1	Case 2
Case a	$p_{1u,m} = p_{1u,m} \pm 10\%$	$p_{1u,m} = p_{1u,m} \pm 10\%$

Table C.56. Test case 8 expected outcome

State/input	Case 3	Case 4
Case a	$p_{1u,m} = p_{1u,m} \pm 10\%$	$p_{1u,m} = p_{1u,m} \pm 10\%$

Table C.57. Test case 8 expected outcome

State/input Case 5	Case 6	
Case a	$p_{1u,m} = p_{1u,m} \pm 10\%$	$p_{1u,m} = p_{1u,m} \pm 10\%$

9. The lower side pressure for the first boom during steady state shall be within a 10% margin of the real system in all cases during 90% of the time.

Table C.58. Test case 9 input values

Input (angular velocity)	Case 1	Case 2
Function 1	$\omega_{f1,max}$	$-\omega_{f1,max}$
Function 2	0	0
Function 3	0	0

Table C.59. Test case 9 input values

Input (angular velocity)	Case 3	Case 4
Function 1	$\omega_{f1,max}$	$\omega_{f1,max}/2$
Function 2	$\omega_{f2,max}$	$-\omega_{f2,max}$
Function 3	$V_{f3,max}$	$-V_{f3,max}$

Table C.60. Test case 9 input values

Input (angular velocity)	Case 5	Case 6
Function 1	$-\omega_{f1,max}$	$-\omega_{f1,max}/2$
Function 2	$\omega_{f2,max}$	$-\omega_{f2,max}$
Function 3	$V_{f3,max}$	$-V_{f3,max}$

Table C.61. Test case 9 initial state

State (position)	Case a
Function 1	N/A
Function 2	N/A
Function 3	N/A

Table C.62. Test case 9 expected outcome

State/input	Case 1	Case 2
Case a	$p_{1l,m} = p_{1l,r} \pm 10\%$	$p_{1l,m} = p_{1l,r} \pm 10\%$

Table C.63. Test case 9 expected outcome

State/input	Case 3	Case 4
Case a	$p_{1l,m} = p_{1l,r} \pm 10\%$	$p_{1l,m} = p_{1l,r} \pm 10\%$

Table C.64. Test case 9 expected outcome

State/input	Case 5	Case 6
Case a	$p_{1l,m} = p_{1l,r} \pm 10\%$	$p_{1l,m} = p_{1l,r} \pm 10\%$

10. The lower side pressure for the second boom during steady state shall be within a 10% margin of the real system in all cases during 90% of the time.

Table C.65. Test case 10 input values

Input (angular velocity)	Case 1	Case 2
Function 1	0	0
Function 2	$\omega_{f2,max}$	$-\omega_{f2,max}$
Function 3	0	0

Table C.66. Test case 10 input values

Input (angular velocity)	Case 3	Case 4
Function 1	$V_{f1,max}$	$-\omega_{f1,max}$
Function 2	$\omega_{f2,max}$	$\omega_{f2,max}/2$
Function 3	$V_{f3,max}$	$-V_{f3,max}$

Table C.67. Test case 10 input values

Input (angular velocity)	Case 5	Case 6
Function 1	$\omega_{f1,max}$	$-\omega_{f1,max}$
Function 2	$-\omega_{f2,max}$	$-\omega_{f2,max}/2$
Function 3	$V_{f3,max}$	$-V_{f3,max}$

Table C.68. Test case 10 initial state

State (position)	Case a
Function 1	N/A
Function 2	N/A
Function 3	N/A

Table C.69. Test case 10 expected outcome

State/input	Case 1	Case 2
Case a	$p_{2l,m} = p_{2l,r} \pm 10\%$	$p_{2l,m} = p_{2l,r} \pm 10\%$

Table C.70. Test case 10 expected outcome

State/input	Case 3	Case 4
Case a	$p_{2l,m} = p_{2l,r} \pm 10\%$	$p_{2l,m} = p_{2l,r} \pm 10\%$

Table C.71. Test case 10 expected outcome

State/input	Case 5	Case 6
Case a	$p_{2l,m} = p_{2l,r} \pm 10\%$	$p_{2l,m} = p_{2l,r} \pm 10\%$

Appendix D Stakeholder requirements

This project aims to reduce load interference without changing hardware components such as the hydraulic valve or pump; instead it will be done by adding a controller platform to the crane system and implementing a control structure between the user and the hydraulic valve. At the beginning of this project, stakeholder requirements were created in collaboration with Hiab as:

1. Create a linear relationship between users' input and crane function velocity.
2. Ensure that different crane functions move at the same velocity for the same input.
3. Make a tip point controller to enhance manoeuvrability for the user.

These specifications have been analysed and the following requirements were extracted.

D.1 Design and verify a model of the crane

1. A model describing the dynamics of the system with enough precision so that a controller can be developed and control the crane.
   Rationale: The model should contain a detailed description of both the hydraulic system and the mechanical system, using a XS 022 crane from Hiab as a reference. The model may disregard dynamics that are not necessary for developing a controller.
2. The model should have three input signals in order to control the spool corresponding to each function and should accurately simulate the dynamics of the first boom, second boom and the extension boom as well as the pressures in the hydraulic system.
   Rationale: As the slew is disconnected for safety reasons, the functions only consists of moving the first boom, the second boom and the extension boom in a positive or negative direction.
3. The model shall be verified with the crane XS 022 and the hydraulic valve Parker P70 currently located in at KTH

D.2 Develop a control system that manages interference in the system

1. Regardless of disturbances the crane shall follow first boom reference signal provided by the operator.
   Rationale: Disturbance signals are for example varying loads or actuations of different cylinders simultaneously. The reference signal is the wanted cylinder velocity of the first boom
2. Regardless of disturbances the crane shall follow second boom reference signal provided by the operator.
   Rationale: Disturbance signals are for example varying loads or actuations of different cylinders simultaneously. The reference signal is the wanted cylinder velocity of the second boom.
3. Regardless of disturbances the crane shall follow extension boom reference signal provided by the operator.
   Rationale: Disturbance signals are for example varying loads or actuations of different cylinders simultaneously. The reference signal cylinder velocity.

4. The control system should use as few sensors as possible.
   Rationale: The purpose of switching to open center valve is to reduce the cost to the costumer so additional sensors mean more cost for the costumer customer.

5. The control system should be able to perform when using the sensors currently installed in the crane.
   Rationale: Sensors installed are spool position sensors, pressure sensors, two wire encoders, a laser distance sensor and two inclinometers.

D.3 Develop a tip point controller for the crane

1. The controller shall be able to move the tip of the crane at a constant velocity in a horizontal motion by the combined usage of the two booms and extension.

2. The controller shall be able to move the tip of the crane at a constant velocity in a vertically vertical motion by the combined usage of the two booms and extension.

Appendix E System requirements

By analyzing the stakeholder requirements given by Hiab, it was possible to derive a great deal of detailed and measurable system requirements. These will be used to verify that the output of this project is suitable. It is important to mention that these requirements are preliminary, and the values used are just estimates. More accurate values will be determined during the course of the project.

E.1 System requirements for crane model

1. The simulated steady state angular velocity of the first boom shall be within a 10% margin of the real system during 90% of the time.
   Rationale: The model only needs to follow the real system most of the time as high frequency spikes may occur.

2. The simulated steady state angular velocity of the second boom shall be within a 10% margin of the real system during 90% of the time.
   Rationale: The model only needs to follow the real system most of the time as high frequency spikes may occur.

3. The simulated steady state velocity of the extension boom shall be within a 10% margin of the real system during 90% of the time.
   Rationale: The model only needs to follow the real system most of the time as high frequency spikes may occur.

4. The rise time of the model with regard to the angular velocity of first boom shall be within a 10% margin of the real system.

5. The rise time of the model with regard to the angular velocity of second boom be within a 10%margin of the real system.

6. The rise time of the model with regard to the velocity of extension boom shall be within a 10% margin of the real system.

7. The gallery pressure during steady state shall be within a 10% margin of the real system in all cases during 90% of the time.

8. The rod side pressure for the first boom during steady state shall be within a 10% margin of the real system in all cases during 90% of the time.
   Rationale: This applies to a system without an external load.

9. The piston side pressure for the first boom during steady state shall be within a 10% margin of the real system in all cases during 90% of the time.
   Rationale: This applies to a system without an external load.

10. The piston side pressure for the second boom during steady state shall be within a 10% margin of the real system in all cases during 90% of the time.
    Rationale: This applies to a system without an external load.

E.2 System requirements for HW/SW

1. The controlled system for interference shall run on Controller (IFM CR7032).
2. The controller process shall perform in real time.
3. The controller code shall be written in a programming language with CODESYS support.
4. The controller code shall not interfere with the built in safety functionality.
5. The output delay when changing input shall not exceed 250 ms.

E.3 System requirements for crane safety

1. If the remote control emergency button is pressed, all movement shall cease within 1 second
2. All control aspects of the crane shall be turned off from the time that the emergency button is pressed until it is reset.
3. Once the emergency button is reset, the system shall be run in open loop from the interface and ignore all control actions.
   Rationale: if the system becomes unstable, it can be hazardous if it is reset.
4. Each person who intends to run the crane shall read through, sign and follow the safety document Safety Routine for running the crane.

E.4 System requirements for interference control

1. The control system shall be stable for all reference input values on either port.
2. The steady state output angular velocity of the first boom shall be within a 10% margin of the scaled reference value during 90% of the time when operating independently of all other functions.
   Rationale: The scaled reference value is used in order to not exceed the maximum capacity of the crane.
3. The steady state output angular velocity of the second boom shall be within a 10% margin of the scaled reference value during 90% of the time when operating independently of all other functions.
   Rationale: The scaled reference value is used in order to not exceed the maximum capacity of the crane.
4. The steady state output velocity of the extension boom shall be within a 10%margin of the scaled reference value during 90% of the time when operating independently of all other functions.
   Rationale: The scaled reference value is used in order to not exceed the maximum capacity of the crane.
5. The steady state output angular velocity of the first boom shall be within a 15% margin of the scaled reference value during 90% of the time when operating alongside other crane functions.
   Rationale: The scaled reference value is used in order to not exceed the maximum capacity of the crane.
6. The steady state output angular velocity of the second boom shall be within a 15% margin of the scaled reference value during 90% of the time when operating alongside other functions.
   Rationale: The scaled reference value is used in order to not exceed the maximum capacity of the crane.
7. The steady state output velocity of the extension boom shall be within a 15% margin of the scaled reference value during 90% of the time when operating alongside other functions.
   Rationale: The scaled reference value is used in order to not exceed the maximum capacity of the crane.
8. The rise time of the angular velocity for a step response from rest to scaled maximum velocity shall be less than 1 second when operating the first boom independently of all other functions.
   Rationale: independently means that all other boom functions are at rest.

9. The rise time of the angular velocity for a step response from rest to maximum velocity shall be less than 1 second when operating the second boom independently of all other functions.
   Rationale: independently means that all other boom functions are at rest.

10. The rise time of the velocity for a step response from rest to maximum velocity shall be less than 1 second when operating the extension boom independently of all other functions.
    Rationale: independently means that all other boom functions are at rest.

11. The overshoot for a step response shall for step inputs be less than 15% for all input values for the first boom.

12. The overshoot for a step response shall for step inputs be less than 15% for all input values for the second boom.

13. The overshoot for a step response shall for step inputs be less than 15% for all input values for the extension boom.

14. The control system shall be able to withstand a load varying from 0 to 250 kg.

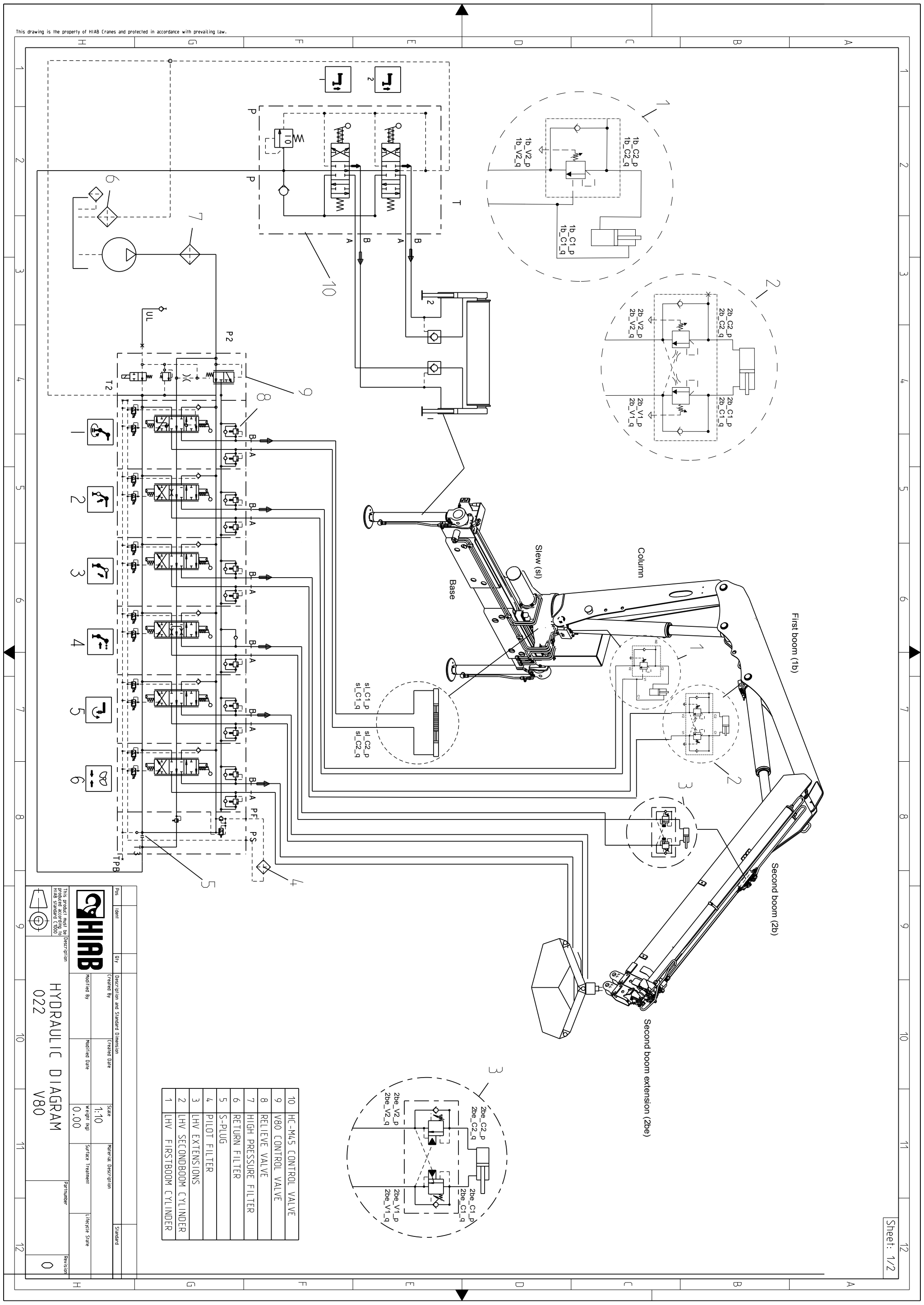
E.5 System requirements for tip point control

1. The tip shall be able to move for a certain distance in the horizontal plane without deviating more than 10 cm from said plane.

2. The tip shall be able to move for a certain distance in the vertical plane without deviating more than 10 cm from said plane.

3. The speed of the tip shall not deviate more than 10% from the scaled reference value

Appendix F Hydraulic diagram XS 022



Appendix G Test of counterbalance valves

G.1 Introduction

Right now in our model of the Hiab XS 022, we have assumed that the counterbalance valves have a linear relationship between the pressure V1 and the opening area (until fully open) through the counterbalance valve. This area, and the linear relationship is just an initial guess and measurements is needed in order to investigate if this assumption is valid and what the opening area is under different pressures V1. An additional test in order to see how much impact different flows have on the relationship is also needed.

Our suggestion is to measure the flow through the counterbalance valves, the pressures $C2$ and $V2$ and the pilot pressure $V1$ (see fig. G.1) in order to investigate the relationship. The equation for the relationship between pressure and flow can be described as

$$C_v \dot{p} = q \tag{G.1}$$

Where C_v is the fluid capacitance, $\dot{p}$ is the pressure derivative and q the flow. Using the equation above and using the terminology from a standard counterbalance valve the equation can be written as:

$$C_v \dot{p}_{c2} = q = C_d \sqrt{\frac{2}{\rho} p_{c2} - p_{v2}} A_0(p_{v1}, p_{c2}, q) \tag{G.2}$$

C_d is a discharge constant and ρ is the density of oil. Measurements for flow q and the different pressures are needed for an interval where the counterbalance valve is fully closed to fully open. A test where the flow is varied and the pressures are kept constant in order to investigate the impact of flow is also crucial.

G.2 First boom counterbalance valve (395-8078)

The schematic for the first boom counterbalance valve is seen in fig. G.1. The purpose of the test is to check how different pressures and flows impact the opening area (or the pressure drop) over the valve.

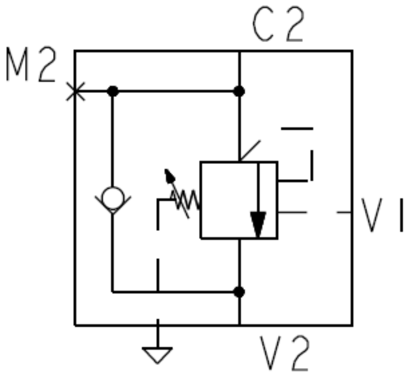


Figure G.1. Schematic for the counterbalance valve for the first boom

G.2.1 Test 1a: Constant flow, variable pressure

Input and Initial conditions: A flow q and a pressure $C2$ which do not meet the conditions to open the valve until the pressure $V1$ becomes large enough. Then incrementally increase $V1$ until $V2$ becomes constant.

Table G.1. Inputs to test 1a

Variable	Input	Notes
q	10-20 (l/m)	Constant
$C2$	$50bar$	Constant
$V1$	$0 - 120bar$	Increase with 5 bar when steady state pressure is measured at $V2$, end test when $V2$ becomes constant
$V2$	-	Only measured, end when it becomes constant

Output: .mat file, .txt file or .xls with following format:

Table G.3. Outputs from test 1a

q(l/min)	$C2$(bar)	$V1$(bar)	$V2$(bar)	time (s)
q_1	$C2_1$	$V1_1$	$V2_1$	t_1
...	...	...	...	...
q_n	$C2_n$	$V1_n$	$V2_n$	t_n

Expected outcome: A relationship between the pressure $V2$ and $V1$, where the pressure $V2$ should be zero until $V1$ is large enough to open the counterbalance valve. The pressure $C2$ also have an impact for when the valve opens but since its constant the pilot ratio is known, its impact can be calculated.

G.2.2 Test 1b: Constant pressure, variable flow

Input and Initial conditions: A pressure $V1$ and a pressure $C2$ which do not meet the conditions to open the valve until the flow q becomes large enough. Then incrementally increase q until $V2$ becomes constant independent of increased flow.

Table G.5. Inputs to test 1b

Variable	Input	Notes
q	0-35 (l/m)	Increase with 1 l/min when steady state pressure is measured at $V2$. End test when pressure $V2$ becomes constant
$C2$	$50bar$	Constant
$V1$	$50bar$	Constant
$V2$	-	Only measured, end when it becomes constant

Output: .mat file, .txt file or .xls with following format:

Table G.7. Outputs from test 1b

q(l/min)	$C2$(bar)	$V1$(bar)	$V2$(bar)	time (s)
q_1	$C2_1$	$V1_1$	$V2_1$	t_1
...	...	...	...	...
q_n	$C2_n$	$V1_n$	$V2_n$	t_n

Expected outcome: A relationship between the pressure $V2$ and the flow q, where the pressure $v2$ should be zero until the flow q is large enough to open to counterbalance valve. The pressures $C2$ and $V1$ also have an impact on which flow opens the valve but since they are constant their impact can be calculated from test 1a.

G.3 Second boom counterbalance valve (396-2768)

The schematic for the second boom counterbalance valve is seen in fig. G.2. The purpose of the test is to check how different pressures and flows impact the opening area (or the pressure drop) over the valve.

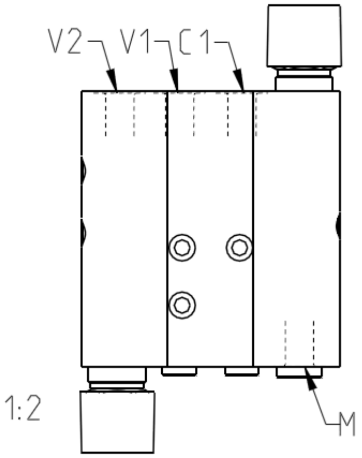


Figure G.2. Schematic for the counterbalance valve for the second boom

G.3.1 Test 2a: Constant flow, variable pressure

Input and Initial conditions: A flow q and a pressure $C2$ which do not meet the conditions to open the valve until the pressure $V1$ becomes large enough. Then incrementally increase $V1$ until $V2$ becomes constant.

Table G.9. Inputs to test 2a

Variable	Input	Notes
q	10-20 (l/m)	Constant
$C2$	$50bar$	Constant
$V1$	$0 - 120bar$	Increase with 5 bar when steady state pressure is measured at $V2$, end test when $V2$ becomes constant
$V2$	-	Only measured, end when it becomes constant

Output: .mat file, .txt file or .xls with following format:

Table G.11. Outputs from test 2a

q(l/min)	$C2$(bar)	$V1$(bar)	$V2$(bar)	time (s)
q_1	$C2_1$	$V1_1$	$V2_1$	t_1
...	...	...	...	...
q_n	$C2_n$	$V1_n$	$V2_n$	t_n

Expected outcome: A relationship between the pressure $V2$ and $V1$, where the pressure $V2$ should be zero until $V1$ is large enough to open the counterbalance valve. The pressure $C2$ also have an impact for when the valve opens but since its constant the pilot ratio is known, its impact can be calculated.

G.3.2 Test 2b: Constant pressure, variable flow

Input and Initial conditions: A pressure $V1$ and a pressure $C2$ which do not meet the conditions to open the valve until the flow q becomes large enough. Then incrementally increase q until $V2$ becomes constant independent of increased flow.

Table G.13. Inputs to test 1b

Variable	Input	Notes
q	0-35 (l/m)	Increase with 1 l/min when steady state pressure is measured at $V2$. End test when pressure $V2$ becomes constant
$C2$	$50bar$	Constant
$V1$	$50bar$	Constant
$V2$	-	Only measured, end when it becomes constant

Output: .mat file, .txt file or .xls with following format:

Table G.15. Outputs from test 1b

q(l/min)	$C2$(bar)	$V1$(bar)	$V2$(bar)	time (s)
q_1	$C2_1$	$V1_1$	$V2_1$	t_1
...	...	...	...	...
q_n	$C2_n$	$V1_n$	$V2_n$	t_n

Expected outcome: A relationship between the pressure $V2$ and the flow q, where the pressure $v2$ should be zero until the flow q is large enough to open to counterbalance valve. The pressures $C2$ and $V1$ also have an impact on which flow opens the valve but since they are constant their impact can be calculated from test 2a.

G.4 Extension boom counterbalance valve (193-7171)

The schematic for the second boom counterbalance valve is seen in fig. G.3. The purpose of the test is to check how different pressures and flows impact the opening area (or the pressure drop) over the valve. The extension contains a dual counterbalance valve but in a initial test only one direction is measured.

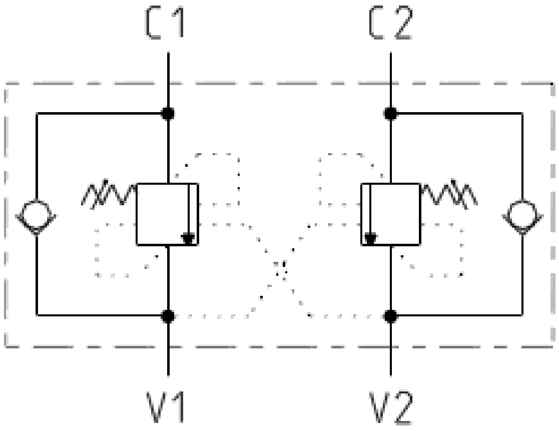


Figure G.3. Schematic for the counterbalance valve for the extension boom

G.4.1 Test 3a: Constant flow, variable pressure

Input and Initial conditions: A flow q and a pressure $C2$ which do not meet the conditions to open the valve until the pressure $V1$ becomes large enough. Then incrementally increase $V1$ until $V2$ becomes constant.

Table G.17. Inputs to test 3a

Variable	Input	Notes
q	10-20 (l/m)	Constant
$C2$	$50bar$	Constant
$V1$	$0 - 120bar$	Increase with 5 bar when steady state pressure is measured at $V2$, end test when $V2$ becomes constant
$V2$	-	Only measured, end when it becomes constant

Output: .mat file, .txt file or .xls with following format:

Table G.19. Outputs from test 3a

q(l/min)	$C2$(bar)	$V1$(bar)	$V2$(bar)	time (s)
q_1	$C2_1$	$V1_1$	$V2_1$	t_1
...	...	...	...	...
q_n	$C2_n$	$V1_n$	$V2_n$	t_n

Expected outcome: A relationship between the pressure $V2$ and $V1$, where the pressure $V2$ should be zero until $V1$ is large enough to open the counterbalance valve. The pressure $C2$ also have an impact for when the valve opens but since its constant the pilot ratio is known, its impact can be calculated.

G.4.2 Test 3b: Constant pressure, variable flow

Input and Initial conditions: A pressure $V1$ and a pressure $C2$ which do not meet the conditions to open the valve until the flow q becomes large enough. Then incrementally increase q until $V2$ becomes constant independent of increased flow.

Table G.21. Inputs to test 1b

Variable	Input	Notes
q	0-35 (l/m)	Increase with 1 l/min when steady state pressure is measured at $V2$. End test when pressure $V2$ becomes constant
$C2$	$50bar$	Constant
$V1$	$50bar$	Constant
$V2$	-	Only measured, end when it becomes constant

Output: .mat file, .txt file or .xls with following format:

Table G.23. Outputs from test 1b

q(l/min)	$C2$(bar)	$V1$(bar)	$V2$(bar)	time (s)
q_1	$C2_1$	$V1_1$	$V2_1$	t_1
...	...	...	...	...
q_n	$C2_n$	$V1_n$	$V2_n$	t_n

Expected outcome: A relationship between the pressure $V2$ and the flow q, where the pressure $v2$ should be zero until the flow q is large enough to open to counterbalance valve. The pressures $C2$ and $V1$ also have an impact on which flow opens the valve but since they are constant their impact can be calculated from test 3a.

Appendix H Preliminary test cases

This appendix concerns the test cases that were used to preliminarily test the model, as well as base the velocity controller on. Below, a number of tables are presented which contain test length in seconds, input reference in terms of a percentage value for spool current (and therefore displacement), as well as initial positions. There are no expected results presented as these test cases are not designed to verify the model or controller, but used for data collection and modeling. It should be mentioned that these test cases are based on those defined for controller verification in Appendix C, hence the disjoint numbering below.

The first set of test cases regards the first boom and attempts to cover the entire range of step inputs, as seen in table H.1. The aim is to reach the steady state velocity with each of the tests.

Table H.1. Test case 1 parameter values

Case	A	B	C	D	E	F	G	H	I	J	K
Length [s]	3	3	3	3	3	3	3	3	3	3	3
Input reference (spool action) [%]											
1st boom	100	80	60	40	20	0	-20	-40	-60	-80	-100
2nd boom	0	0	0	0	0	0	0	0	0	0	0
Extension	0	0	0	0	0	0	0	0	0	0	0
Cylinder initial position [mm]											
1st boom	200	200	200	200	200	200	560	560	560	560	560
2nd boom	360	360	360	360	360	360	360	360	360	360	360
Extension	10	10	10	10	10	10	10	10	10	10	10

Similarly, the second set of test cases regards the second boom and attempts to cover the entire range of step inputs for that, as seen in table H.2.

Table H.2. Test case 2 parameter values

Case	A	B	C	D	E	F	G	H	I	J	K
Length [s]	3	5	5	5	5	5	5	5	5	5	3
Input reference (spool action) [%]											
1st boom	0	0	0	0	0	0	0	0	0	0	0
2nd boom	100	80	60	40	20	0	-20	-40	-60	-80	-100
Extension	0	0	0	0	0	0	0	0	0	0	0

Cylinder initial position [mm]											
1st boom	350	350	350	350	350	350	350	350	350	350	350
2nd boom	20	20	20	20	20	20	600	600	600	600	600
Extension	10	10	10	10	10	10	10	10	10	10	10

Subsequently, the final test case set for single step inputs, test case 3, regards the second boom extension. These parameter values are presented in table H.3.

Table H.3. Test case 3 parameter values

Case	A	B	C	D	E	F	G	H	I	J	K
Length [s]	2	3	5	5	5	5	5	5	5	3	3
Input reference (spool action) [%]											
1st boom	0	0	0	0	0	0	0	0	0	0	0
2nd boom	0	0	0	0	0	0	0	0	0	0	0
Extensino	100	80	60	40	20	0	-20	-40	-60	-80	-100
Cylinder initial position [mm]											
1st boom	350	350	350	350	350	350	350	350	350	350	350
2nd boom	320	320	320	320	320	320	320	320	320	320	320
Extension	1300	1300	1300	1300	1300	1300	20	20	20	20	20

The previous test cases concern single input, single output behavior and is necessary to get a deep understanding in each individual function. The test case presented in table H.4 excites several functions simultaneously using step inputs, to investigate the multiple input, multiple output-behavior of the system.

Table H.4. Test case 7 parameter values

Case	A	B	C	D	E	F	G	H
Length [s]	2	2	2	2	2	2	2	2
Input reference (spool action) [%]								
1st boom	100	100	0	-100	-100	0	100	-100
2nd boom	100	0	100	-100	0	-100	100	-100
Extension	0	-100	-100	0	100	100	-100	100
Cylinder initial position [mm]								
1st boom	400	400	400	500	500	500	400	500

2nd boom	250	250	250	450	450	450	250	450
Extension	20	20	20	800	800	800	20	800

Finally, another type of inputs are considered. In test case 8, presented in table H.5 below, the input signals are ramps instead. The notation for the input being where the starting value and stopping value are presented as a range. The ramp is performed evenly over the test length.

Table H.5. Test case 8 parameter values

Case	A	B	C	D
Length [s]	2	2	2	2
Input reference (spool action) [%]				
1st boom	0–100	0–(-100)	0–100	0–(-100)
2nd boom	0–100	0–(-100)	0–100	0–(-100)
Extension	0–(-100)	0–100	0–(-100)	0–100
Cylinder initial position [mm]				
1st boom	400	500	400	500
2nd boom	250	450	250	450
Extension	20	800	20	800