HAN_UNIVERSITY OF APPLIED SCIENCES

PROJECT LOCK & KEY

DESIGN AND CONTROL OF AN INNOVATIVE LOCKING MECHANISM

Final Report

HAN_UNIVERSITY
OF APPLIED SCIENCES

Master Engineering Systems – Minor project

June 2020, Arnhem
The Netherlands

Group 7C
Version 1.0

HAN_UNIVERSITY OF APPLIED SCIENCES

PROJECTLOCK&KEY

FINAL REPORT CONTAINING PROJECT DESCRIPTION, RESULTS AND CONCLUSIONS

Project member	Function	Number	Contact details
Mr. Emiel Gerrits	*Project leader*	*548518*	*E.Gerrits@student.han.nl*
Mr. Toms Alpa-Luks	*Communicator*	*641473*	*T.AlpaLuks@student.han.nl*
Mr. Manish Raathimiddi	*Project member*	*644781*	*MV.Raathimiddi@student.han.nl*
Mr. Lakshan Raaj Karthikeyan	*Project member*	*634747*	*LR.Karthikeyan@student.han.nl*

Company	**HAN – University of Applied Sciences**
Supervisor	**Mr. N. Bouwman**
Date	**June 2020**

Executive Summary

This project is a part of the theoretical phase taught at the HAN University of Applied Sciences. In more detail, the objective of this project is to design a controllable servo-motor system which would accurately account for the desired placement of a key within a lock.

The project was launched by describing the desired motion profile of the key. Further, the main objective was stated together with some sub-objectives. The main objective of this project is to achieve a fully controllable lock and key system. Hence, this goal was achieved by:

- Firstly, deriving and describing the system with the help of mathematical differential equations. Later on, the system was modelled within the SIMULINK software;
- Secondly, the developed model in SIMULINK had to be validated. Thus, this was done by obtaining measurements from the real lock and key system, and comparing them to the modelled system;
- Thirdly, two different controller design methods were used for developing compensators for the system: SIMULINK PID autotune and the Root-Locus technique. The results showed that the root-locus method is a feasible one, however, it is of great importance to correctly account for the limitations of the system by using this approach. On the other hand, the autotune approach is a faster and more robust design method as it provides the user with an instant insight into the systems response;
- Finally, the resulting system responses of the controlled system were presented, and the results were critically looked at.

At the final phase of this project it was concluded that the main objective was, indeed, achieved. Moreover, after comparing the measurements of the real system to the developed model, it was possible to state that the modelled system is a good representation of the real system.

For further study applicable for this project, a velocity controller is purposed for a ‘full’ control of the system. Another recommendation is to investigate and implement the unmodelled systems’ disturbances within the made model. Hence, this would allow to design an idealistic feedforward controller that would efficiently react to changes in the setpoint value.

CONTENTS

1. INTRODUCTION

BACKGROUND

It is a well-known fact that control engineering has contributed enormously to the development of the present industrial society. Furthermore, in the automation systems of today, control engineering remains as one of the most important aspects (Linzenkirchner, 1999). So, it is fear to say that automation and control play a very important role in most industries. It also brings quite a few advantages over uncontrolled industrial processes, such as:

- Higher productivity;
- Better quality;
- Flexibility;
- Higher accuracy of information;
- Safety and security.

By acknowledging that controlled processes can be found everywhere, it is of great importance to learn basic principles that are working behind them. For this reason, HAN University of Applied Sciences has assigned a 'Lock and Key' project related to systems' control. Here, the project team is going to go through the process of designing a controlled process. In more detail, a model will be designed for condoling the position of a moveable key in a fixed lock. The controllability is based on two servo motors, one establishing a linear movement of the key and the other applying rotational movement around the key's centre axis.

PROBLEM DEFINITION

In this project, the main objective is to design a controller in order to change an uncontrolled process to a controlled process. This can be done by completely understanding the working of the lock and key mechanism and by modeling and simulating it in MATLAB/Simulink. Due to designing a suitable controller, the lock and key mechanism must be able to control the position of the key and to unlock the lock and to overcome the overshoot caused by the velocity motor of the key. Thus the problem definition is as followed:

"The standard lock-and-key system exists out of an uncontrolled positioning process and therfore it's function is to rectify the overshoot of the velocity motor and to make it into a controlled process."

Figure 1. Graphical representation of a gear rack and a servo motor
Source: Maxon Motors

Goal(s) for control	➢ Control the position and velocity of the two servo motors in such way that system overshoot is avoided or minimalized
Process output(s) *» with sensor(s)*	➢ Output current [A] towards the servomotors ➢ Rotational displacement [rad or degrees] and speed [rad/sec] of both servomotors. - Encoder [pulses/revolution]
Process input(s) *» with actuator(s)*	➢ Input voltage [V] to both servo motors through pulse-width-modulation (PWM).
Disturbance(s)	➢ (Dynamic) friction between components and/or masses. ➢ Load torque

Table 1. Process definition

A more concise process definition is given in Table 1(above). First, the goal for control is introduced. This goal, which is similar to the project objective, states that the desired system must have a minimal or no overshoot for the rotational movement of the key. Thus, this is related to the fact that an overshoot could lead to not fitting the key in the lock. Secondly, another factor to consider is the quickness of the translational motion with respect to the rotational movement. From the motion profile (see below), it is clear that the system response with an overshoot will lead to the issue related to the fact that the rotational movement is not finished before the key enters the lock.

In order to understand the desired system response, a 3-step motion profile is set up:

STEP 1 – CHOOSE A HOMING POSITION

Choosing a homing location of the system allows to set the origin in a consistent and repeatable fashion. Hence, this would give the advantage of knowing the initial position of the key with respect to the location of the lock.

STEP 2 – SYSTEM RESPONSE

The key rotates and translates towards the lock at the same time. However, the rotational motion of the key has ended before the key enters the lock. This results in translational motion by x1 [mm] and a complete rotational motion of the key by Θ_{m1} [rad or degrees]. As the rotational motion should be finished before key enters the lock, the translational motion is finalized at x2 [mm]. This results in the total translational and rotational distances of the key by X_m [mm] = x1 + x2 and Θ_m [rad or degrees], respectively.

STEP 3 – UNLOCK AND LOCK

When the key enters the lock (X_m is reached), the key rotates by Θ_{m2} [rad or degrees] to open the lock. Next, the key rotates back by - Θ_{m2} [rad or degrees] to lock the lock. Finally, the key moves back to the reference homing position at x0 [mm] = 0.

PROJECT OBJECTIVES

Every project has a cause for starting up. Some of them are based on innovation and/or development, others on problem solving. Besides there two, execution of research projects, mainly to obtain specific information about or related to a certain subject, also occurs. There is one thing all projects have in common, namely a project (main) objective, a formulation that describes the purpose of the project and/or final result. The main objective for this project is described below:

"Design a controller in order to realize accurate positioning and rotating of a lock-and-key system by using two servomotors."

The main objective states the final outcome of the project. However, to present a controlled lock and key system the project team has defined sub-objectives which are of great importance for achieving the main objective:

- Deal with an expected overshoot related to the slide motor;
- Get acquainted with the set-up of lock and key system and general controlling methods of a servosystem. In addition, the members also need to get acquainted with validation methods in order to achieve the desired position and meet the stated requirements;
- Develop a mathematical model of the system by using MATLAB/Simulink;
- Verify the modelled system with available literature and measurements from the physical system;
- Design a controller and either implement it in the EPOS2 controller board (present in the physical system) or simulate other controller configurations with MATBAL/Simulink software.

OUTLINE OF THE MINOR PROJECT REPORT

This report is divided into five sections such as, introduction, methods, results, discussion and conclusions. Initially, some background information is gathered in order to start the introduction Then the problem, which will be encountered in the project is defined in the problem definition and the objectives which are used to solve the problem. In methods, the data flow diagram is proposed based on the system, then the mathematical equations required to model the systems is derived. In order to implement the controller, the Root locus technique is used to control the system.

The results section contains the overall transfer function of the model and the validation of the model with EPOS2 system. It also contains the graphs and gain values obtained from the Root locus technique. In the discussion chapter, the problems faced by the controllers and to achieve the goal are briefly discussed. Finally, the objectives which are achieved are mentioned in the conclusion chapter and the ways to improve the current models are discussed in the recommendations chapter.

2. LITERATURE SURVEY

Servo motor is a part of closed loop system. It consists of several parts like control circuit, DC motor, encoders etc. It creates two types of mechanical motion one is rotary and the other is linear. Servo motors consists of two major components: electrical and mechanical. Through feedback sensor loop servo motor drives the motor to the desired output from the received control signal. To calculate the generated serial output signal from the motor in the lock and key set up an incremental encoder has used. Based on the desired output various incremental encoders can be used.

There are wide range of coding been used for servo motor to tune the motor for precisely controlling. To improve the transient response and system response there are plethora of controllers are in handy to use. In this project a proportional-integral-derivative (PID) controller has been used. PID reduces the time constant thus making the system to respond fast. Moreover, PID auto tuning is the major advantage to obtain proper desired values.

The nature of graph trend obtained from the results, is a motion profile of a servomotor for position, velocity and for acceleration. Validating the model based on the resultant graphs lead us to the proper conclusion that whether obtained values are realistic or not. From this literature survey transfer function has been derived and concludes result produced were meaningful. In Appendix B Literature survey, an in depth literature has been explained how literature has helped us to achieve in this project.

3. METHODS

The following chapter provides information on the methods used within this project. Moreover, the project was approached in the exact same order as described below.

PROCESS DATA FLOW DIAGRAM (DFD)

For a precise process definition, it is of great importance to account for all the inputs and outputs. Hence, a data flow diagram is developed. This particular method is used due to its simplicity and overview ability. Furthermore, the DFD is a great illustration of all the system components and their relation to one-another. Essentially, a proper DFD maps out the flow of information for any process or system.

MATHEMATICAL DESCRIPTION OF THE SYSTEM

After designing a detailed DFD, the system had to be described with commonly used differential equations. As the Lock & Key system consists of two servomotors which are both attached to the gearboxes, each motors electrical and mechanical parts were derived. Thus, the following equations were derived:

- Electrical part
  - Equation describing the electrical circuit relation to the motor shafts' velocity.
- Mechanical part
  - Equation representing the torque balance of the system and its connection to motor current.

In more detail, electrical part of the servo system is very closely related to its mechanical part. This is due to the fact that the torque produced by the motor is equal to the motors' produced current multiplied by the torque constant. Thus, the equations can be directly related to one-another. Later on, the derived differential equations were used to obtain the systems transfer function, which is needed for designing a controller for the model. In addition to the hand-calculated controller design, a similar controller is designed using tuning functions that are available in Matlab Simulink. Later on, both controllers are compared on output behavior, performance and simplicity.

The information required for the mathematical description of the system was mostly taught within the Applied Physics course at HAN University. Additionally, available online sources regarding servo-mechanism modelling were used. This was possible since servo-motor models are widely modelled within the control industry.

SYSTEM MODELLING

The modeling approach used within this project can be compared to the modelling procedures applied within the industry. The modelling software used for this for project was MATLAB – SIMULINK, version 2019b. Within the software, a computer-based model of the Lock & Key system was developed.

First, the derived differential equations were modelled separately in sub-systems. Secondly, the input(s) and output(s) were related to one another. For instance, the output of the electrical part (current) was multiplied by the motor torque constant which is the produced motor torque. But the torque produced by the motor is actually an input to the mechanical part of the system. Finally, a MATLAB scrip was generated. This script included all the parameters values used in the SIMULINK model. Moreover, by running the scrip, the model is simulated, and the required results and plots are generated.

MODEL VALIDATION

In this project, validation of the designed model is performed by comparing real-system measurements to the ones from the model. This was done by reconstructing the existing EPOS2 controller into the model structure. Hence, a PI and PID controllers for current and position, respectively, were incorporated within the model. Furthermore, these two controller types are also described in the EPOS manual.

Next, by running the real system trough EPOS software, with the corresponding controllers and specified position setpoint, it was possible to obtain the autotuned system gains together with the system readings. Thus, by converting the EPOS gains to SI units and setting the same position setpoint in the modelled system, similar current and position responses were expected.

Validation of a modelled real system is of great importance prior to designing controllers. This is due to the fact that, if a controller is designed for an inaccurate representation of a real system, this controller will most likely fail to work if implemented on the real system. The particular method for validating the model was used as this is one of the most commonly known approaches to design a controller.

CONTROLLER DESIGN – ROOT LOCUS

As stated in Introduction chapter, a fully controlled Lock and Key system should be obtained at the end of this project. Moreover, to achieve this objective, a controller designing technique chosen for this purpose is the Root-Locus. This approach is known for its simplistic implementation and possibilities to account for plant uncertainties (Kostadin Kostov, 2008). Thus, this method is gaining more and more attention, especially in the modern control analysis (-, 2018). Additionally, this technique was studied at the HAN universities course Feedback Control. These are the main reasons why this method is utilized within this project.

It is well known that most industrial processes have a digital closed loop control system. Hence, the main algorithm applied at these processes is the Proportional Integral Derivative (PID) structure (Campo, 2012). Regards the controller structures, the Proportional Integral Derivative (PID) structure is used for the position control and the Proportional Integral (PI) structure is used for current controller. These structures are selected since the real systems EPOS2 controllers have the same formation.

To sum up, the design of a current and displacement controlled system with the help of root-locus technique is performed within this project. In more detail, a commonly used PID controller design procedure is applied, and this procedure consists of the following steps (Nise, 2010):

1. Evaluation of the uncompensated system – specification of the required parameters like settling time/overshoot/rise time;
2. Design of a PD controller – meeting the specifications (step 1) of transient response. The design includes finding the zero location and the loop gain;
3. Simulation – evaluate if the transient response requirements are met (if not - redesign);
4. Design of a PI controller – steady-state error improvement;
5. Determination of the Proportional, Integral and Derivative gain values – K_p, K_i and K_d;
6. Simulation – controlled system behaviour.

In case of a PI controller design, the design procedure is simpler than for PID. This is a result of the fact that the search for the zero location of the PD compensator is not needed.

4. RESULTS

This chapter provides the reader with the results. Firstly, the DFD of the system is presented. Next, the mathematical description of the system is explained in detail. Further, the derived mathematical model is simulated within the SIMULINK and some results are presented. Finally, the model is validated with the real-system measurements and the controllers are designed with the root-locus technique.

DATA FLOW DIAGRAM

The data flow diagram made by the authors is visible in Figure 2. The system consists of two motors which both have a voltage input. Furthermore, both motors are attached to a gearbox with a gearhead reduction 6.6:1. Both motor structures have equal outputs – motor armature current and the rotational displacement. However, the slide motor accounts for the translational motion of the key whereas the key motor is responsible for the rotational position of the key.

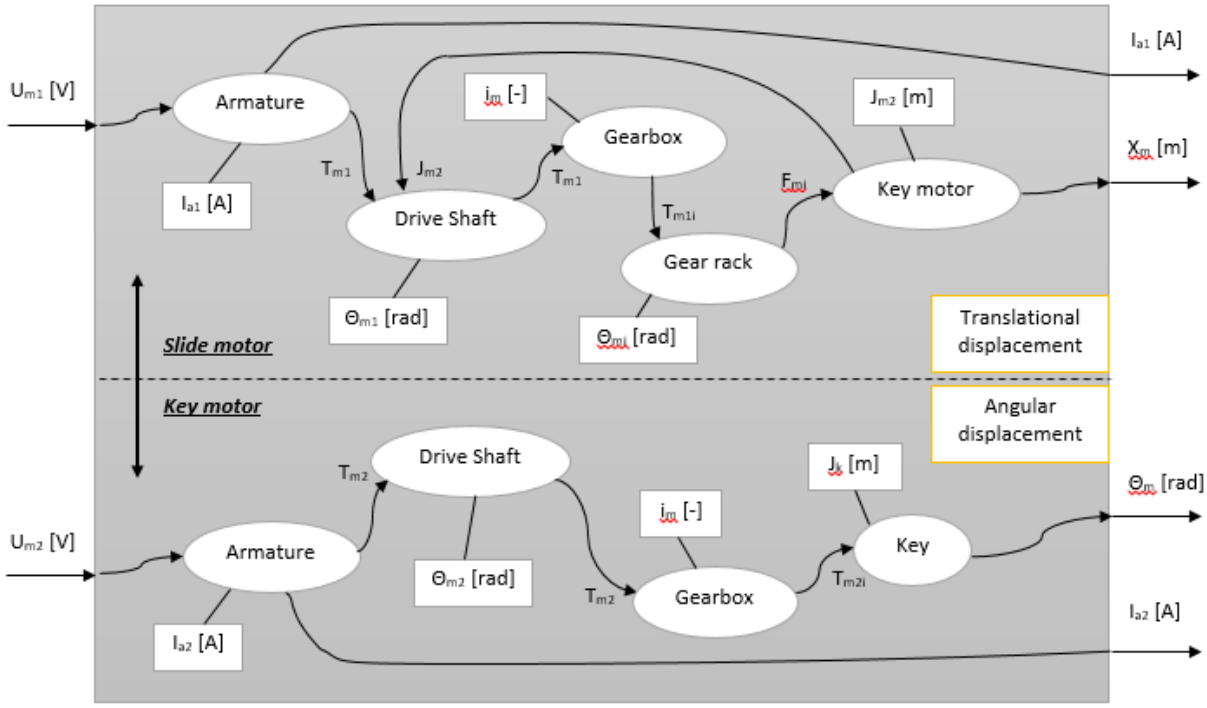

Figure 2. Data Flow Diagram of the Lock & Key system

The parameters visible in the DFD are:

- U_{m1} and U_{m2} [V] are controlled inputs for voltage supply towards the slide and key motors.
- I_{a1} and I_{a2} [A] are the current flows through the armature windings of motor 1 and 2, respectively.
- Θ_{m1} and Θ_{m2} [rad] indicates the angular displacement of each motor's 'no load' drive shaft.
- X_m [m] is the translational displacement of the key motor from its home position (zero reference). The limit switches are used in order to establish this zero reference or restore it after a disruption.
- I_m [-] represents the gear ratio of the planetary gearbox that is mounted on each servomotor.
- Θ_{mi} [rad] is the reduced angular displacement of the toothed shaft (gear), where the circumference of the gear and its angular displacement determine the translational displacement of the key motor

MATHEMATICAL REPRESENTATION OF THE SYSTEM

The following chapter describes the entire process of obtaining the final plant function that is used for designing suited controllers and executing simulations. Most of the functions, disturbances and other related subparts are mathematically described in order to be implemented. Certain parts are to be neglected or don't apply to this typical system nor the project objectives, it will be shortly mentioned.

MODEL OF THE DC-MOTOR

A typical input signal to a DC servomotor is the voltage. On the other side, the output in almost all cases is the angular position of the motor shaft. The motor presents a relation between current and torque. The latter one provides the spinning motion to the shaft and there is a relationship between this spinning and the back electromotive force. Moreover, other parameters of such system include shaft inertia, viscous friction (also known as damping), circuit resistance and inductance. Some of these parameter values are provided by a particular-motor manufacturer and, of course, these are only approximate values. Hence, it cannot be expected from the model to represent the exact same readings as from the physical system

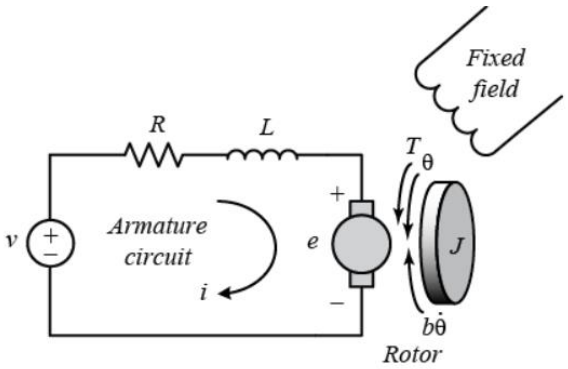

Figure 3. Electrical and mechanical component overview DC-motor
Source: Google Images

The DC servomotor is designed on the basics of the Lorentz power. When a charge moves through a magnetic field, a mechanical force is applied on the charge.

Charge over time is formulated as current and is transported over a current carrying conductor, that is in case of a DC motor armature winding out of a conductive material. The amount of charge determines the multiplier of the mechanical force that is applied. The amount of force that is applied on one particle of charge is directly described by the magnetic field strength that is surrounding that particle.

The amount of charge that is exposed to the magnetic field is depend of the current and the length of the conductor that crosses the magnetic field lines.

Therefore, the equation for Lorentz force is as followed:

$$F_L(t) = \beta * l * i_a(t) \quad (4.1)$$

With $\beta = Magnetic\ flux\ or\ field\ strength$

$l = Conductor\ length\ in\ magnetic\ field$

$i_a = Armature\ current$

ELECTRICAL PART OF THE DC-MOTOR

Application of voltage on the terminals of the armature will lead to a certain flow of current. This flow of current is limited by the internal resistance of the conductor and the armatures impedance. This resistance is a function of the voltage and current and is described with Ohm's law:

$$R_a(t) = \frac{V}{i_a(t)} \tag{4.2}$$

With $V = Voltage\ supply$

The interaction of the magnetic field within the armature windings produces an electromagnetic force or EMF, that by Faraday's law is known as the function of inductance. The voltage that is generated through the inductor is defined as:

$$\varepsilon_L = L_a \frac{di(t)}{dt} \tag{4.3}$$

With $L_a = Inductance\ of\ armature\ coil\ (constant)$

$$\frac{di}{dt} = Derivative\ of\ current\ over\ time$$

Force that is applied on the armature windings realizes torque that causes it to move around its axis. This torque is dependant of both the current and magnetic flux that is flowing through the armature. The magnetic flux is constant in a DC motor and therefore the torque can be written as a function of current:

$$T_m(t) = K_T * i_a(t) \tag{4.4}$$

With $K_T = Torque\ constant$

Due to this circular movement, the armature crosses magnetic field lines on different angles, generating a voltage on the terminal that is related to the number of turns. This phenomenon is called flux linkages and causes an EMF to flow in opposite direction of the armature current. The amount of back EMF induced by the armature is described as:

$$e_b = K_b * \frac{d\theta_m(t)}{dt} \tag{4.5}$$

With $K_b = Back\ EMF\ constant$

$$\frac{d\theta_m}{dt} = Derivative\ of\ angular\ displacement\ over\ time\ (velocity)$$

Stated by Kirchhoff's Voltage Law (KVL), the sum of all voltages in a circuit must be zero. The equivalent circuit of the total armature is expressed as:

$$V(t) = R_a i_a(t) + L_a \frac{di(t)}{dt} + K_b * \frac{d\theta_m(t)}{dt} \tag{4.6}$$

MECHANICAL PART OF THE DC-MOTOR

The mechanical part of the DC motor exists out of a rotor shaft and a (mass)load that is attached to it. The bearings in the housing will cause friction on the rotor shaft and thus behaves like a damper. The torque that is applied on the rotor shaft, due to the armature, is causing rotational motion.

This mass-damper system can be represented with the following equation:

$$T_m(t) - T_L = J_m * \frac{d^2\theta_m(t)}{dt^2} + B_m * \frac{d\theta_m(t)}{dt} \quad (4.7)$$

With $J_m = Total\ mass\ inertia\ of\ the\ motor$

$\frac{d^2\theta_m}{dt^2} = Second\ derivative\ of\ angular\ displacement\ over\ time\ (acceleration)$

$B_m = Damping\ coefficient\ or\ viscous\ friction\ constant\ of\ bearings$

T_L = *Load torque applied to the motor shaft*

The equations 1 and 2 are of significant importance for formulation of the transfer function. The first step exists out of substituting the electrical equation for the torque *T(I)* in the opposing mechanical equation. Due to the fact that both equations are describing exactly the same characteristic, thus only from a different perspective and domain, the bonding between the two domains is realized. Next, this final equation is rewritten in the standard format of a (open-loop) transfer function. To finalize the transfer function describing the DC-motor, a derivation of the Laplace transform is necessary in order to bend the physical domain from time (t) to frequency (S). The overall transfer function describing a (basic, velocity-controlled) DC-motor is:

$$G(s) = \frac{Output\ C(s)}{Input\ R(s)} = \frac{\omega(s)}{V(s)} = \frac{K_T}{L_aJ_mS^2+(R_aJ_m+L_aB_m)S+R_aB_m+K_bK_T} \quad (4.8)$$

To open and close the lock with the key, position is the variable that needs to be maintained. The system is final and operational if and only if the key is positioned in the right angle and is transported through the lock. Maintaining this value for position is critical for the fundaments of this system. By implementing a pure integrator, multiplication of the overall transfer function with the term $\frac{1}{s}$, the output of G(s) is transformed from velocity to position:

$$G(s) = \frac{Output\ C(s)}{Input\ R(s)} = \frac{\theta(s)}{V(s)} = \frac{K_T}{L_aJ_mS^3+(R_aJ_m+L_aB_m)S^2+R_aB_mS+K_bK_TS} \quad (4.9)$$

REQUIRED ADDITIONS FOR PLANT MODEL

In this subchapter, several required or even mandatory additions to the above DC-motor model are described. To setup a pre-finalized functional description and/or model of the total plant, all influences, additions and disturbances, that are present on the real lock-and-key system, must be appointed and, if not otherwise mentioned, implemented.

NO-LOAD SPEED CURVE

For this project, it was decided to assume a constant load torque. Moreover, this assumption was made to simplify the non-linear function and dynamic behavior of the load torque (Cholakkal, 2009). So, as a constant load is assumed, the speed is proportional to the supplied voltage. Thus, when supply voltage is constant, the speed must be inversely proportional to the load experienced by the motor (COLLINS, 2017). This relationship can be represented with motor's torque-speed cure - Figure below. Essentially, what this figure presents is the fact that an increase in the load torque on the motor would result in a decrease in speed.

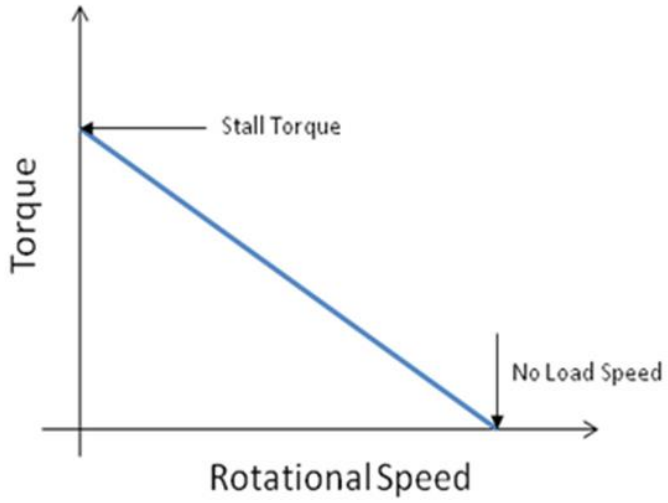

Figure 4. Load Torque versus rotational speed
Source: (COLLINS, 2017)

STARTING SATURATION

When no current is applied to the armature windings, no armature torque is realized. Due to the constant influence of surface friction as negative torque on the armature output, the model simulation is dependent of critically chosen start-up settings. There will only be a semi-positive armature torque output when the full impact of load torques is balanced. Therefore, the input torque entering the mechanical part of the DC motor must be saturated with a minimum of zero and as security interval with a maximum value equal to the maximum gearbox input torque.

START DELAY

There is a start delay in the system due to the backlash of the gears in the gearbox. This start delay is not described as time unit, but as manner of angular displacement. The model uses radians [rad] as applicable unit, where in the gearbox datasheet is spoken of degrees [deg] backlash. This start delay can be implemented as an actual delay function or with the IC (initial conditions) block.

NON-LINEAR SLIP STICK

Due to the slip stick effect between interacting mass surfaces, a larger current is drawn from the motor at start-up. This current is drawn to apply enough torque on the motor shaft to balance out the static forces/frictions. The current drops down when the mass is beginning to rotate. The influence of the static forces and frictions disappears at this moment and a new type of influence trade places. This new type is described as the dynamical influence, existing out of mass inertia, friction due to normal forces and/or applied external forces.

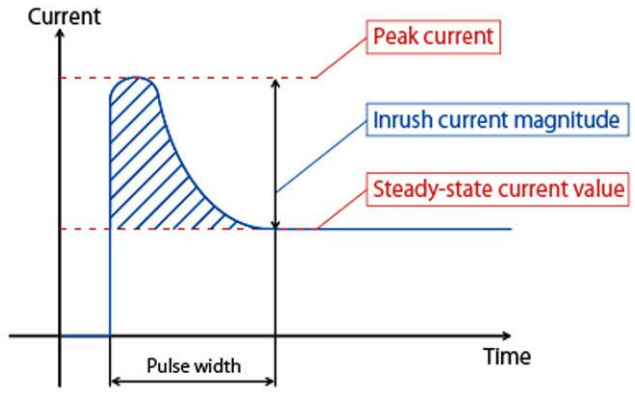

Figure 5. Current characteristic in addition to a pulse width
Source: https://www.sunpower-uk.com/glossary/what-is-peak-current/

Motor characteristics regarding the current are divided in two major variables, namely the peak current and the nominal or steady-state current. The absolute difference between both values indicates the inrush current magnitude and represents the overload current that can be applied on the armature windings for a short period of time without damaging or overheating the windings itself. When the motor enters his first power cycle, a peak current will rise and simultaneously charge the available capacitors in the power supply chain. When the peak current value is reached, the current drawn will reduce till it reaches the nominal current value.

SURFACE FRICTION

The only friction considered here, is the surface friction of the subsystem containing the gear rack and key motor assembly. The key is directly mounted on the output shaft of the second motor setup and has no (direct) friction nor continuously contact with other parts. Due to the low inertia of the key itself and the mediocre dynamical behavior of system, it is neglected out of the model and simulations.

In order to compute the disturbance torque encountered by the slide motor the following approach was used. Firstly, the force required for moving the key motor was determined. Hence, this was done by using the physical relationship between the coefficient of friction (commonly used values for gear rack are 0.1 to 0.15 (Dynamics, -))and normal force:

$$F_{friction} = \mu * F_N \tag{4.10}$$

Where the normal force F_N is obtained by multiplying the mass of the key motor with the gravitational acceleration constant g:

$$F_N = m * g \tag{4.11}$$

LOAD TORQUE

Secondly, the obtained force requirement ($F_{friction}$) had to be transformed into a torque requirement, or in other words – load torque (torque needed to move the key motor). This step was evaluated by using the assumed value (12 mm) for the radius (r) of the rotating disk, which is attached to the servo drive shaft. Thus, the rotational shaft of the servo drive system would need to overcome the following load torque:

$$T_L = F_{friction} * r \text{ (OMRON, -)} \quad (4.12)$$

GEARBOX REDUCTION

Finally, by dividing the obtained load torque with the gearbox ratio i_{gb}, the final torque balance is established. In most cases, and thus also in this case, a gearbox or redactor is used to transform the output (maximum) velocity of the DC motor to the velocity bandwidth where it is usable as input for the system itself.

Most DC motors or servomotors have a velocity bandwidth between 2500 till 6000 [RPM]. A gearbox will reduce the velocity and increase the output torque of the motor, in most cases. This is done with the gearbox ratio, that indicates the amplifier value between the amount of tooth on the input gear in comparison to the output gear. Modelled from the DC motor's perspective, the velocity is divided by this ratio and the torque is multiplied with it.

In the situation sketched above, the load is used as starting point and thus the gear ratio must be used inverted. To invert the load torque back to the motor shaft and subtract it from the armature torque, it must be amplified with the division of the gearbox efficiency coefficient C_{gb} over the gear ratio i_{gb}. The total sum of torque applied to the motor shaft T_m, is:

$$T_m = T_a - (\frac{C_{gb}}{i_{gb}} * T_l) \quad (4.13)$$

PARTICLE MASS INERTIA

The plant model is obtained by using two different setups. The first setup is the key motor. The key motor's mass inertia exists out of the mass inertia of the servomotor itself with an additional inertia of the gearbox. This gearbox inertia is added with the mass inertia of the servomotor and this is implemented in the transfer function of the plant.

$$J_{key} = J_m + J_g \quad (4.14)$$

For the second system, the story changes. The slide motor has a larger mass inertia due to the displacing function of the key motor. The mass inertia of the slide motor is a sum of the setup explained above with an additional inertia of the gear (disk) that is attached in the gearbox output shaft. This disk is transported back to the motor shaft by using the following equation:

$$J_{disk} = \frac{1}{i_{gear}^{\ 2}} * 0.5 * m_{disk} * r_{disk} \quad (4.15)$$

$$J_{slide} = J_m + J_g + J_{disk} \quad (4.16)$$

MODEL VALIDATION

As described in the methodology chapter, the validation of the model was performed by comparing the real systems' measurement data to the modelled one.

First, the EPOS2 controller is set up for the position control. This, by default, results in EPOS2 to set up PI current and PID position controllers. Further, the values of the controller gains are extracted from the software and converted to SI units like in Table 2.

Key Motor		
Current PI controller	*EPOS2*	*SI Units*
P Gain	651	2,543
I Gain	527	20586
Position PID controller	*EPOS2*	*SI Units*
P Gain	35	0,35
I Gain	103	8,034
D Gain	64	0,0051

Slide Motor		
Current PI controller	*EPOS2*	*SI Units*
P Gain	651	2,543
I Gain	543	21211
Position PID controller	*EPOS2*	*SI Units*
P Gain	32	0,32
I Gain	89	6,942
D Gain	60	0,0048

Table 2. EPOS2 gain conversion to SI units for use in the simulation

The SI values of the gains are then used within the PID SIMULINK blocks together with the simplified model of the system (described in previous chapter). Hence, the measured data was expected to match the simulation results. The comparison between the measured data and the simulation of the key motor are visible in Figure 7Figure 6 and Figure 6.

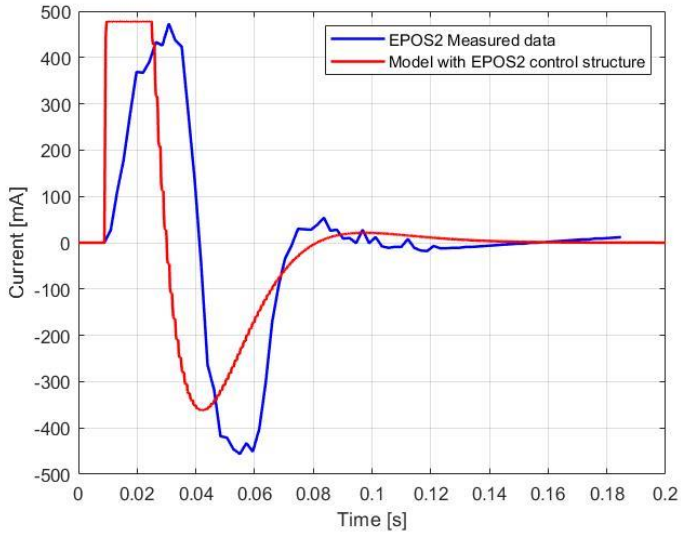

Figure 6. Comparison between the real system and the simulation response

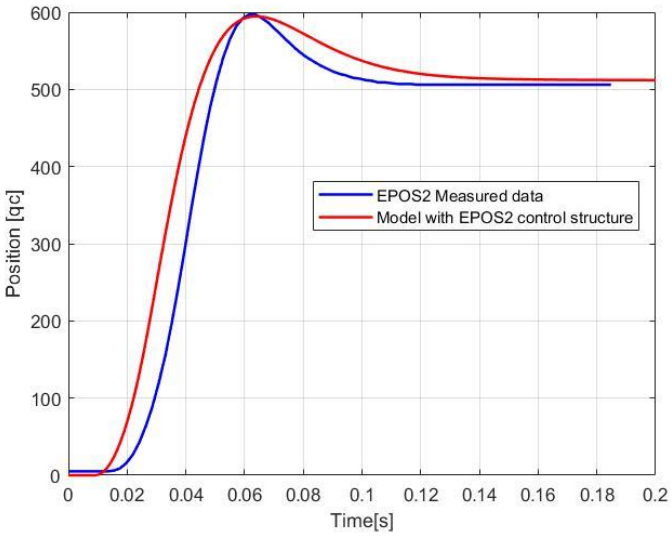

Figure 7. Comparison between the real system and the simulation response

As visible in both figures the simulation with the replicated EPOS2 controllers have a very similar responses with respect to current and position of the key motor. This proves that the model is truly a good representation of the real system.

PID CONTROLLER DESIGN VIA SIMULINK AUTOTUNE

Autotuning of a PID-block in Simulink is, when solely applied, an easy task to perform. But when there are multiple (PID)controllers involved, the task becomes more difficult. A PID-block in Simulink is designed in such a way that it only gains the right behavior or even find the right plant function when it is directly coupled with a transfer function.

Because the controller model is based on an inner loop, controlling the current, and an outer loop, controlling the position, there is one important statement that needs to be recognized while tuning:

The fastest response is applied by the inner loop controller and the outer loop needs to react ten times slower, as a rule of thumb. Also, during the tune process, it is important to check the domain of the controller and solver. In case of this project, many attempts where unsuccessfully executed, due to a fixed-step solver. The fixed-step solver made the system response discrete instead of time-continuous. The expected result didn't show up in the step response until the solver was set to variable stepsize.

KEY MOTOR | AUTOTUNE

In order to compare the results obtained from the root locus method and differentiate the performance with a second controller, the autotune method inside Simulink is used to obtain some PID-settings.

The first step is adding a PI-controller block as current controller in the short loop. After tweaking some response parameters, the following performance table is obtained, where the first one is the standard response of the model without PID-tuning and the second is the tuned response. The extremely small percentage overshoot is neglected. Thus, it is safe to conclude that the current will never exceed its maximum limit or get cut off by the limit safe and intervene with controlling the behavior.

Performance and Robustness

	Tuned	Block
Rise time	4.48e-05 seconds	7.46e+03 seconds
Settling time	7.63e-05 seconds	1.33e+04 seconds
Overshoot	0.0133 %	0 %
Peak	1	0.999
Gain margin	Inf dB @ NaN rad/s	Inf dB @ NaN rad/s
Phase margin	89.1 deg @ 4.77e+04 r...	90.1 deg @ 0.000295 r...
Closed-loop stability	Stable	Stable

Figure 8. Simulink PID Autotune performance for key motor current controller

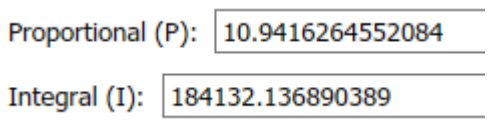

Figure 9. Simulink PID Autotune settings for key motor current controller

The second step is to control the position. This is done by short looping the PID-controller block for position with the plant model. Therefore, the current controller will not intervene with the output response while tuning. The response of the standard PID performance against the tuned one is shown below:

Performance and Robustness

	Tuned	Block
Rise time	0.00165 seconds	0.0273 seconds
Settling time	0.00314 seconds	0.0951 seconds
Overshoot	0.0542 %	5.13 %
Peak	1	1.05
Gain margin	42.3 dB @ 5.38e+04 ra...	51.4 dB @ 1.54e+03 ra...
Phase margin	86.4 deg @ 1.25e+03 r...	65.8 deg @ 50 rad/s
Closed-loop stability	Stable	Stable

Figure 10. Simulink PID Autotune performance for key motor position controller

To receive these performance characteristics, the following PID-gains must be implemented in the PID-controller:

Proportional (P): 21.1933903046011

Integral (I): 208.881396825971

Derivative (D): 0.195482407008888

Figure 11. Simulink PID Autotune settings for key motor position controller

The response of the closed-loop system, after tuning it within Simulink, is shown below:

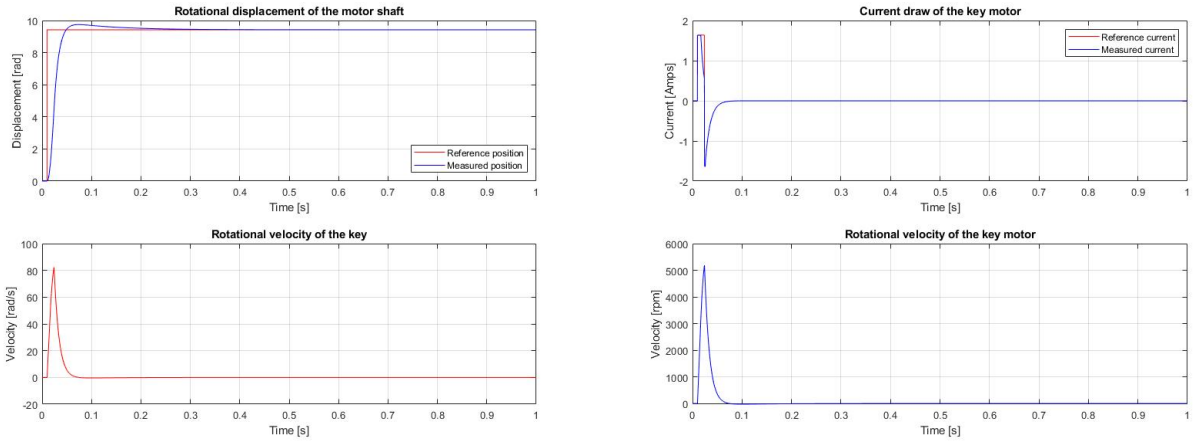

Figure 12. Step response of (autotuned) closed-loop response of the key motor

As visible in these graphs, the key motor shows a little bit over overshoot, but drives steady back to the given setpoint. It takes approximately 0.15 [s] to turn a precise 540 [deg]. But the key is already in position to enter the lock at 0.10 [s].

The current peak is only visible for a short time period of less than 20 [ms] and is not exceeding the maximum current limit as specified in the datasheet of the DC-motor. The motion profile of the motor shaft velocity is showing a tri-angular curve. Since the current is at his limit while accelerating/decelerating, it can be said that

this is the fastest response possible without exceeding the specifications/limits supplied by the hardware manufacturers.

Due to these observations, it can be concluded that the retrieved parameter configuration is applicable and meets the stated controller requirements.

As for the rule of thumb regarding speed ratio between inner and outer controller, it is calculated that the rise time ratio is 36.83 [-] in favor of the inner loop controller. The ratio in settling time is determined to be 41.14 [-] and thus also applies to the earlier described requirement.

SLIDE MOTOR | AUTOTUNE

After setting the parameters of the key motor controller to the obtained values, the controller for the slide motor can be tuned. The architecture of this system is majorly equal to that of the key motor. Where the key motor has a directly coupled load on the gearbox shaft, the slide motor has a disturbance load torque, larger mass inertia on the motor shaft and a second gearing (gear rack). The way this controller is configured, is equal to the steps that are performed for the key motor controller. The limitations for the hardware components are also equal, due to similar types of motor and gearbox. In the figure below, the performance of the tuned closed-loop output for current control is given:

Performance and Robustness

	Tuned	Block
Rise time	4.55e-05 seconds	7.46e+03 seconds
Settling time	7.74e-05 seconds	1.33e+04 seconds
Overshoot	0.0143 %	0 %
Peak	1	0.999
Gain margin	Inf dB @ NaN rad/s	Inf dB @ NaN rad/s
Phase margin	89.1 deg @ 4.7e+04 ra...	90.1 deg @ 0.000295 r...
Closed-loop stability	Stable	Stable

Figure 13. Simulink PID Autotune performance for slide motor current controller

The gains for the proportional and integral action are stated in the figure below:

Proportional (P): 10.7862225574014

Integral (I): 181411.343167786

Figure 14. Simulink PID Autotune settings for slide motor current controller

Now the tuning of the current controller is finalized, the outer loop controller for position can be accessed. After finding the right balance between robustness and settling time, without keeping sight on the requirements regarding stability and overshoot, the following performance is obtained:

Performance and Robustness

	Tuned	Block
Rise time	0.000783 seconds	0.0276 seconds
Settling time	0.00139 seconds	0.091 seconds
Overshoot	0.209 %	3.12 %
Peak	1	1.03
Gain margin	41.6 dB @ 6.48e+04 ra...	49.3 dB @ 1.5e+03 rad...
Phase margin	81 deg @ 2.32e+03 ra...	69 deg @ 51.1 rad/s
Closed-loop stability	Stable	Stable

Figure 15. Simulink PID Autotune performance for slide motor position controller

The controller gains that will lead to these performance characteristics are shown in the figure below:

Proportional (P): 45.1909493509304

Integral (I): 792.830287609687

Derivative (D): 0.301292972239938

Figure 16. Simulink PID Autotune settings for slide motor position controller

The values of both tuning sequences are applied on the PID-blocks and the following (step) response is obtained:

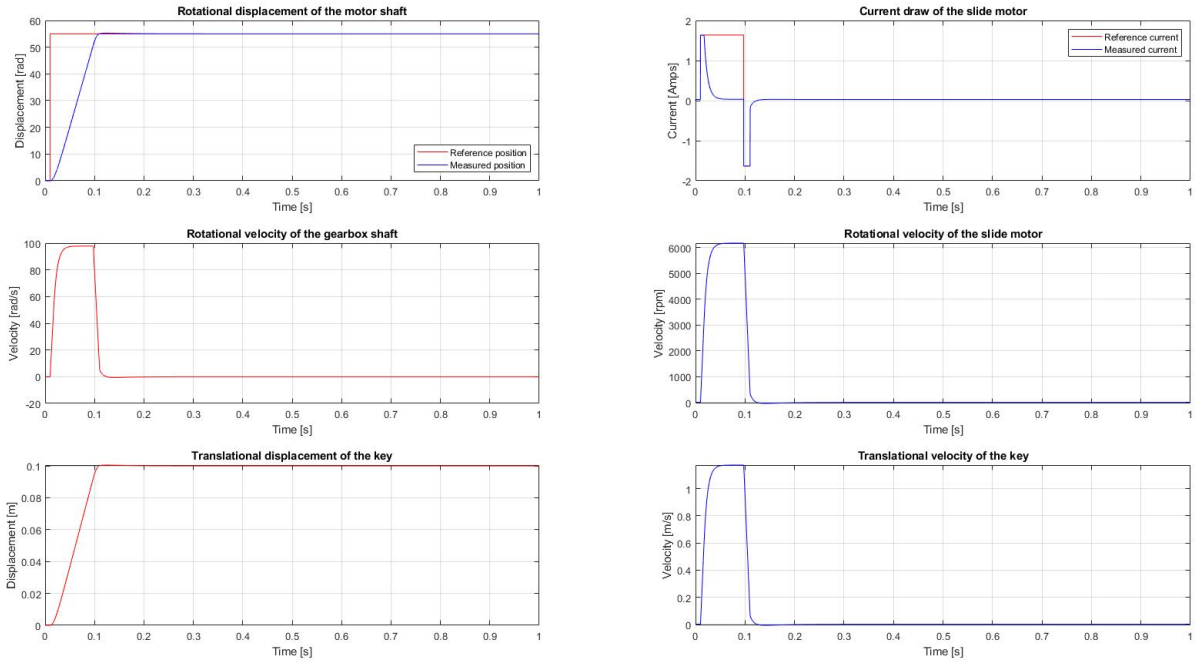

Figure 17. Step response of (autotuned) closed-loop response of the slide motor

By comparing the settling time of the inner loop current controller with the outer loop position controller, it can be concluded that the minimum ratio of ten is not compromised. The ratio in settling time between the two loops is 17.96 [-], in favor of the (fast) inner loop.

The rise time of this inner loop also applies to the rule of thumb and is by means a factor 10.12 [-] smaller/faster than the outer loop.

The maximum measured velocity of the DC-motor motor shaft is not exceeding the limit that is given by the datasheet, provided by Maxon. Also, the rotational velocity of the gearbox output shaft is 6.6 times smaller than the motor velocity and shows exactly the same behavior. This indicates that the gearbox is implemented in the right way.

The rotational motion on the gearbox output shaft is transformed to translational motion by means of a second gearing in form of a gear rack. The translational displacement of the load (key system) is, with a negligibly small overshoot of $e(x) < 0.002$ [%], precisely following the applied value in the step response and shows a trapezoidal motion profile. This trapezoidal motion profile is obtained by reaching the maximum allowed motor velocity of near over 6000 [rpm]. This means that the motor velocity is maximum for a short amount of time before decelerating to zero. The observation can be made that the closed-loop response is, as formulated in the controller requirements, as fast as possible without compromising the key's position relative to the lock.

OVERALL OBSERVERATIONS | AUTOTUNE

A global comparison of both current controller gains shows that the quantity of the proportional and integral terms are in the same range. There is a minor variety of less than 1.5 [%] between the gains of both controllers.

As for the quantity of the position controller gains, both the proportional, integral as well as the derivative are in the same range. The difference between the controller values is the consequence of an increase in mass inertia and application of an external load on the slide system.

PID CONTROLLER DESIGN VIA ROOT-LOCUS

In the following sub-chapter, the design procedure of the obtained system compensators by using root-locus method is presented. For even more elaborate stepwise procedure see Appendix D Root locus description. Later in the chapter, the results of applying the RL compensators on the modelled system are observable.

KEY MOTOR | ROOT LOCUS

PID POSITION CONTROLLER DESIGN

Firstly, it is of great importance to propose a design criterion that the controller should meet. For the key motor the following design requirements are stated:

- Settling time of less than 0.01 seconds;
- Maximum overshoot of 15%;
- No steady – state error.

The design procedure is started by presenting the open-loop transfer function:

$$G(s) = \frac{Output\ C(s)}{Input\ R(s)} = \frac{\theta(s)}{V(s)} = \frac{K_T}{L_a J_m S^3 + (R_a J_m + L_a B_m) S^2 + R_a B_m S + K_b K_T S} \quad (4.9)$$

With $J_m = J_{key} + J_g = 6.14 * 10^{-7}\ [kg\ m^2]$, $L_a = 0.000231\ [H]$, $R_a = 3.69\ [Ohm]$

$B_m = 1 * 10^{-6}\ \left[\frac{Nm}{rad}\right]$ and $K_b = K_t = 0.0184\ [-]$

Parameters	Uncompensated	PD	PI
Plant & Compensator	$G(s) = \frac{K*0.0184}{1.418*10^{-10}S^3 + 2.266*10^{-06}S^2 + 0.0003389S}$	$(s + 867.861)$	$\frac{(s+0.1)}{s}$
Dominant poles	-71 +/- 118i	-1590+/-520i	-1590+/-520i
Ts	0.0435	0.00328	0.00328
K	2.43	0.3343	0.3343
Overshoot	1.32	15	15
e(∞)	0	0	0

Table 3. Values obtained from the root locus technique for key motor position via MATLAB

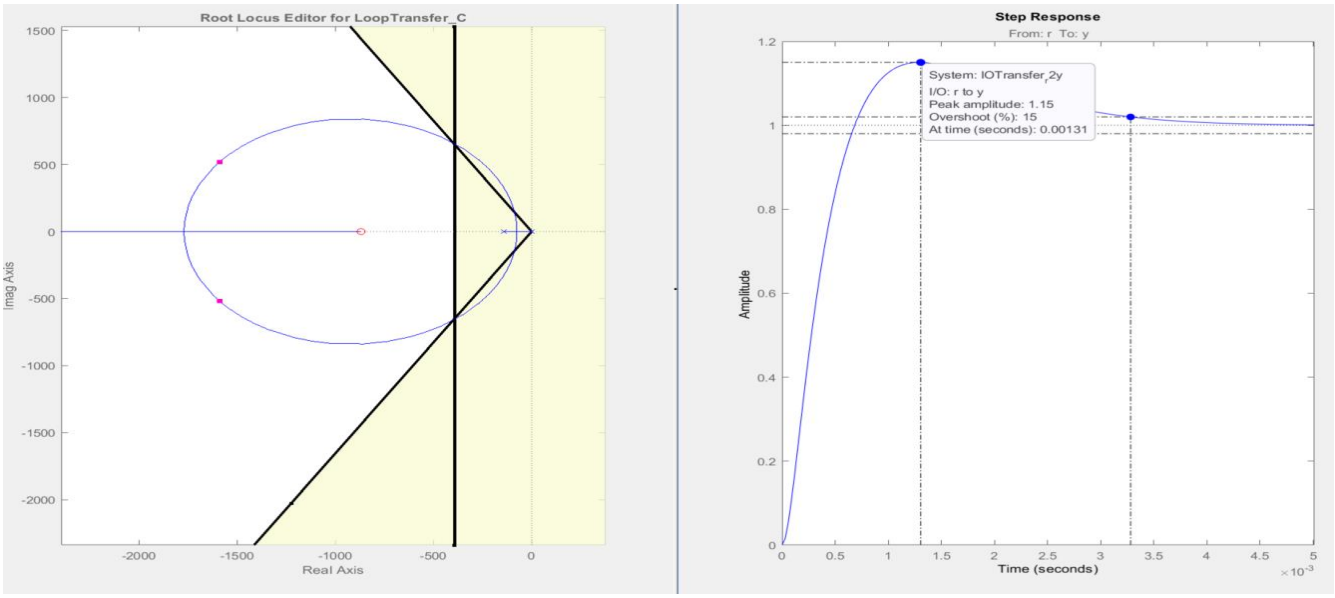

Figure 18. Step Response of the PID controller for key motor position

Initially, the transfer function of the uncompensated system is plotted via root locus technique and from the damping ratio line, the gains are moved to find the dominant pole location which is at -71+/- 118i. From the dominant pole location, the gain values and the settling time can be found. In order to reduce the settling time, the zero-pole location is found to be at 867.86 of the PD controller such that the settling time reduces from 0.0435 of uncompensated to 0.00328.

The transfer function of the PD controller is $G_d = 0.3343 * (s + 867.861)$

After designing the PD controller, the PI controller has been introduced to drive the steady to zero. Zero location has been arbitrarily chosen closer to the origin in order to get the pole zero cancellation. From the **Error! Reference source not found.**, it is clear that the PID controller has been used to improve the steady state error and transient response of the system.

$$G_i = \frac{0.3343(s+0.1)}{s} \qquad (4.14)$$

From the Figure 18, settling time of the position controller has decreased by increasing the gain values. Thus, making the system to perform quickly by reducing the settling time.

After designing the separate parts of the controller, the PID compensator gains can be extracted from the overall compensator transfer function by multiplying PI&PD.

$$G_c(s) = G_i(s)G_d(s) = \frac{0.3343s^2+290.158s+29.1}{s} = \frac{K_p s+K_i+K_d s^2}{s} \qquad (4.19)$$

Where, $G_i(s)\&G_d(s)$ are the designed PI&PD controller, which is obtained in the Root locus technique.

Hence, the gains to be inserted in the PID SIMULINK block are K_p = 290.158, K_i = 29.1and K_d = 0.3343.

PI CURRENT CONTROLLER DESIGN

Same as for the position controller, some possible design criterion must be stated:

- Settling time of lower than for the position controller;
- Zero steady-state error on the step response.

The derivation of the open-loop transfer function for the current controller resulted in:

$$G(s) = \frac{Output\ C(s)}{Input\ R(s)} = \frac{I(s)}{V(s)} = \frac{J_m s + B_m}{L_a J_m S^2 + (R_a J_m + L_a B_m)S + R_a B_m + K_b K_T} \quad (4.20)$$

Where the parameter values are the equal to the ones used for the PID position controller. First, the uncompensated system is plotted in the s-plane and the root-locus of the system is investigated.

Parameters	Uncompensated	PI
Plant & Compensator	$G(s) = \frac{6.14*10^{-7}s+1*10^{-7}}{1.418*10^{-10}s^2+ 2.266*10^{-6}s+0.0003389}$	$\frac{(s+0.1)}{s}$
Dominant poles	NA	-1.05*10^8
Ts	0.0333	0.00282
K	NA	24158
Overshoot	15	15
e(∞)	NA	0

Table 4. Values obtained from the root locus technique for key motor current via MATLAB

Further, the following PI compensator is added to the root-locus:

$$G_i(s) = \frac{s+0.1}{s} \quad (4.21)$$

This action reduces the steady-state error to zero. Additionally, to achieve a faster settling time than the one for PID position controller, the dominant poles are moved along the root-locus. Thus, at a gain of 10 000 the settling time of 0.00282 seconds is found. The root-locus and step response of the compensated system are visible in Figure 19.

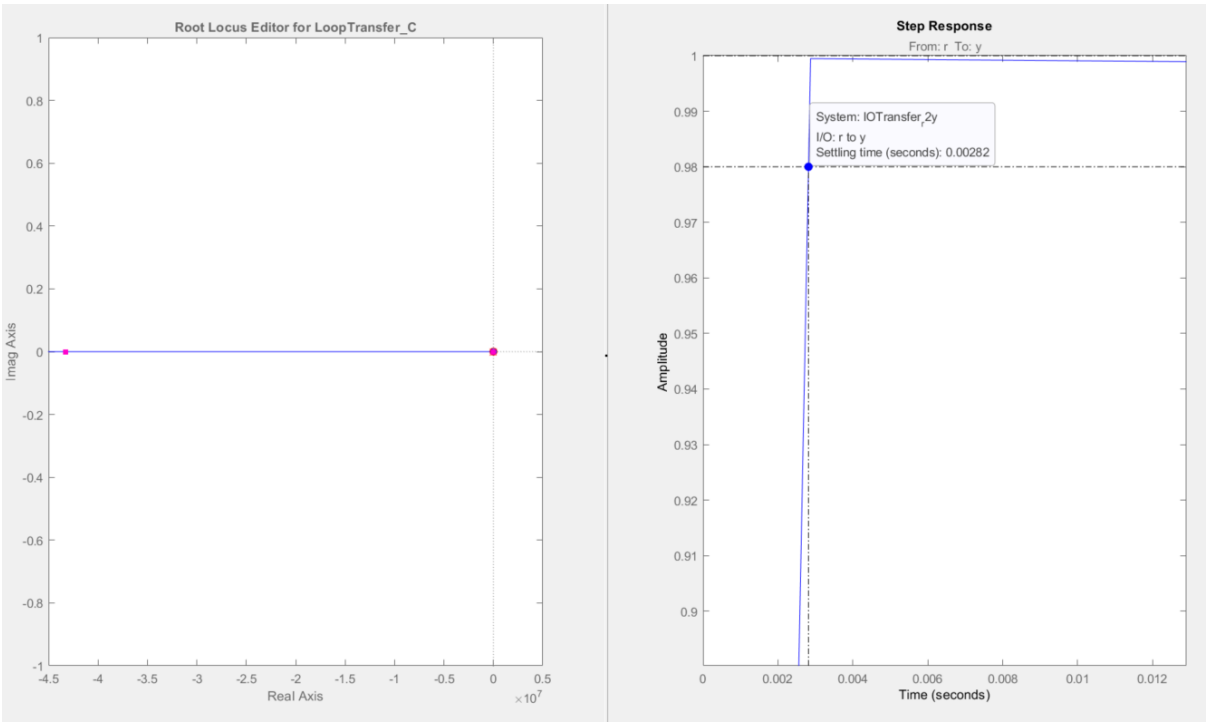

Figure 19. Step Response of the PI controller for key motor current

Since there were no dominant poles due to zero percent overshoot, PD controller implementation was not possible even though designing it, the changes were quite negligible in the overall system. Therefore, PI controller is implemented to calculate the respective gain values of the system.

The resulting overall PI compensator transfer function is:

$$G_c(s) = G_i(s) * 10000 = \frac{10000s+1000}{s} = \frac{K_p s+K_i}{s} \quad (4.22)$$

Where, $G_i(s)$ is the designed PI controller, which is obtained in the Root locus technique. Hence, the PI gains are Kp = 10000 and Ki = 1000.

SLIDE MOTOR | ROOT LOCUS

PID POSITION CONTROLLER DESIGN

Similar to the Key motor, PID controller has been introduced to improve the steady state error and transient response of the system. From the below Table 5, uncompensated and compensated values of the slide motor (position controller) are obtained from the root locus technique.

$$G(s) = \frac{Output\ C(s)}{Input\ R(s)} = \frac{\theta(s)}{V(s)} = \frac{K_T}{L_a J_m S^3 + (R_a J_m + L_a B_m) S^2 + R_a B_m S + K_b K_T S} \quad (4.23)$$

Parameters	Uncompensated	PD	PI
Plant & Compensator	$G(s) = \frac{K*0.0184}{1.495*10^{-10}S^3+2.388*10^{-06}S^2+0.0003389S}$	$(s+250)$	$\frac{(s+0.1)}{s}$
Dominant poles	-71.4-18.7i	-88.2+23.6i	-88.2+24i
Ts	0.074	0.0549	0.0538
K	0.70121	0.0042748	0.0042958
Overshoot	0	0	0
e(∞)	0	0	0

Table 5. Values obtained from the root locus technique for slide motor position via MATLAB

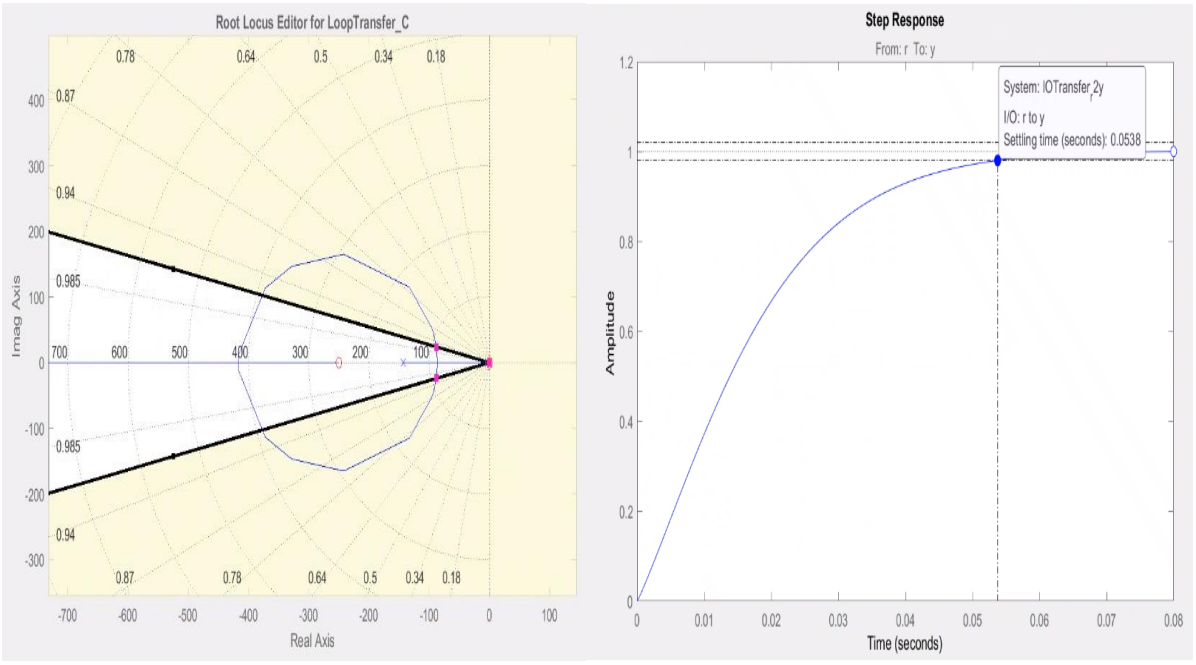

Figure 20. Step Response of the PID controller for slide motor

The same procedure which was used to control the key motor position is followed for the slide motor. From the Figure 20, the graph of the slide motor is quite similar to the graph of key motor as there were only few changes in the inertia of the motor. It can be clearly seen that the slide motor position settling time has been decreased from uncompensated 0.074 to 0.0538. In order to obtain the zero location, the percentage overshoot value is considered as 0.01 [%] instead of 0 [%].

The transfer function of the PD controller is $G_d = 0.0042748(s + 250)$

The transfer function of the PI controller is $G_i = \frac{0.0042958(s+0.1)}{s}$

After designing the separate parts of the controller, the PID compensator gains can be extracted from the overall compensator transfer function:

$$G_c(s) = G_i(s)G_d(s) = \frac{0.0043s^2+1.00043s+0.1075}{s} = \frac{K_p s+K_i+K_d s^2}{s} \quad (4.24)$$

Where, $G_i(s)\&G_d(s)$ are the designed PI&PD controller, which is obtained in the Root locus technique. Hence, the gains to be inserted in the PID SIMULINK block are K_p = 1.00043, K_i = 0.1075 and K_d = 0.0043.

PI CURRENT CONTROLLER DESIGN

As discussed above for key motor current controller, PI controller has been used to improve the steady state error of the system. From the below Table 6, uncompensated and compensated values of the slide motor (current controller) are obtained from the root locus technique.

As there were no intersection to find dominant poles for zero percent overshoot. PD controller implementation was not possible even though designing it, the changes were quite negligible in the overall system. Therefore, PI controller gain can be found by arbitrarily selecting the gain value on the real axis line.

Parameters	Uncompensated	PI
Plant & Compensator	$G(s) = \frac{6.417*10^{-7}s+1*10^{-7}}{1.49*10^{-10}s^2+2.388*10^{-6}s+0.0003389}$	$\frac{(s+0.1)}{s}$
Dominant poles	NA	-2.23*10^8
Ts	0.0333	0.00208
K	NA	51586
Overshoot	0	0
e(∞)	NA	0

Table 6. Values obtained from the root locus technique for slide motor current via MATLAB

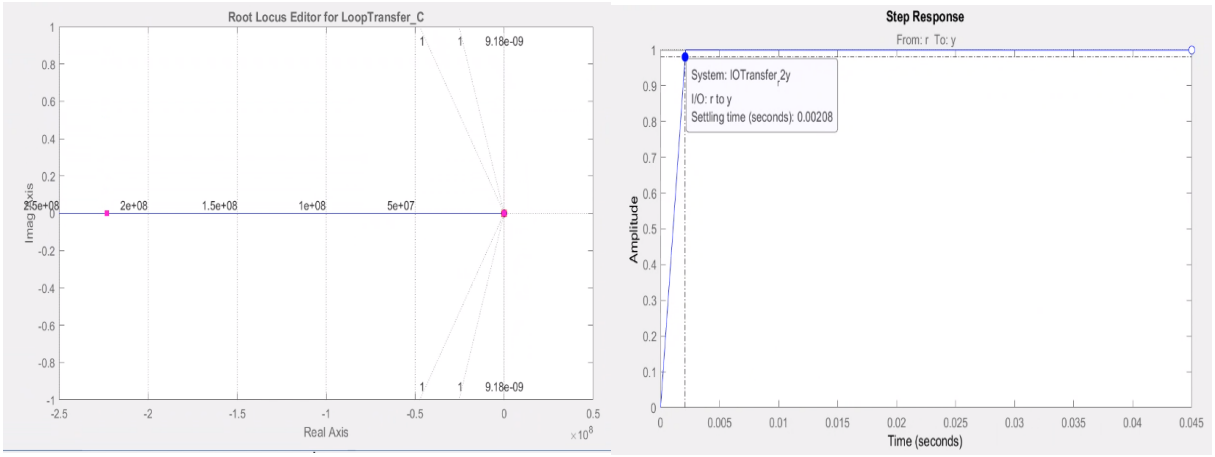

Figure 21. Step Response of the PI controller for slide motor current

By comparing both key motor and slide motor step response graph it can be noticed that the settling time is similar to that of key motor. As the value of the gain been selected arbitrarily, the trend of settling time will decrease. The steady state error of the system will attain the value zero.

The resulting overall PI compensator transfer function is:

$$G_c(s) = G_i(s) * 51586 = \frac{51586s+5158.6}{s} = \frac{K_p s+K_i}{s} \qquad (4.25)$$

Where, $G_i(s)$ is the designed PI controller, which is obtained in the Root locus technique. Hence, the PI gains are Kp = 51586 and Ki = 5158.6.

ROOT LOCUS CONTROL

Results of implementing the previously obtained controller gains within the SIMULINK PID blocks are visible in Figure 22(key motor) and Figure 23(slide motor). Both figures show how the position and current controllers are controlling the respective variables within the motors.

Figure 22 shows the system response on a rotational displacement setpoint of 540 [deg]. It can be noted that there is some overshoot present, however, this is acceptable for the key. Furthermore, it is acceptable due to the motion profile introduced in the Introduction chapter. To recall, the key should rotate in its final position before it enters the lock. Hence, by looking at the **Error! Reference source not found.**, this requirement is achieved as the translational displacements' final position is reached after the key has rotated in its final position.

By looking at Figure 23 one can see the behaviour of the translational displacement of the key. An overshoot of around 4 [%] is present. However, this amount is acceptable for the given application. Regards the current controllers within the both motors, they are behaving very similarly. Additionally, it can be seen that initially the measured current is not following the reference one, but it starts to replicate it after the first decrease in the curvature of the reference current.

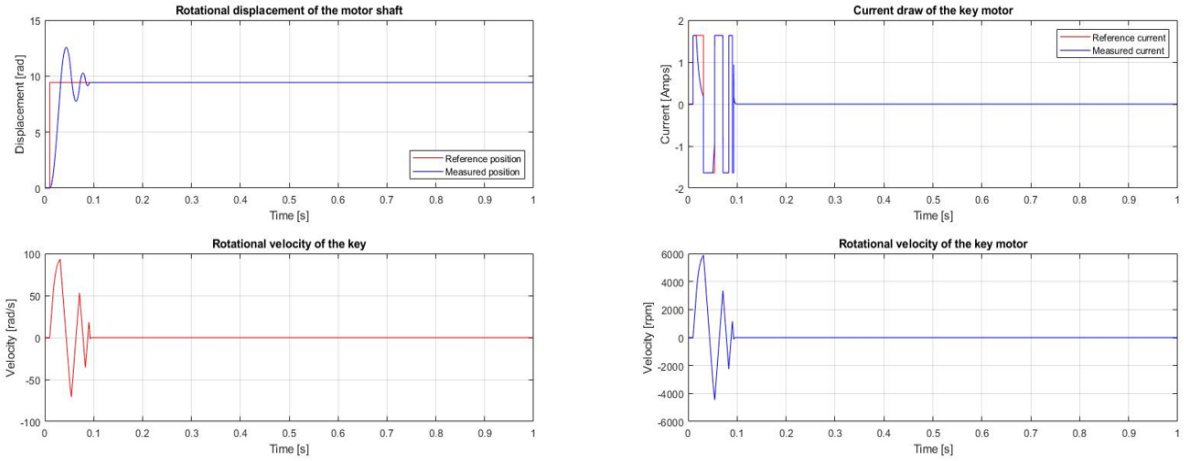

Figure 22. Key motor - System response, input: rotational displacement setpoint of 540 [deg].

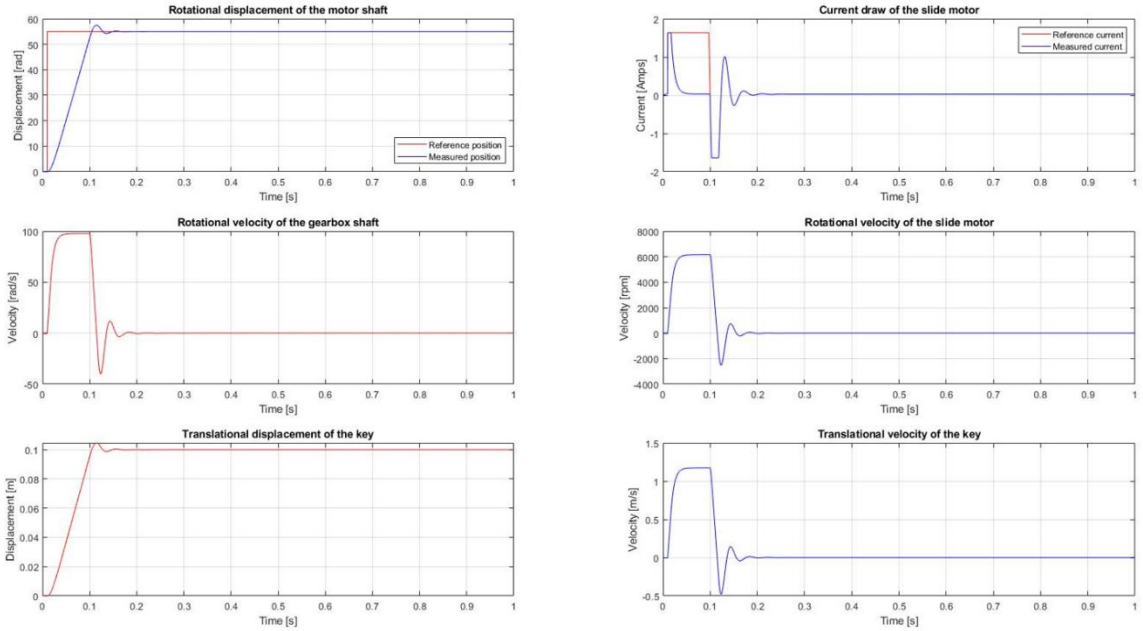

Figure 23. Slide Motor - System response, input: translational displacement setpoint of 0.1 [m].

5. DISCUSSION

In this chapter we are going to discuss the results gained with our model of the plant and controller.

LOAD TORQUE DISTURBANCE

The model contains only the frictional force by the surface. Non-linear slip stick and friction between the tooth of the gear is not implemented. A better load torque disturbance can be obtained by plotting the actual response of the system with the step response, calculating the exact residuals between the two curves and find add a linear regression on the residuals for implementation on the plant model.

CURRENT CONTROLLER DESIGN

A PID controller is chosen to be the best fit, due to the controller principle it is based on. In case of this system, both the plant parameters as well as the disturbances are not fully determined. The transient response of the system is analyzed by comparing the model simulation with measurement data. In global, both responses show similar behavior, form and quantity in input/output, but similar is not precisely. There are still some defects or minor differences between both. This can be due to the fact that the simulation is in continuous time, where the measurements are based on discrete signals and a partially discreet system. As mentioned earlier, there are some parts and parameters of both the plant as the disturbances that need improvement or need to be determined more precisely. When this is done, more (stable) options for controllers will be available to implement. Investigation of multiple controller principles showed that PID is in the current situation the best fit due to its response based on the error between setpoint (input) and controlled value (output). For example, a feedforward controller needs an exact description of both plant as disturbances. If this is not provided, the controller will not have the right response. Other possibilities for (feedback) control is lead/lag, who both show major similarities with a PID-controller. Therefore, the PID controller is chosen to be best suitable for the current phase of the project.

UPGRADED CONTROLLER DESIGN

A (possible) solution for an upgraded controller design is using the feedforward controller in combination with a feedback controller. The feed forward will respond on the known model functions and the error that still remains, thus is not described in the feedforward function, can be solved/controlled by the feedback controller

If a combination of feedforward and feedback is not desirable, or better tracking performance is expected, a second (possible) solution can be designed. When the complete model is accurately formulated and mathematically described, an expanded and more detailed plant function can be obtained. In addition a new motion profile can be obtained from the real system. The current measurements only include step responses with the Epos controller. The motion profile used is only an educated guess from the project members. When the exact motion profile is obtained including balanced translational and rotational displacement, a feedforward controller can be designed and implemented. Using this feed forward controller, disturbances can be isolated and performance can be optimized, resulting in faster response- and settling time. This will eliminate all small residuals and offsets that appears in the current closed-loop responses. The feedback controller only influences the input signal based on what is already happened. A feedback controller has a predictive function, in order to optimize the plants input based on the timespan in which the motion track, applied disturbances and system behavior occurs.

NON-LINEAR SLIPSTICK

The non-linear slip stick will cause a current peak in the beginning. This current peak is not observed in the step response of the plant model. This current peak will alter the current to start higher when the motor shaft is actually starting to rotate and thus will influence the behavior of the output (angular velocity).

EPOS SYSTEM MEASUREMENTS

The measurements are a good estimation of the used values in the plant model. There is a small delay between the measured position on the Epos set-up and the simulated plant behavior. It can be said that this delay is caused by the backlash of the gears inside the gearbox. Also the Epos controller has a lower negative current (reversing the direction of the servomotor). This causes the displacement behavior to settle down faster to the final position. This can be caused by not completely bypass the epos controller.

MOTOR AND CONTROLLER WIRE RESISTANCE

Thus perhaps not completely negligibly is the wiring that is used to power the DC-motors. If the wiring is getting hot, this will influence the overall resistance, creating a lower current to flow. In the beginning, the wire will be cold and thus this can be neglected. But the performance of the DC-motor can vary over time, resulting in a higher output of the controller. Because the motor input is voltage and this is maxed out at 12 [VDC], the maximum current flowing through the armature is (partially) limited by the wire resistance. A lower current through the armature will decrease the Lorentz power by the coil and thus also the maximum torque that can be applied. To verify this, an additional endurance test can be performed.

ROOT-LOCUS CONTROL

The root locus technique is considered as a ‘royal road’ to designing a controller. However, as observed from the design procedure, this method has its drawbacks. For example, take the key system. The controller was designed by using the RL tool available from MATLAB, thus, the step response (of controlled system) within this tool was observed to have an overshoot of 15% and a settling time of around 0.003 seconds (these were the design requirements). Next, the PID gains were implemented within the SIMULINK and a completely different response was observed. In more detail, the overshoot showed to be around 23% with a settling time of approximately 0.1 seconds. This difference is due to the saturation of current and voltage within the developed model. So, it can be summed up that the root-locus method is viable, unfortunately, it does not account for the system limits. Furthermore, this could be overcome by considering the limits before starting the design procedure. Regrettably, the saturation limits of the system were not taken into account prior the root-locus design within this project.

6. CONCLUSIONS AND RECOMMENDATIONS

The main objective of the project was to realize an accurate positioning and rotation of a Lock and Key system consisting of two servomotors. Thus, this objective is achieved as a fully controllable system is obtained. Additionally, the system was initially modelled by using its' mathematical description and validated with measurements from a real-life system. So, it can be stated that the developed model is a good representation of the real lock and key system.

Further, two control design methods were used to obtain a controlled system. First, the root-locus method showed that satisfactory results can be obtained, however, the systems' limits should be considered very carefully while designing a controller via root-locus. Next, the autotuning was used to make optimal controllers for the system. This method presented an easier and straight to the point approach on designing a controller. Hence, it is concluded that both methods are feasible for controlling the system in question.

Regards recommendations, several points of interest can be proposed for further study:

- Velocity controller for even more 'fully controlled' process. This could imply that the speed of the translational and rotational motion could be controlled;
- Investigation and implementation of the disturbances within the model would give an even more realistic representation of the real system. Also, it would allow to implement an ideal feedforward controller to account for the setpoint changes;
- The measurements consist out of discrete signals (time discrete or z-domain). The system is divided in an analog part and a digital part. In between an ADC (Analog to Digital converter) is implemented to transform signals input and output signals to the expected domain. Because the measurements are discrete, better analysis can be performed by transforming the measurements to time-continuous signals/values or transforming the simulation response to a discrete response. Important is that the same sample frequency is applied on both sides;
- Human-Machine-Interface (HMI) could be designed for the developed controlled system. This would allow to manipulate the responses of the system. Also, this would make it reasonably visual to people with no engineering background.

7. REFERENCES

-. (2018, July 25). *Root Locus Technique in Control System | Root Locus Plot*. Opgehaald van Electrical4U: https://www.electrical4u.com/root-locus-technique-in-control-system-root-locus-plot/

(DunkerMotoren), U. (2014). *FTP*. Opgehaald van Dunker Motoren: https://www.dunkermotoren.com/fileadmin/files/knowledge/White_Papers/Whats_exactly_a_servo motor_Whitepaper_EN_20180129.pdf

Campo, A. (2012, September 26). *PID Control Design.* Opgehaald van IntechOpen: https://www.intechopen.com/books/matlab-a-fundamental-tool-for-scientific-computing-and-engineering-applications-volume-1/pid-control-design

Cholakkal, S. (2009). *Load Disturbance Torque Estimation for Motor Drive Systems.* Ontario, Canada: University of Windsor.

COLLINS, D. (2017, March 9). *The Torque Equation and the Relationship with DC Motors*. Opgehaald van Motion Control Tips: https://www.motioncontroltips.com/torque-equation/

Collins, D. (2019, October 02). *What is a motion profile*. Opgehaald van MotionControlTips: https://www.motioncontroltips.com/what-is-a-motion-profile/

Dynamics, A. (-, - -). *CALCULATING RACK AND PINION, HOW DO YOU DO THAT?* Opgehaald van APEX Dynamics: https://www.apexdyna.nl/en/calculate-rack-and-pinion/

Dynapar. (2020). *Encoders*. Opgehaald van Dynapar: https://ecatalog.dynapar.com/ecatalog/absolute-encoders/

Ellis, G. (2012). Control System Design Guide. In G. Ellis, *Control System Design Guide.* Butterworth-Heinemann.

Hasan, M. (2020). *Top 15 Best Embedded Systems Programming Languages*. Opgehaald van Ubuntu PIT: https://www.ubuntupit.com/top-15-best-embedded-systems-programming-languages/

Kaiser, D. (2003, February 07). *Fundamentals of Servo Motion Control*. Opgehaald van Automation: https://www.automation.com/en-us/articles/2003-1/fundamentals-of-servo-motion-control

Kostadin Kostov, V. K. (2008). *Robust Root Locus Application in Design and Analysis.* Sofia: Technical University, Faculty of Automation, Department of Industrial Control Systems.

Linzenkirchner, E. (1999, October 18-23). *The Importance of Control Engineering in Automation.* Opgehaald van Plant Automation.com: https://www.plantautomation.com/doc/the-importance-of-control-engineering-in-auto-0001

Makableh, Y. (2011). *Efficient control of DC servomotor systems using backpropagation neural networks.* B.S. The University of Jordan, Jordan.

Nise, N. S. (2010). Design via Root Locus. In N. S. Nise, *Nise's Control Systems Engineering* (pp. 449-524). Pomana: Wiley.

OMRON. (-). *Technical explanation for inverters.* -: Omron.

Sargent, R. G. (2011). *Verification and validation of simulation models.* Syracuse, NY, USA: Syracuse University.

Wikipedia. (sd). *PID Controller*. Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/PID_controller

APPENDIX A NOMENCLATURE

SYMBOLS

R_a	Armature resistance	[ohm]
L_a	Armature inductance	[H]
K_t	Torque constant	[Nm/A]
K_b	Back EMF constant	[Vs/rad]
V	Voltage	[V]
I	Current	[A]
B	Damping coefficient	[Nms/rad]
J	Mass moment of inertia	[kg*m^2]
T	Torque	[Nm/s]
θ	Angular displacement	[rad]
ω	Angular velocity	[rad/s]

ACRONYMS

EMF	Electromotive force
MMF	Magnetomotive force
RL	Root locus
PID	Proportional-Integral-Derivative
SP	Setpoint
CV	Controlled value
I/O	Inputs/Outputs

APPENDIX B LITERATURE SURVEY

EXISTING TYPES OF SERVOMOTORS AND SERVOSYSTEMS

"What exactly is a servomotor?" is a commonly asked question. There are multiple answers possible, so let's go the roots of the name. Servo is derived from the Latin word "Servus" that means sleeve. Motor is derived from the word "Moto" and means move. The translation of the term servo motor is thus "sleeve move" or "moving sleeve".

Servomotors can create two main types of mechanical motion. The first one is rotary and the second one is linear. The most common motion form is rotary. A servomotor normally exists out of a rotor and a stator. One contains a permanent magnet and the other an electric. As the name indicates, the rotor turns, where the stator is a fixed part inside the motor. The difference between a servomotor and a normal electric motor is the feedback possibility. The Servo motor comprises of three wire system known as Power, Ground and Control whereas DC motor is two wire system known as Power and Ground. Servo motor has an assembly of four things DC motor, gearing set, control circuit and a position sensor. DC Motor does not comprise of any assembly. A servomotor is controllable due to the feedback function about his position, speed and/or acceleration. This can be also due to power-regulation ((DunkerMotoren), 2014).

ELECTROMECHANICAL DEFINITION OF A SERVOMOTOR

The servomotor is build out of different mechanical and electrical components, together yield the possibility to control a certain motion. In **Error! Reference source not found.**. the circuit diagram of a typical servomotor is showed (Nise, 2008).

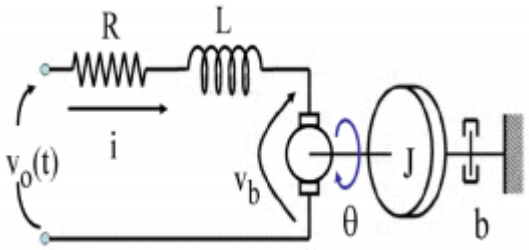

Figure 24. Circuit diagram of a typical servomotor

As shown, the circuit is divided into an electrical part and a mechanical part. The electrical part exists out of a coil that is displayed in a (permanent) magnetic field. This coil developes a resistance R, an inductance L and a back electromagnetic force (EMF) V_b. The EMF is a result of applying a current by the power source V_0. Due to the presence of this back EMF, the motor is a self-regulating machine. The motor only draws an armature current that is in relation with the required torque needed for the load.

The mechanical part exists out of a shaft where a rotational force is applied, the corresponding inertia J and a certain damping b.

The fundamental reason of using a servo motor is basically because the need for improving transient response, reducing steady state errors and reducing the sensitivity to load parameters. When applying a (controlled) servo system, the load may vary and the delivered torque will be adjusted to the amount needed to fulfill the desired setpoint or SP.

SPECIFIC COMMUNICATION PROTOCOLS USED FOR SERVOMOTORS AND/OR SERVOSYSTEMS?

Communication is one of the pillars for the evolution of the human species. But this formulation also applies in the evolution of technology.

Communication in the world of servo system is described as the method in which data travels between two or more subsystems. These subsystems can be component but can also be smaller on for example software architectural basis, controller states or sub-states.

The most apprehensive communication method is between components and are often frequency- or pulse based. Common communication or data protocols are mentioned below (Makableh, 2011).

RS232/485 (SERIAL COMMUNICATION)

Single-master, single-slave (RS232) or multi-slave (RS485) protocol
Communication speed of max. 115 [kbit/s]
Limited length of signal wires. RS485 is extended in comparison to RS232.
RS232 (half-duplex) / RS485 (full-duplex)

CANBUS (SERIAL COMMUNICATION BY BOSCH)

Multi-master, multi-slave protocol, uses nodes to prioritize
Communication speed of max. 10 [Mbit/s]
Extensively used in automotive for environments with (possible) EMC

ETHERCAT (FIELDBUS SYSTEM BY BECKHOFF)

Single-master, multi-slave protocol
Communication speed up to 100 [Mbit/s]
Uses networkcable for connections
Multi-layer for communication setup (4 host, 3 media)

MODBUS TCP/IP (SERIAL COMMUNICATION BY SCHNEIDER ELECTRIC)

Single-master, multi-slave protocol
Invented for communication of PLCs
Multi-layer for communication setup (Link, Internet, Transport and Application)

ETHERNET-IP (ETHERNET BASED COMMUNICATION BY ODVA)

Single-master, multi-slave protocol
Communication speed up to 100 [Mbit/s]
Uses Common Industrial Protocol (CIP)
Multi-layer for communication setup (4 host, 3 media)
Real-time communication

CODING LANGUAGES OFTEN USED TO PROGRAM A SERVOSYSTEM

Most industrial servo drives come with integrated software platforms where the engineer can program and configure the servo motor and it's response.

The position and behavior of servo motors is widely determined by using the output of different sensors. Commonly used are encoders and other positioning sensors for providing relevant information in the motion process.

Open coding languages that are used for programming software and for this subject also motion programs, are (Hasan, 2020):

- C/C++/C#
- Python
- Ruby
- Typescript
- VB.NET
- MINT
- R programming language
- Java
- JavaScript

* *More information and highlights about several programming languages are available at:* https://www.ubuntupit.com/top-15-best-embedded-systems-programming-languages/

CONTROLLERS AVAILABLE FOR CONTROLLING SERVO MOTORS

There are multiple manners in which a servo system can be controlled. The most commonly used control principles are feed forward, PID and PIV.

FEED FORWARD

Feed forward is based on predicting the future and creation such a reaction that the following error is zero. This prediction is often created by the operator, that signs a pre-defined path in which the system must control. It does not react on the systems behavior due to changes in environment or load. In this control manner, the load is not part of the controlled behavior. The system is not error-based, but so-called reaction-based. It only reacts on the feedback of the sensor data in order to maintain the desired control strategy or process description. The quality of the process description in a mathematical model determines the quality of the controlled result (Kaiser, 2003).

PROPORTIONAL POSITION LOOP-INTEGRAL-PROPORTIONAL VELOCITY LOOP OR PIV

The main difference between a PID- and a PIV-loop is that the PIV loop also reacts on the changes in velocity, where a PID-loop only reacts on position.

The velocity function in this control loop begins by multiplying the position error in the proportional term with a certain factor K_p. The result is a velocity correction command that increases or decreases the steepness of the controlled system behavior. This velocity error signal is applied in the integral term, thus the integral term is reacting on the error-value or steepness of the measured velocity output. The same action is followed by the

derivative term, where the K_d factor is replaced with a equal unit-based factor K_V (Kaiser, 2003).

PROPORTIONAL-INTEGRAL-DERIVATIVE POSITION LOOP OR PID

A PID controller can be used as a means of controlling one or more process variables. Moreover, as the name implies, the controller is a combination of proportional, integral and derivative adjustments which help to automatically compensate for changes in a system. Hence, the purpose of such a controller is to force the feedback to match a setpoint, such as placing an object in a desired position or maintain a desired temperature in a building.

This type of control is based on a feedback loop that measures the outcome of the process. PID returns a signal to the system based on the remaining error between the setpoint SP and measured controlled value CV. The controller exists out of an iterative loop that constantly tries to decrease this error to zero.

The proportional term applies a gain on the error to increase or decrease the speed of the controller's reaction. It is "proportionately" to the error signal. If there is no error signal, the proportional term will not have any effect.

The integral term reacts on previous output values of the system in comparison to the current output. The integral term integrates this difference over time, to eliminate the residual error. This term is depended of the proportional term. When there is no error, the integral term will cease to grow resulting in a decreasing effect of the proportional effect.

The last term is the derivative term. This term estimates the course of the error trend, based on the current rate of change in the output. This term is majorly depending on the speed of the previous controlling elements. The more rapid the other terms are changing the output signal, the larger the dampening effect is created by the derivative term (Wikipedia, n.d.) (Kaiser, 2003).

The mathematical model of the PID-controller is shown in **Error! Reference source not found.**. (Nise, 2008)

$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(t')\,dt' + K_\mathrm{d} \frac{de(t)}{dt}$$

Figure 25. Mathematical representation of PID

KEY FACTORS FOR ENCODERSELECTION

Motor is the key factor for selecting the encoder, we opted servomotor as of our encoders. Servo motors offer closed loop feedback control systems to application that require higher precision and accuracy. Motor encoder used on servo motors can be modular, incremental or absolute depending on the level of resolution and accuracy required.

ENCODERTYPES

We use incremental encoders for angular motion and position of a shaft. Basically, it converts angular motion and position of shaft into analog or digital code to identify position or motion. This type is commonly used in rotary encoders (Dynapar, 2020).

It provides excellent feedback for position and motion. It is limited by only providing change in information, so the encoder requires a reference device to calculate motion.

An Incremental encoder works as a specified amount of pulses in one rotation of the encoder. The output can be a single line of pulses(an A-channel) or two line of pulses(an A-channel and B-channel) that are offset in order to determine rotation. The phasing between the two signals is called quadrature. It consists of spindle assemble, PCB(printed circuit board) and cover. The PCB contains a sensor array that creates just two primary signals for the purpose of position and speed. A Z-channel can be provided as one pulse per revolution signal for homing and pulse count verification on the A and or B channels. This index can be gated either a or B in their various states. It can also be ungated and vary in width. (Dynapar, 2020)

We will encounter different kind of incremental coders based on our desired output.

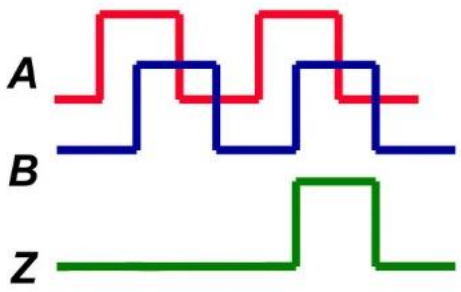

Figure 26. Output pulses of encoder signal

MOTION PROFILE

The motion profile defines the (controlled) movement a servomotor makes. This motion profile is dependent of time and indicates position, velocity and acceleration of the complete movement. The motion is normally used to determine which commands need to be send to the servo drive in order to obtain the desired result. (Collins, 2019)

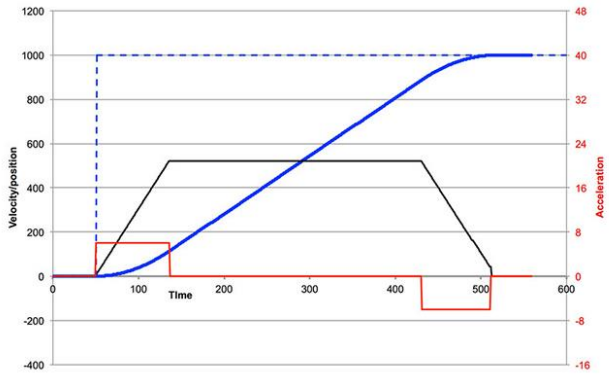

Figure 27. **Format motion profile**
Source: **https://www.motioncontroltips.com/what-is-a-motion-profile/**

The two most common types of motion profiles are the triangular motion profile and the trapezoidal motion profile.

TRIANGULAR MOTION

The triangular motion is shaped as a pyramid and indicates that there is no constant velocity. The whole movement is processed with an acceleration command and a deceleration command. The motion profile is simply the result of dividing the allowed time by two. The first period indicates the acceleration and the second period indicates the deceleration. This profile is used when a constant velocity is not necessary and the only important matter is the end position of the object (pick and place) (Collins, 2019).

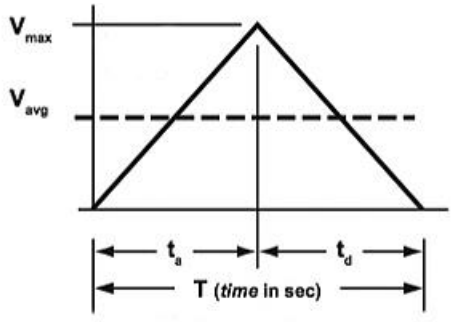

Figure 28. **Triangular motion profile**
Source: **https://www.motioncontroltips.com/what-is-a-motion-profile/**

TRAPEZOIDAL MOTION

When there is a restriction on maximum velocity, or it is important that during a period of the track the velocity is constant, a trapezoidal motion profile can be used. This profile has a steady constant velocity in between the acceleration and deceleration. The track requirements or motor requirements are important in designing the right profile. (Collins, 2019)

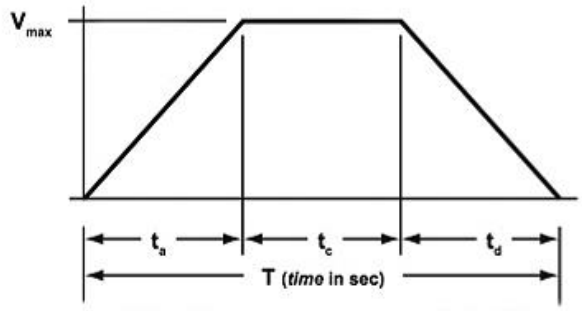

Figure 29. **Trapezoidal motion profile**
Source: **https://www.motioncontroltips.com/what-is-a-motion-profile/**

PROCESSES AND/OR PROTOCOLS AVAILABLE FOR VALIDATING A CONTROLLER

There are several ways of validation methods available to validate simulations, models and other system representations. The following processes can be used in order to validate a controller (Sargent, 2011):

[1] PARAMETER VARIABILITY
Sensitivity analysis is a technique of changing the values of the input parameters by +/-10% to determine the effect of the model's behavior or the characteristics of the output.

[2] COMPARISON TO OTHER MODELS
Results of the simulated model are compared to the results of the other valid models.

[3] ANIMATION
The operational behavior of the model is graphically displayed when the model is moved through time.

[4] PREDICTIVE VALIDATION
The model is used to predict or forecast the systems behavior and the comparison are made between models' behavior and the forecasted model to verify its results are the same.eg, field tests.

[5] TRACES
The traces or behaviors of different types of model are followed and its determined if the model's logic is correct and the needed accuracy is obtained.

[6] TURING TESTS
Turing tests are done by individuals who has adequate knowledge about the system, and they are asked if they are able to discriminate between the model and the modelled outputs.

[7] MULTISTAGE VALIDATION
It consists of combining the three historical method of rationalism, positive economics and empiric.

[8] INTERNAL VALIDITY
The model is replicated several numbers of times to determine the amount of stochastic variability in the model.

HOW DO LIMIT- OR ENDSWITCHES WORK AND WHAT IS THEIR FUNCTION?

Limit switches are often used as safety stops for amongst others motion hardware. They can detect the presence of objects and therefore indicate movement limits in a system. The limit switch is a combination of a mechanical moving part that activates an electrical circuit inside the switch. Their function is based on either PNP or NPN. These types of sensors are so-called 3-wire sensors. The difference between the two is the type of transistor that is used for the output (Ellis, 2012).

A PNP sensor has two wires for the power supply (normally brown and blue) and a signal cable (black) as switching wire. This black cable goes to the other hardware and switches from the negative pole of the power supply.

A NPN sensor works in a similar principle, the main difference is that the NPN sensor switches towards the positive pole of the power supply.

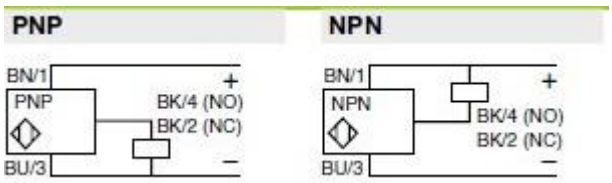

Figure 30. **PNP and NPN electrical circuits**
Source: **Google Images**

The limit switch can be normally open (NO) or normally closed (NC). When the switch is applied in NO state, the voltage supply is cut off when the switch is not pressed. In the NC state, the voltage supply is applied to the secondary hardware component when the switch is in resting position.

Figure 31. **Examples of limit switches**
Source: **Google Images**

APPENDIX C EXTENDED MODEL DESCRIPTION

OVERVIEW OF THE EPOS2 CONTROLLER CONFIGURATION

Most controlled systems are based on a feedback controller and therefore have an feedback loop implemented in its control scheme. The EPOS2 controller is based on three feedback controllers, placed in series, with each an own error signal that is fed back to the input of the controller.

The standard configuration of the closed-loop model is maintained in order to retrieve the (unbiased) natural behavior of the system. The simulation must approach the behavior of the real system as close and accurate as possible in order to get the model validated and ready for controller implementation.

The main principle of the control loop is focused on two isolated main systems, representing the controller and the system. Surrounding these two blocks are several switches, the main in- and outputs of the system and a scope for analyzing the overall behavior and system response.

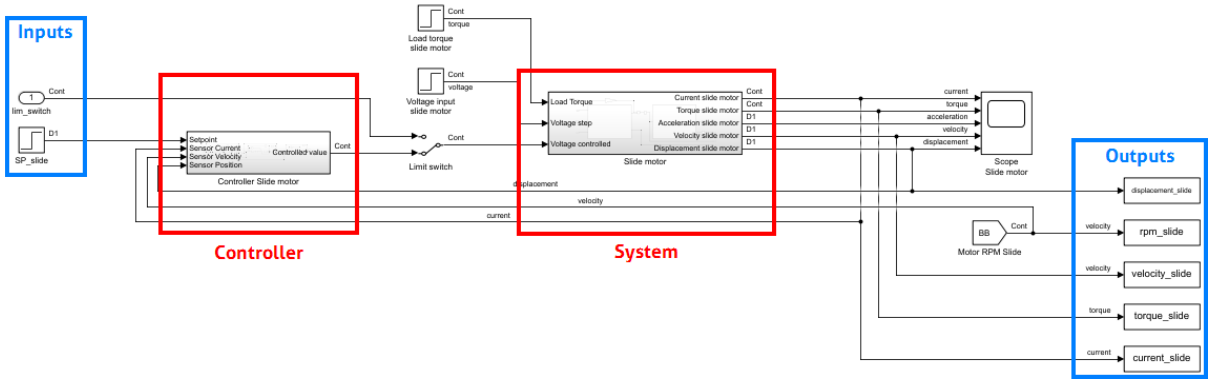

Figure 32. Overview of extended model

EXTENDED MODEL | CONTROLLER OVERVIEW

The EPOS setup uses three control loops in order to achieve the desired response. These three loops are, from inner to outer loop, respectively: *Current, position and velocity.*

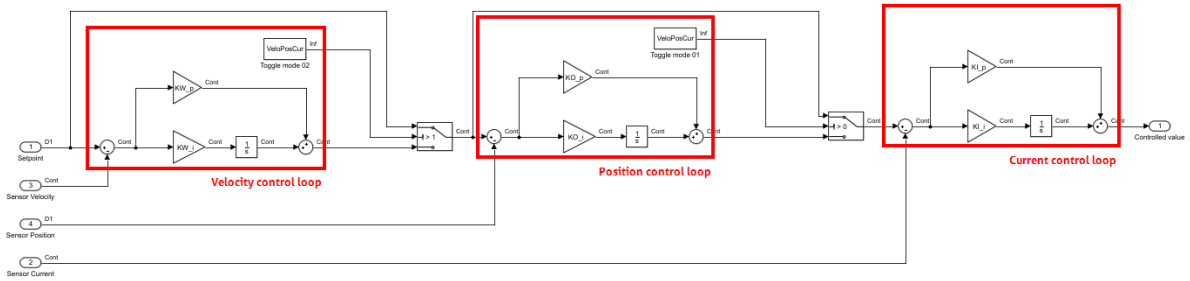

Figure 33. Overview of controller block in extended model

All three controller principles are identical to each other. The common of this controller is noted as a PI-controller, where the proportional step adds a gain to the system in order to change the location of the pole on the root locus and the integral step adds a (pure) zero to the system, that leads towards a higher systems type and thus elimination of steady state error.

EXTENDED MODEL | SYSTEM OVERVIEW

The second main system (DC-motor) is build-up by connecting all relative variables and functions, including the inductance, resistance, mass inertia and damping. In order to get a detailed view on the establishment of the magnetomotive force (MMF), the resulting torque and how this torque is driving the motor shaft is graphical represented in the model. All relations/connections between different variables and functions are low-level included. Separation of the electromechanical system by distinguishing domains leads to different subsystems for both electrics as mechanics. Between the two domains (shown as subsystems in Simulink) the presence of several interconnecting constants, and not to forget, redirection of the load towards the motor shaft, significantly improve the desired simulation results.*

** in comparison with the retrieved measurement data from the actual lock-and-key system.*

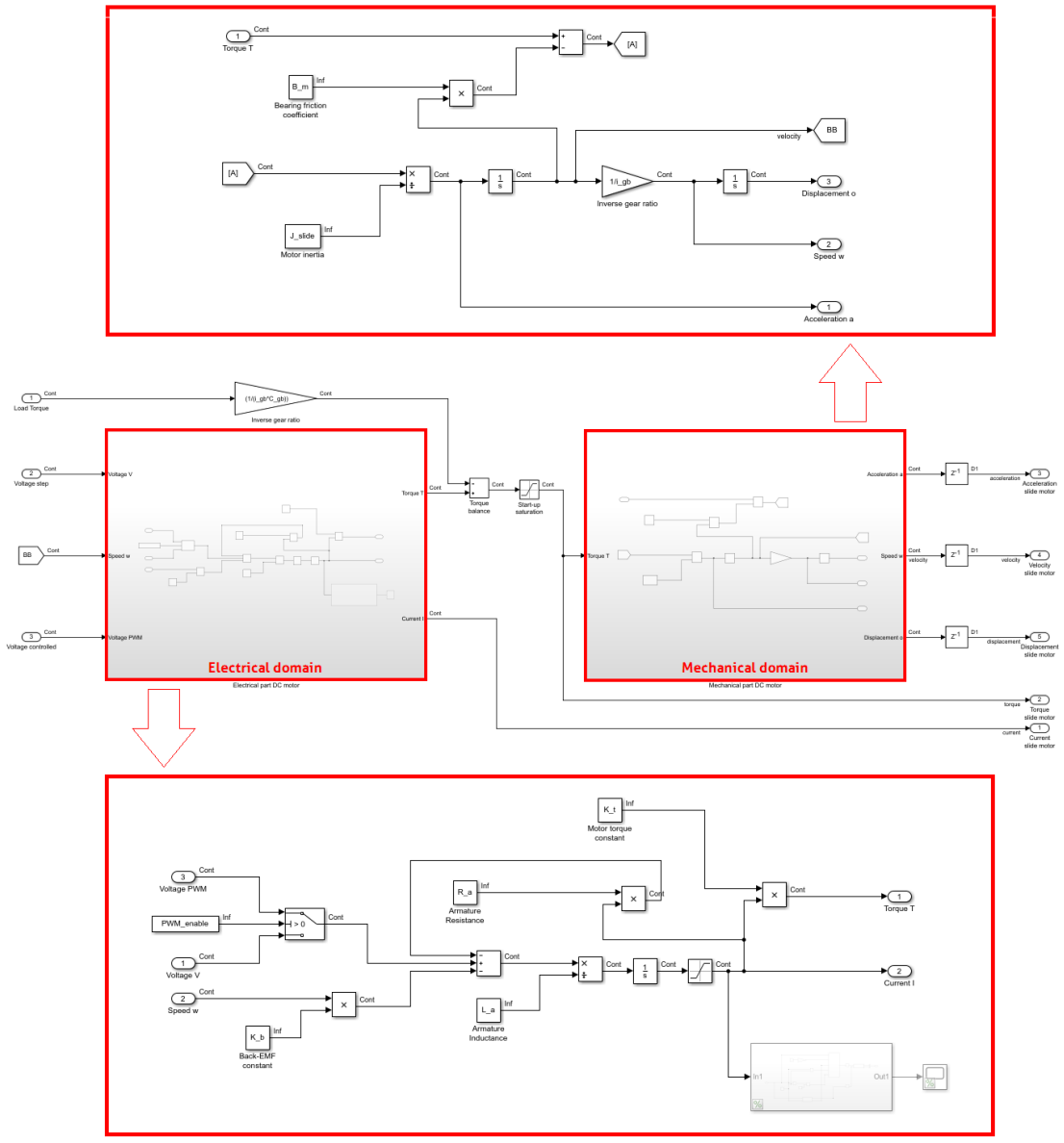

Figure 34. In-depth overview of extended model

APPENDIX D ROOT LOCUS DESCRIPTION

Firstly, it is of great importance to propose a design criterion that the controller should meet. For the key motor the following design requirements are stated:

- Settling time of less than 0.01 seconds;
- Maximum overshoot of 15%;
- No steady – state error.

The design procedure is started by presenting the open-loop transfer function (Equation 4):

$$G(s) = \frac{Output\ C(s)}{Input\ R(s)} = \frac{\theta(s)}{V(s)} = \frac{K_T}{L_a J_m S^3 + (R_a J_m + L_a B_m) S^2 + R_a B_m S + K_b K_T S}$$

Where,

$$J_m = J_{key} + J_g = 6.14 * 10^{-7}\ [kg\ m^2]$$

$$L_a = 0.000231\ [H]$$

$$R_a = 3.69\ [Ohm]$$

$$B_m = 1 * 10^{-6}\ [\frac{Nm}{rad}]$$

$$K_b = K_t = 0.0184\ [-]$$

Thus, the uncompensated closed -loop systems with the substituted parameters can be observed in Table 7. Further, the open loop root-locus figure is drawn by using MATLAB. The damping ratio line is calculated:

$$15\%OS \rightarrow \zeta = \frac{-\ln(0.15)}{\sqrt{\pi^2 + ln^2(0.15)}} = 0.5169\ [-]$$

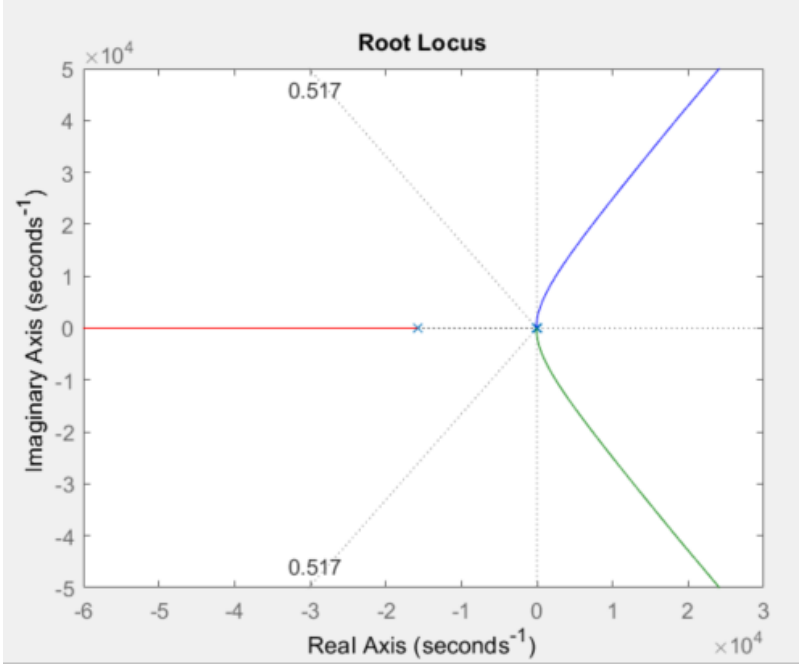

Figure 35 Uncompensated system via root locus

Dominant pole location at the damping ratio line: -71 +/- 118i. The gain K = 2.43. Additionally, the third pole at the respective gain is -1580. Thus. the second order approximation is okay (fifth fold larger than dominant poles).

PD Compensator design

From the specified settling time, the real part of the desired pole location can be computed:

$$T_s = 0.01 = \frac{4}{\zeta\omega_n} \rightarrow Real\ part = \tau_d = \zeta\omega_n = \frac{4}{0.01} = -400$$

To find the imaginary part

$$\omega d = \sigma d * \tan\theta$$

We know that,

$$\tan\theta = \frac{\omega d}{Zc - \sigma d}$$

Next, the imaginary part of the desired pole location is obtained by using the Pythagoras theorem – ω_d = 662.45 [rad/s]. This value corresponds to the damped frequency and is equal to the imaginary part of the desired pole location. Finally, the location of the desired poles is found at: -400 +/- 662.45i.

Computation for the PD controller zeros' angle is done by MATLAB, and results in an angle of 54.768 [deg]. Again, by using Pythagoras the location of the zero for the PD action is found to be at Z_c = -867.861.

Using the RL tool, the following Figure 36 is produced. As visible, the settling time requirement is reached, however, the overshoot requirement is not achieved. To account for this, the poles are moved along the root-locus, and at the gain of 0.3343 the overshoot requirement is also met. Also, as the gain is increased, the resulting settling time is reduced even more.

Finally, the PD compensator results to be $G_d(s) = 0.3343(s + 867.861)$

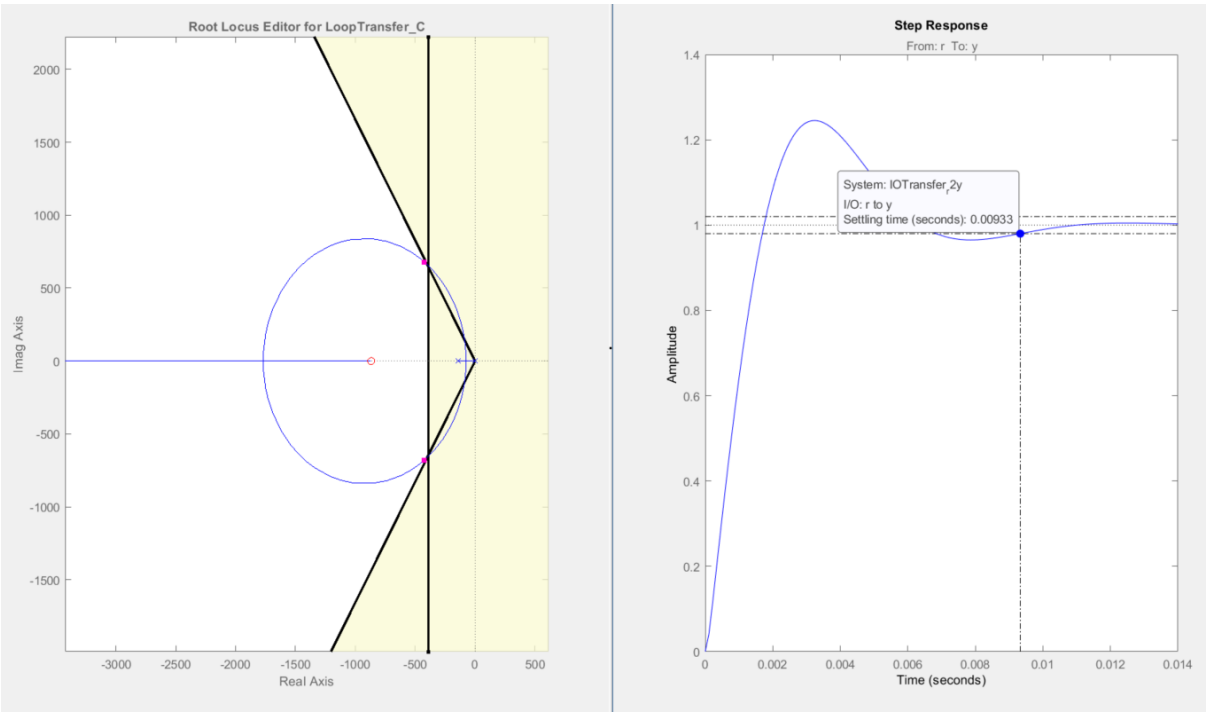

Figure 36 PD-Compensated system via root locus

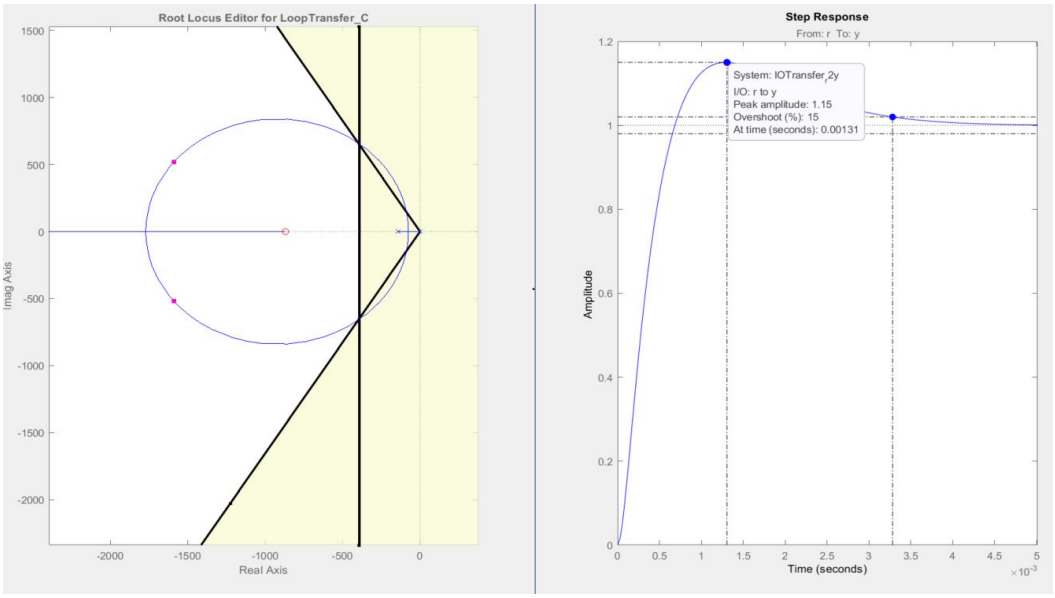

Figure 37 PD-Compensated system via root locus

PI compensator design

In order to reduce the steady-state error to zero, an integrator action is added to the previously made PD. For this an integrator and a zero close to the origin is added, thus, the compensator is:

$$G_i(s) = \frac{(s + 0.1)}{s}$$

Important to mention that the root-locus does not change due to the addition of this compensator.

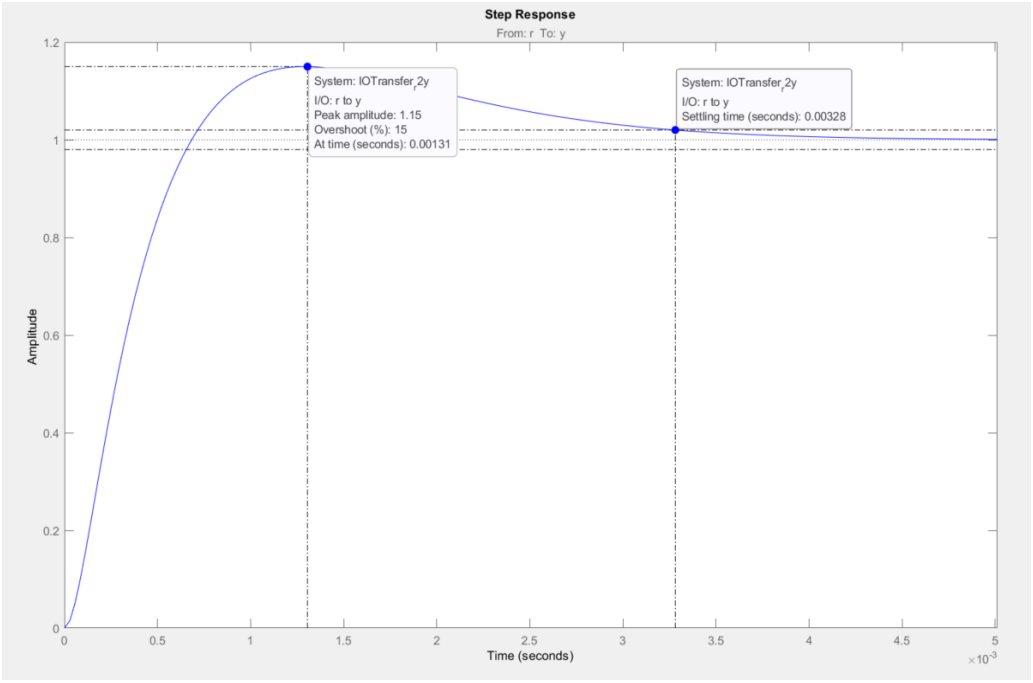

Figure 38 PI-Compensated system via root locus