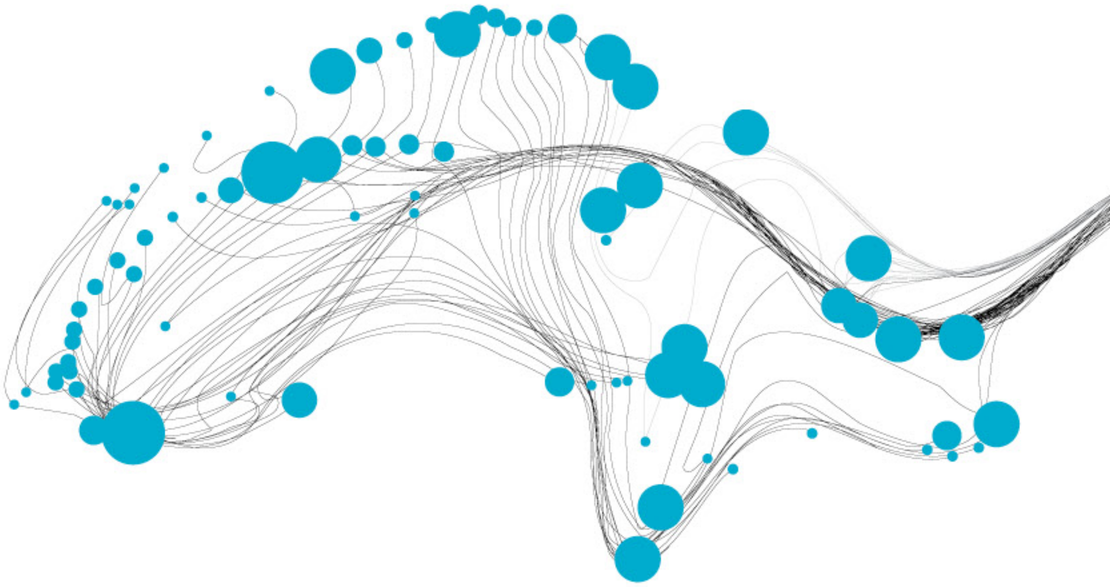
Lab Manual

Segway Project, 2024-2025



Module 6: Systems & Control

Heime Jonkers
Mostafa Selim
Mirko Mizzoni
Ahmed Ali
Teaching Assistants and Project Organizers

Antonio Franci
Teacher and Module Organizer

Contact:
h.r.jonkers@student.utwente.nl / m.s.selim@utwente.nl
m.mizzoni@utwente.nl / ahmed.ali@utwente.nl

October 30, 2024

Revision history

Revision	Year	Authors	Notes
3.1	2023-2024	M. Selim, Y. Wotte, H. Jonkers	-
3.0	2022-2023	Y. Wotte, D. Kosmas	-
2.2	2021-2022	D. Kosmas	-
2.1	2020-2021	M. van Minnen, S.T. Bunda, M. Schouten	-
2.0	2019-2020	M. Schouten	-
1.0	2016-2019	T.G. Broenink, W.Horlings	Original
1.0	2016-2019	T.G. Broenink, R.J.T. Mentink	Appendix

Acknowledgements

This manual is the result of the combined contribution of a large number of teaching and student assistants since the conceptualization of the Segway project as the practical part of Module 6 Systems & Control, at 2016.

First, a thank you to Tim Broenink and Wouter Horlings for organizing, drafting and compiling the original document, as well as maintaining it. Moreover, a thank you to Roel Mentink for the original work on the readers and appendices.

Fast-forward to 2019, a thank you to Martijn Schouten who over the course of two years, together with Mathijs van Minnen and Sebastian Bunda, further refined, updated and augmented the original work.

We would like to also thank the teachers and colleagues of CE and ESD courses, Antonio Franki, Davide Bicego, Chiara Gabellieri, Geert Folkertsma, Wesley Roozing, and Gijs Krijnen for their feedback, advice and productive discussions towards a well integrated module.

Moreover, a big thank you to Mickey Derks, Daan Pendavingh, Pascale van de Ven, Roel Mentink, Wouter Horlings and Tim Broenink for their work on the design, fabrication, maintenance and repair of the mini-Segways.

Last but not least, a thank you to all students, student assistants and colleagues who, over the years, helped shape this manual with their questions and feedback.

– *Dimitris Kosmas, Yannik Wotte*

Contents

Part I

Modelling, Characterization & Control of a DC motor

Chapter 1

Introduction

This manual covers the project and pre-projects of module 6: Systems and Control. The project consists of two parts. First, there are four pre-projects to help you carry out the motor modeling and motor measurement, and to familiarize yourself with essential control concepts. Second, the Segway control part.

At the end of this project you are able to model, characterize, and control a dynamic system, which for this system is a small Segway. Essential skills obtained are model design, parameter estimation, and controller design and tuning.

1.1 Organization

1.1.1 Planning

The pre-projects will be on Thursdays at the beginning of the quartile. Each will take three-quarters of a day, with the remaining quarter intended for journaling.

The pre-projects for the motor report are done in a total of 5 days, and they end with the submission of a motor report. You need a passing grade on this motor report to be able to participate in the Segway project.

The Segway project is done in a total of 11 days. Halfway through the project, you will have to present your Segway model to the TA's. At the end of the project, there will be a demonstration where you can show off the performance of the controller of your Segway. The Segway project ends with a soccer competition between the groups and the submission of a final project report.

Deadlines

- Friday 06 December 2024 (17:30 GMT+1): Submit motor report
- Monday 20 January 2025 (all day): Segway model presentation groups 1-51
- Friday 24 January 2025 (morning): Segway demonstration groups 1-25
- Friday 24 January 2025 (afternoon): Segway demonstration groups 26-51
- Friday 31 January 2025 (23:59 GMT+1): Submit Segway report

1.1.2 Groups and Segway Distribution

Both pre-projects and the project are done in a group of 3 students. This group should remain the same for all the pre-projects and the project, and it will be assigned to a

Segway for the project. All students should participate in the work as their contribution will be accounted for in the final grade.

A schedule will be provided on Canvas which assigns you a specific Segway. There will be one group with which you will share the Segway (your Segway buddies). There is a low chance that you will be assigned a different Segway during this project, e.g. if the Segway breaks.

At last, remember to clean your Segway at the beginning and the end of the day with the provided IPA wipes or a tiny bit of soap.

1.1.3 Journal

All students are strongly advised to keep a logbook of their work. You do not have to hand it in, and it will not be graded. A logbook will allow you to always revert back to a previous functional state, or to show to a TA all the modifications you applied before an issue appears. This simplifies debugging and makes it significantly less stressful.

This is especially useful during the project where you will be working 11 days with models and controllers. Additionally, a logbook with all your actions and notes during each measurement and simulation, makes writing a report easier.

1.2 Grading

The grade of this project will depend on two reports, one for the motor and one for the Segway part. The pre-projects will have to be completed before you can participate in the Segway project.

For your final grade, the motor measurement report will count for 30% and the project report for 70%.

To try to be fair to all students, each team member will be evaluated based on their contribution & knowledge:

- It should be mentioned in all reports the contribution of each student to the project.
- During the model presentation, each student in the group will be asked questions evaluating his/her understanding.

1.2.1 Motor Measurement

The grade for the motor measurement will be based on the report and the observations from the SA. The focus of this report should be on the following points.

- How is your bond graph model structured? Why did you include or leave out specific elements?
- How does your model compare with the actual behavior? How did you determine the parameters?
- What kind of controller did you choose to implement? How did you tune the controller?

1.2.2 Segway

The grade for the Segway control will be based on the report and the observations from the SA. The focus of the report should be on the following points.

- How was the Segway model created? Why did you include or exclude specific elements?
- How were the Segway parameters determined, and more importantly, how were these validated?
- What design choices did you make when designing the Segway controller, and why?
- How were the Segway controllers calculated and tuned?
- How does the final system perform, how did you validate this?

1.2.3 Responsibility and fraud

If you are not planning to copy the work of another group: great! Also make sure to cite all sources that you use, in your report, and to write the content you find in your own words.

If you are planning to copy the work of another group: don't. It can be hard to disentangle who committed fraud: all group members share responsibility for the work handed in. When you distribute work among group members, check each other's work, discuss the work among all group members, and make sure you understand it all.

For details on what constitutes fraud (and how seriously we take it), see as an indication (the extent of) the text below.

The beginning of (the extent of) the text below

You must submit original work.

Plagiarism, i.e. copying someone else's work without proper reference, is a serious offense which in all cases will be reported to the Examination Board. Refer to the Student Charter (2022).

In cases where a non-trivial amount of work is copied with proper reference, indicate which parts are copied and which parts are your own original work. The copied work will not be considered for grading. If you use documents or measurements of models provided during this course, you do not have to mention that.

What constitutes fraud?

When it comes down to handing in assignments, every year there are students who do not understand the borderline between, on the one hand, cooperating and discussing solutions between groups (which is allowed), and on the other, copying or sharing solutions (which is forbidden and counted as fraudulent behavior). Here are some scenarios that may help in making this distinction.

Scenario 1 Peter and Lisa are quite comfortable with programming and have pretty much finished the assignment. Mark and Wouter, on the other hand, are struggling and ask Lisa how she has solved it. Lisa, a friendly girl, shows her solution and takes them through it line by line. Mark and Wouter think they now understand and go off to create their own solution, based on what they saw. Is this allowed or not?

Verdict. No problem here, everything is in the green. It is perfectly fine and allows for Lisa to explain her solution, even very thoroughly. The important point is that in implementing it themselves and testing their own solution, Mark and Wouter are still forced to think about what is happening and gain the required understanding, though probably they will not get as much out of it as Lisa (explaining stuff to others is about the best a possible way to learn it better yourself!).

Scenario 2 The start is as in the previous case. However, while Mark and Wouter implement their own solution, inspired by that of Lisa, some error crops up that they do not understand. Lisa has left by now; after they mail her, still trying to be helpful she sends them her solution for them to inspect. They inspect it so closely that in the end, their solution is indistinguishable from Peter and Lisa's, except for the choice of some variable names and the comments they added themselves. Is this allowed or not?

Verdict. This is now a case of fraud. Three are at fault: Lisa for enabling fraud by sending her files (even if it was meant as a friendly gesture) and Mark and Wouter for copying the code. Peter was not involved, developed his own solution (together with Lisa), and is innocent.

Scenario 3 Alexandra and Nahuel are not finished, and the deadline is very close. The same holds for Simon and Jaco. On the Friday night train home, Jaco and Nahuel meet and during the 2-hour train ride work it out together. They type in the same solution and hand it in on behalf of their groups. Is this allowed or not?

Verdict. This is also a case of fraud. Actually, there are two problems here. The first is that both Nahuel and Jaco handed in code on behalf of their groups that had been developed by them alone, without their partners. This is unwise and against the spirit of the assignment (Alexandra and Simon also need to master this stuff!) but essentially undetectable and not fraudulent. The second problem is that the solution was developed, and shared, in collaboration between two groups; this is definitely forbidden. All four students are culpable; Alexandra and Simon cannot hide behind the fact that they did not partake in the collaboration, as they were apparently happy enough to have their name on the solutions and pretend they worked on it, too.

Note that we are not on a witch-hunt here: let us stress again that cooperating and discussing assignments are OK, even encouraged; it is at the point where you start copying or duplicating pieces of code that you cross the border.

Chapter 2

Pre-project: Basics

2.1 Introduction

The goal of this pre-project is to familiarise you with the basics of modeling, simulating, identifying, and controlling a real physical system. The purpose is also to establish initial contact with the software 20-sim.

Here we focus on the Segway-motors. First, you will describe the dynamic behavior of the *electrical* and *mechanical* side of the DC motors with an ideal physical model (IPM), and simulate the combined system in 20-sim. Then you will implement a basic controller for the wheel angles. But do not worry, this manual will guide you through the process by a series of questions & commands.

Make sure to log your answers and steps in your logbook (see e.g. subsection 1.1.3). It will help you and us with debugging, when something goes wrong.

If you want to make sure you completed this pre-project correctly, you can show your work to an SA at the end! Some of your work will be important for the next pre-projects, and the motor report hand-in at the end after the third pre-project.

2.2 20-sim

20-sim is the modeling and simulation software used in this module. It is unique in the fact that it can model bondgraphs, but for this first pre-project we will be working with the ideal physical models (IPMs) from the Iconic Diagrams library. For instructions on how to install 20-sim see Canvas. Furthermore, a 20-sim guide is also available on Canvas - this guide contains a getting started section. You can always refer back to this guide when looking for an explanation of specific features.

Assignment: *Install 20-sim (and for later, 20-sim4C).*

Assignment: *Familiarize yourself with a few features of 20-sim using the 20-sim guide. Where can you find the components for IPMs? How do you connect components? How do you plot specific variables after a simulation, and how can you make the plot readable? Are there useful shortcuts?*

Assignment: *How to access the "inside" of components and change parameters? What programming language is used here? Can you find information on it in the 20-sim guide or online?*

Best practices

When creating your model, the "best practices" as shown in the 20-sim manual will allow you to keep an overview. In summary: when applicable, make sure your model is annotated with the different domains, that the units are correct, and that all parts have the correct names. You don't need to put the different parts of the model in different sub-blocks (you will learn more about these later). But you should clearly show the separation between the motor, the electrical environment, and the mechanical environment. In short, make your model as easy to understand as possible in a single view.

2.3 The System

The system that you will model is the electric motor, which drives the wheel that is connected to the shaft of the motor. A graphical overview of this system is given in Figure 2.1.

When looking at this system you can identify 3 different parts:

- The electrical power supply
- The electric motor
- The shaft and wheel

When making a model of this system, keep these parts separate.

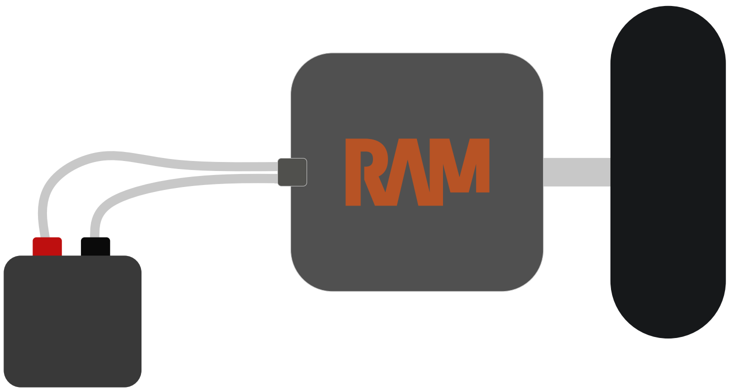


Figure 2.1: The system to model, an electric motor driving the wheel of a Segway.

Significant behavior

When using a model for a problem, the model should be as simple as possible, but also as complex as necessary. Thus you need to investigate the effects of the various components, to see if they contribute to the behaviour you are interested in.

2.3.1 First model

You start the model of the system with a few simple IPMs of the components in the system. Later in this pre-project you will replace these with more complete models, which provide a more accurate representation of the actual behaviour. Most of you have not modeled mechanical systems yet, so the mechanical side will be kept as simple as possible. You will learn more about mechanical modeling in ESD during week 2 of the module.

For this first version, we make strong simplifying assumptions: Assume that all components are 100% efficient and that the shaft and wheel are one component. 20-sim requires an electric ground to be connected somewhere in your circuit as a reference.

Assignment: *What domains can you identify in this system and what IPM elements do you need from these domains, to model the 3 components?*

To make sure you model a semi-realistic system, we provide a few (wrong) system parameters:

- The power supply outputs 24 V.
- The motor has a torque constant (K_t) of $0.6\,\mathrm{N\,m\,A^{-1}}$.
- The wheel has a radius of 5 cm.
- The wheel and shaft have a rotational inertia of $0.1\,\mathrm{kg\,m^2}$.

Assignment: *Make an ideal model of this system using the IPM elements and the parameters above. Do all parameters affect the model? Why / why not?*

Assignment: *How do you need to manipulate signals within your 20-sim file to find the distance that a point on a wheel covers over the simulation?*

Assignment: *Perform a simulation where you plot against time the electric current and the distance that a point on the wheel covers. Do the results seem realistic? Argue about possible short-comings of the model.*

2.3.2 A first transfer function

To get a mathematical description of this system we can determine a transfer function using 20-sim. This can be done with the model linearization tool, found under *Tools>Frequency Domain Toolbox>Model Linearization*. For instructions on how to use this tool please refer to the 20-sim guide.

Assignment: *Generate the transfer functions from voltage to current, and from voltage to wheel angle.*

Assignment: *Is this the transfer function that you expected? How does it relate to your observations when simulating the system?*

2.4 Model Organization

The first model you created is ideal. This means that many of the non-ideal effects that are present in a real system are not represented. To make the model more realistic we will create a more detailed implementation of each component. In order to make a comparison between the ideal elements and the new realistic versions we will use the submodel feature in 20-sim.

Therefore, we want to keep the current ideal components and add a *new implementation* for the new more detailed components. To do so, each component's model will be in a separate submodel. To split the current model into submodels, select the components in the model that you want to group; right-click on the selected components; and click *implode.*

Assignment: *Look up the implode-feature in the 20-sim guide.*

Assignment: *Separate your model into the three specified submodels: the electrical power supply, the electric motor, the shaft and the wheel.*

Assignment: *Look up the implementations feature in the 20-sim guide.*

Assignment: *Create a separate, new implementation for the submodel of the electric motor, and the submodel of the shaft and wheel.*

In this pre-project you are going to switch between different implementations. So make sure you use them. Be careful: different safe files for each model will not allow you to combine implementations at the end.

Now we will evaluate how changing submodels for the electric motor affects the system's behaviour.

2.5 Electric Motor

Switching on the systems immediately results in an immense electric current. This is because the current motor model has no internal resistance or inductance, which is not very likely. This section is about creating a new implementation for the electric motor that includes both effects. For the moment, use the following (wrong) parameters:

- The motor coil has a resistance of $2\,\Omega$.
- The motor coil has an inductance of $160\,\text{mH}$.

Assignment: *Improve your motor model to include winding inductance and resistance. What effect does this have on the transfer function of your system? And what effect does it have on the step response? Briefly explain your observations.*

2.5.1 Voltage versus angular velocity plot

Every domain has different variables of interest and there are different ways to plot these. By default, 20-sim puts time on the x-axis, but by selecting different variables than time, the relation between variables of interest can be analysed. This can reveal the properties of these components.

You can plot different variables on the x-axis, using the plot settings. More information about different plot options can be found in the 20-sim guide.

To illustrate this we will be looking at the back-EMF of the motor.

Assignment: *Create a new plot of the voltage over the ideal DC motor element against* *the angular velocity of the motor. Is this what you expect?*

2.5.2 The wheel

So far you have modelled the mechanical side by an ideal inertia for the wheel. In reality, there is friction, even for a freely rotating wheel. In this case, friction occurs between the motor shaft (to which the wheel is attached) and the bearings constraining it.

Assignment: *Make a new implementation for the mechanical side in which you include friction between the rod and the casing of the motor. Use the value $b = 0.1\frac{Nms}{rad}$ for the damping coefficient.*

Assignment: *Again observe the step-function of the system: plot the angular velocity against time, for the case with and without damping. What changed, and why?*

(Optional) Assignment: *Can you find an explicit relation between the final (i.e. steady state) angular velocity, battery voltage, and damping coefficient?*

2.5.3 A full transfer function

Now that you have a more realistic description of the full motor system, let's find its transfer function.

Assignment: *Find the transfer function of the combined system, with the output being the wheel angle. Again make use of the Model linearization tool.*

Assignment: *Compare this transfer function to the first transfer function. What are the biggest qualitative changes?*

Assignment: *Consider the transfer function with angular velocity as the output. At what value of the complex variable s does the transfer function give the ratio between steady state angular velocity and battery voltage? Does this fit with your prior simulations?*

2.6 Proportional Controller - Simulation

In this section, you will implement a first controller for the angle of the motor. Several strategies are possible for this: for now, we want to implement a proportional controller in a standard feedback loop, see figure Figure 2.2. If you have time in the end, you can try to take it a little further than just proportional control (e.g. with a proportional-derivative-integral (PID) controller). However, various other control-architectures will also be treated in a later pre-project.

Assignment: *Generate a constant reference angle and implement a proportional feedback controller that adjusts the voltage based on the difference with the real wheel angle.*

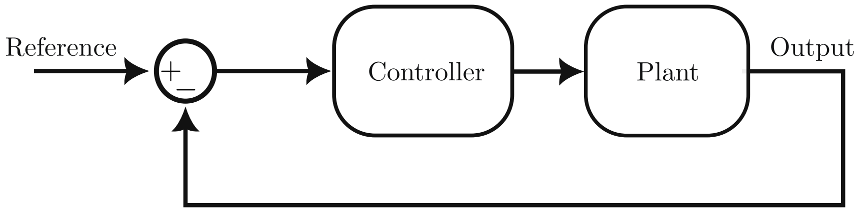


Figure 2.2: A standard feedback loop scheme.

Assignment: *Use the toolbox to determine the closed loop transfer function.*

To further analyse the system, you can use the root locus method. To generate a root locus using 20-sim, we use the model linearisation toolbox (*Tools>Frequency Domain Toolbox>Model Linearization*). Open the model linearisation toolbox and make a pole-zero plot. Next, right click on the pole-zero plot and enable the root-locus. Note that the root-locus of a transfer function (H) shows the location of the poles of the closed-loop version ($\frac{k \cdot H}{1+k \cdot H}$) of that transfer function.

Assignment: *Generate the root locus of your system and controller.*

Assignment: *What needs to happen within your root locus plot, for the system to be stable?*

We would like to stabilize the system using only proportional gain. You can use the probe in 20-sim to check what system gain is required to achieve certain pole locations. (Double click on the root locus line).

Assignment: *Can you find a gain parameter that will result in a stable system using the root locus method? Is there a gain that seems ideal for proportional control? (these are yes/no questions)*

Assignment: *Does your input voltage remain within* $\pm 24\,\mathrm{V}$ *over the control range?*

(Optional) Assignment: *Can you improve the control performance by adding other terms to the controller? What would the best choice be, in your opinion?*

(Optional) Assignment: *Does your controller work for a sinusoidal reference signal? How could you improve the performance?*

Chapter 3

Pre-project: Segway Motor

3.1 Introduction

The goal of this pre-project is to improve the electric motor model, by characterizing the real Segway DC motor. By the end of this pre-project you will know the parameters of the Segway's DC motor, and have them implemented in your model.

In this chapter, you will find a series of questions, commands, and assignments to help you along. Make sure to log your answers and steps in your logbook (see e.g. subsection 1.1.3). It will help you and us with debugging when something goes wrong.

Segway distribution

A Segway will be assigned to your group. Due to limitations on the number of Segways, you will have limited time to test on the actual Segway. By the start of this pre-project, details of the distribution schedule should be available on Canvas.

For more information on how the Segway functions, see appendix C. In case you have an issue with the Segway be sure to first check appendix D.

3.1.1 Bond Graph

You can convert your IPM model to a bond graph. With bond graphs, we can conveniently describe multi-domain dynamical physical systems, and additionally, bond graphs are supported by 20-sim.

Assignment: *Convert your IPM model of the motor and wheel to a bond graph and simulate it in 20-sim.*

Assignment: *Find the first-order differential equations that model the current through the inductor and the angular velocity of the wheel.*
Hint: *You can start by writing down the descriptions of all your bond graph components.*

(Optional) Assignment: *Use the system equations to derive the system transfer function from voltage to angular velocity. You may evaluate your derivation with the transfer function as generated by the Model Linearization toolbox.*

3.2 Parameter Estimation

Next, you need to carry out a series of experiments to estimate the parameters of the identified components. At the end of this section, your simulation should largely agree with the real side.

Assignment: *Make a list of all the parameters that you need to determine.*

To determine the parameters of the motor, you can use the tools available with the Segway. The specifics of these tools are available in Appendix section C.6. When using these tools, take into account a few things:

- You do not need to use all the tools that are available in the kit.
- You could also influence the wheel using your hands, but do not do this when the wheel is moving quickly!!
- The angle formed between the lever arm and the surface of the table is 21°. This is for the scenario where the tip of the lever rests upon the digital scale assuming the scale and Segway sit on the same plane.
- The output of the current sensor is not in A. Additionally, when using the current sensor, make sure you are using it in a region where it is linear (which it is when the current is larger than 0.3 A, see appendix C.2).
- The signal output from the uploaded submodel/controller is not in volts, you will have to convert the test signal.
- A rope with weights is limited in how far they can move, if used, make sure to keep your experiments within these limits.
- The Segway does have an angle sensor that can measure the angle of the wheels.
- The angle provided by the encoder will loop back to 0. You will need to convert this back. You can use the *encoder_to_total* block provided on Canvas.
- The output of the angle sensor is in rotations, not in radians.
- The 20-sim 4C target provides a switch to open the electrical circuit of the motor. A modeled source can be used to control this output (`PWM disable`). By default, the port is set to 0, and the electrical circuit is disconnected by setting it to 1.
- **It is not necessary to carry out all the described experiments, and you are also invited to design your own.**
- When doing measurements be sure to save your data (not just a screenshot) so that you can later analyse it. **Also be sure to use your logbook so that you know which dataset belongs to which measurement.**

You will use the software 20-sim 4C both for connecting to the segway, and afterwards for data collection.

Assignment: *Follow the video instructions in Canvas on how to upload and connect a 20-sim submodel to the Segway through 20-sim 4C.*

Figure 3.1: The setup used during experiment 1.

To determine the motor's electrical parameters, an experiment has to be performed on the motor setup. Below you will find a description of an example experiment.

Assignment: *Decide, which experiments you need to perform.*
Hint: *use the questions below. You may also draw inspiration from the four example experiments.*

In general, for every experiment try to answer the following questions:

Question: *What information do you require? What kind of experiment could provide this information?*

Question: *How does the experiment follow from your model, and what is the exact setup of the experiment?*

Question: *Does the experiment actually produce the required information?*

Question: *How can you extract the desired information from the experiment outcome? That is, how can you use this to determine model parameters?*

Based on this information you determine your model parameters.

Example experiment 1

Before starting the experiment, the direction of motion of motor is important to know for safe operation. Check direction of rotation in air. Then, a lever must be connected to the axle of the motor. The other end of this lever is resting on a scale, see figure Figure 3.1. The horizontal distance between the center of the axis of the motor and the point where the lever touches the scale is approximately 13 cm (or see Appendix C.6 for the dimensions of the lever). A voltage pulse must be applied to the motor. The motor's maximum applied voltage should not exceed 24 V. Additionally, note that the driver inputs are normalized to $[-1, 1]$. The weight is then measured by the scale. With the motor rotating in the direction of the scale, make sure that the scale is tared such that the original weight of the lever is not taken into account.

Assignment: *Collect the required data using 20-sim 4C.*

Hint: *The weight on the scale is not the only interesting quantity to measure, what is the other?*

Assignment: *Can you use the raw data to accurately estimate your parameters? Yes/No? And why?*

One of the resulting graphs has exponential behaviour (How does this follow from your system equations?). Fitting data which has exponential behaviour becomes easier when the data is plotted with a logarithmic y-axis. This is because of the following mathematical identity:

$$\log(a \cdot e^{-b \cdot x}) = \log(a) - b \cdot x \tag{3.1}$$

Where log is the natural logarithm. Because of this identity a plot of the logarithm of an exponent it will result in a line, which is way easier to fit. To calculate the logarithm in 20-sim use the Function-log block found under *Library →Signal→Block Diagram Non-linear*. Because a logarithm is only defined for positive values, you need to first make sure that the data is positive by passing it through a Function-Absolute block, also found under *Library→Signal →Block Diagram Non-linear*.

Assignment: *Determine the coefficients of the exponential decay in the current when the voltage is reduced back to zero.*

Assignment: *Use the data to identify resistance, inductance, and motor constant.*

Some of your values might not be completely stable. For example, the resistance of the motor coil is very dependent on the temperature of the motor. Have you noticed this effect?

Assignment: *Do you need to take this effect into account for your model? Why/why not?*

Assignment: *How would you model a clamped axle in your model? What effect does this have?*

Assignment: *Update your simulation with the found parameters. Do the results look as expected?*

Example experiment 2

For this measurement, a disk is attached to the motor axle. This disk is described by a moment of inertia of $\approx 4.694 \times 10^{-5}\,\mathrm{kg\,m^2}$.

A sinusoidal input voltage (you may choose the desired amplitude and frequency) is applied to the motors.

Example experiment 3

For this measurement, the same disk as in experiment 2 is attached to the motor axle. However, this time the applied input signal is a pulse, e.g. as in experiment 1.

Example experiment 4

To also model the wheel attached to the motor, the wheel is attached to the axle, all other disks are removed.

Assignment: *Can you now determine the moment of inertia for the wheel? How?*

3.3 Parameter Reflection

After you have determined the parameters of your model, you should reflect on the values of the parameters. Is there a way for you to determine if the results are realistic? (As a basic example, an MRI magnet has an inductance on the order of 10H, so your motor won't have a higher inductance than that)

Assignment: *Reflect on your parameters.*

3.4 Parameter Validation

To validate the performance of the model, compare the model to the real motor.

Assignment: *Apply an appropriate test signal to your motor and look at the angle, angular velocity, and rotations of the motor.*

In order to compare these results to the real motor you will have to connect to the real Segway. When sending this signal to the Segway, take into account the following:

- The Segway controller has a discrete time-step, make sure this is comparable to your model.
- To read imported data you can use the *DataFromFile* component, found under: *Library\Signal\Sources\DataFromFile*.
- When loading measurements, make sure your simulation time step is small enough to actually see the data! Depending on the integration method this can be done by either setting the step size or the maximum step size to a small enough value. This setting can be found in the run properties pop-up menu, under the tab of the used integration method.

For more information on the Segway hardware, see Appendix C.

Assignment: *Compare your model and the real motor using the test signal.*

When comparing the motor and the model, the main point after identifying the differences is to explain how those differences come about, and if they will be relevant for controlling the Segway. You can also correct the model using this test signal. A slight tuning of the model parameters might result in a better fit. You may use the parameter sweep/curve fitting feature of 20-sim.

Assignment: *Correct your model using the real life results, and note down the corrections used. Are the corrections realistic?*

3.5 Proportional Controller

In this section, you will implement the basic controller that you made in the last pre-project 2.6, for the angle of the motor. You now have new system parameters, so it's best to choose the control gain anew. Afterward, there will be a practical implementation on the Segway using 20-sim 4C.

Initially, we want to make the motor angle follow a reference angle in the simulation.

Assignment: *Plot the root-locus: which gain seems ideal for the proportional control, this time?*

Assignment: *Implement the ideal control gain on the real Segway. Plot the error.*

Assignment: *Compare the performance of simulation and reality, and explain the differences.*

(Optional) Assignment: *Can you make one wheel track the other? Use the angle of the second wheel as the reference angle for the first wheel.*

(Optional) Assignment: *Can you improve the control performance by adding other terms to the controller? What would the best choice be, in your opinion?*

Chapter 4

Pre-project: Control

4.1 Control

The goal of this pre-project is to improve your understanding of control, and our example will be the motor wheel system of the previous sessions. You will get to apply a few methods that you learned within the Control Engineering course up until this point, and a few practical tricks on top. If you have time at the end, you can also try to implement them on the real Segway, but that is not yet mandatory.

4.2 Preparation: Rewriting the System

Before getting into the details of a few controllers, let's write down the system transfer function from input voltage to output angle.

Assignment: *Bring your system into the form of $G(s)$ below, for appropriate n_D, n_N and coefficients $a_0, \cdots, a_{n_N}$, $b_0, \cdots, b_{n_D}$. Use any method of your choice, except cheating (see 1.2.3).*

$$G(s) = \frac{\sum_{i=0}^{n_N} a_i s^i}{\sum_{j=0}^{n_D} b_j s^j} \tag{4.1}$$

Assignment: *Implement this transfer function in 20-sim, using the linear system editor. (Take some time to look at the Frequency Domain Toolbox section in the 20-sim manual)*

Lets quickly check that everything is implemented as expected:

Assignment: *Apply a step input to the (un-controlled) system. What happens to the output?*

4.3 Phase-Lead and Phase-Lag Controller

You already designed a simple proportional controller for this system, let's take this a step further.

We will now try to control the system using a controller with one pole and one zero. This class of controllers includes PD controllers, pure I controllers, as well as lead- and lag-compensators. For now, we chose:

- A lead controller (the zero is closer to the imaginary axis than the pole)
- A lag controller (the pole is closer to the imaginary axis than the zero)

Assignment: *Write out the general form of a lead/lag controller. What is the minimum number of coefficients you need?*

Assignment: *Implement a standard feedback loop with these controllers. Use the option to create different implementations of the controller sub-model, see also section 2.4.*

4.3.1 Root locus

You will now use the root locus to analyze the stability of the system and tune the pole and zero locations of the controllers. **Important**: in 20-sim, the root-locus of a transfer function (H) shows the location of the poles of the closed-loop version ($\frac{k \cdot H}{1+k \cdot H}$) of that transfer function.

Assignment: *Use the linearisation toolbox to determine the transfer function of your system and plot the root locus of both controllers.*

Assignment: *Which of these controllers can be used to produce a stable system? What does the root locus indicate for large gains and why is this the case?*

Now that we have a rough idea of the controller's impact on the root locus, we would like to tune the performance and set pole and zero locations, as well as the system gain. Let's begin with the lead controller.

Assignment: *Choose pole and zero positions for the lead controller such that some gain sets the largest time constant in the response to below 0.2 s, but with less than 10% overshoot. Double check the impact on the final root locus.*

Next, we'd like to show a feature of the lag controller:

Assignment: *Implement a ramp input to your plant. How does the **lead** controller of the previous exercise perform on its own?*

Lets try to use both a lead and a lag controller at the same time. Use your old lead controller.

Assignment: *Choose pole and zero positions for the lag controller such that the steady state error **for the ramp input** is 5% of its value in the system with lead controller, without changing the centroid in the root locus too much.*

In reality, your voltage source is limited to an output of $\pm 24\,\mathrm{V}$. This can be implemented in simulation, by placing a signal limiter on the signal going into your source of effort.

Assignment: *Does your control input stay in the range of $\pm 24\,\mathrm{V}$? If not, how can you* *adapt your choice of control to achieve this? Do any criteria have to be changed?*

Now it is time to verify that the controller that you made is actually stabilising the system.

Assignment: *Verify that your system is stable by running a simulation and plotting the output of the control loop.*

Congratulations, you have stabilised the system!

4.4 Reference Tracking

Your controller is able to reliably smoothly reach a set-point, and even handles a ramp input fairly well, but it probably struggles with following a pre-defined path. Here, we are going to fix this issue by adding a *feed-forward* control term. You have not seen this technique before, so make sure to ask questions if something is not clear. But first, let's take a look at the issue.

Assignment: *Create a new implementation for your reference angle, and change it to a sine wave with period* $T = 1\,\mathrm{s}$ *and amplitude* $A = 1\,\mathrm{rad}$.

Assignment: *Plot both the reference angle and the real angle of the system side-by-side. How does your controller perform?*

Let's improve the performance. The idea is the following: the reference angle $\theta_R(t)$ as a function of time t is already known, and we also know the transfer function $G(s)$ from input voltage V to output angle θ, so it should be possible to construct the voltage V_{ff} that causes the desired reference angle $\theta_R(t)$. In fact, we know that $\frac{\theta(s)}{V(s)} = G(s)$. Demanding that $\theta(s) = \theta_R(s)$ (here we took the Laplace transform of $\theta_R(t)$!) then gives us the voltage

$$V_{ff}(s) = \frac{1}{G(s)}\theta_R(s)\,, \tag{4.2}$$

which should cause the system to follow $\theta_R(s)$. If we manage to construct $V_{ff}(t)$ from $V_{ff}(s)$ and feed it into the system as our control input, then this is called *feed-forward* control.

Assignment: *Sanity check: is* $G(s)$ *the open loop transfer function of the plant, or is it the open loop transfer of both plant and controller, or is it related to the closed loop transfer function?*

Assignment: *What problems would you foresee if the controller only uses feed-forward?*

We want to use both feedback and feed-forward, so we are going to add the signal $V_{ff}(t)$ to the output of our existing controller. In general (and also in 20-sim), this is problematic, because the inverted plant $\frac{1}{G(s)}$ is not causal.

Assignment: *What does it mean that* $\frac{1}{G(s)}$ *is not causal? What does this mean for* *input signals?*

20-sim demands causal transfer functions. However, it is possible to explicitly differentiate a signal that is known apriori, such as our desired sinusoidal reference. Hence, the excess of zeros in $\frac{1}{G(s)}$ can be implemented by differentiating the reference signal. Read the following assignment for more clarification.

Assignment: *Implement the transfer function* $\frac{1}{G(s)}$ *in 20-sim.*
Hint: *There are various options: for example, you can use a new signal source that represents the highest derivative of your sinusoidal input signal, and integrate it to construct all other "derivatives". Another option is to use separate signal sources for your higher derivatives. A final workaround is to convert the signal to the discrete domain (by sample and hold blocks, see also section 4.6), and to use discrete differentiation blocks rather than continuous differentiation blocks.*

Assignment: *Add the feed-forward term to the output of your controller. Has the performance improved?*

If your answer to the above question is yes, congratulations!

Assignment: *Compare the output of your controller in the case where feed-forward is present to the case where feed-forward is not used. What do you notice?*

4.5 Robustness & Disturbance Rejection

You now have a pretty good controller, even for tracking reference signals. So far, however, we have fully relied on the model being an accurate representation of reality. This section is going to investigate what happens when there are unexpected disturbances in reality, or when parameters of the real plant are different from those in your model.

4.5.1 Disturbance rejection by integral term

Let's add a disturbance force at the wheel.

Assignment: *Place a source of effort with a torque of* $0.001\,\mathrm{N\,m}$ *at the wheel*[1]*. Again use a constant reference signal. What happens?*

Assignment: *Disable the feedback control. How does the feedforward control do on its own?*

The final value should be different from the set point. We can fix this, even for unknown (but constant) disturbances, by adding an integral term.

Assignment: *Add an integral term to your controller, i.e. if your controller's transfer function is* $C(s)$*, make it* $C(s)+\frac{k_I}{s}$*, for a suitable positive gain* k_I*. Why does this increase*

[1] You can not directly use the transfer function for this, but need to use your full model. If you insist on adding a disturbance to the transfer function (e.g. a 1V difference in voltage), you are allowed to do that instead.

the correspondence of the final value to the set point?

4.5.2 Optional: Tuning the integral term

Tuning the controller such that it is again stable can be quite tedious. To do that more systematically you could use the root locus to optimize a specific parameter. For example the location of a specific pole or zero. To do this you will have to solve the root locus equation for that specific parameter.

Remember the root locus of a certain transfer function H(s), shows the location of the poles of the closed loop version of this transfer function:

$$\frac{k \cdot H}{1 + k \cdot H}$$

The locations of the poles correspond to the points where the denominator $1 + k \cdot H = 0$, which is equivalent to saying that in order to calculate the root-locus you have to calculate the solution of:

$$H = -\frac{1}{k}$$

To calculate the root locus for another parameter, e.g. the effect of adding another zero:

$$G(s) = (s + z)H(s)$$

You have to solve the root locus equation described above for z e.g.

$$z = \frac{-1 - skH(s)}{kH(s)}$$

Next invert and negate it to obtain:

$$\frac{kH(s)}{1 + ksH(s)} = -\frac{1}{z}$$

When 20-sim is asked to calculate the root-locus of $\frac{kH(s)}{1+ksH(s)}$ it will still show the location of the poles of the closed loop transfer function, but this time as a function of the zero z instead of the gain.

(Optional) Assignment: *Adjust the controller such that it is stable with the integrator. Explain why you add or move each pole or zero, using root locus or bode diagrams. Optimise at least one parameter other than the gain using the root locus method.*

Hint: You can use the model linearisation toolbox in 20-sim to calculate your transfer function!

Hint: It is not possible in 20-sim to have a transfer function with more poles as zero, however, there is possible to compute a pure derivative using the derivative block.

Note that the open loop transfer function simply consists of the multiple of both transfer functions. Therefore you don't have to use the linearisation toolbox. This allows you to quickly determine the response of the system.

4.5.3 Robustness

Lets quickly check how robust the different control terms are to changes in the parameters of our system.

Assignment: *Disable the feedback control, remove the disturbance force. How does the feedforward control do on its own?*

Assignment: *Make a new implementation of the electrical side, and change the resistance to 1.1 times its current value.*

Assignment: *How does the feed-forward control perform now?*

You have seen that one part of your controller is not at all robust to changes in the parameters. Let's put the feedback controller back into place and see if it still works.

Assignment: *Enable the feedback control, how does the combined controller perform?*

Assignment: *Disable the feedforward control. Does the feedback controller perform better while left alone?*

4.6 Discrete Controller

The controller you made is continuous. However, controllers are typically discrete since the controller can not produce an arbitrary continuous reference. As a final step, and to be more realistic, we are going to discretize the transfer function of our controller. Since our plant represents a physical system, we will keep using a continuous transfer function for that part of our simulation.

Assignment: *Use the 20-sim implode function to make a separate submodel for both the model and controller.*

We are now going to discretize the transfer functions of the controller. To do so, open the linear system editor and click on the button with $s \leftrightarrow z$. We will use the Forward Euler Transformation Method and the Frequency at which our digital controller is updating is 2000 Hz.

If we want to connect our discrete controller to the analog model, we need to include a discretization step to turn the continuous input signals into discrete signals (i.e. "analog to digital"), and a further "digital to analog" step turning the discrete output signals into continuous signals. In our case, the first step will be done with a *Sample* block and the second one will be done with a *Hold* block.

When simulating the controller and the system we need to again specify at which frequency our controller is updating, such that this can be taken into account during simulation. This is done under run properties and then discrete system (see Figure 4.1)

Assignment: *Transform the continuous transfer function of your controller into a discrete one*

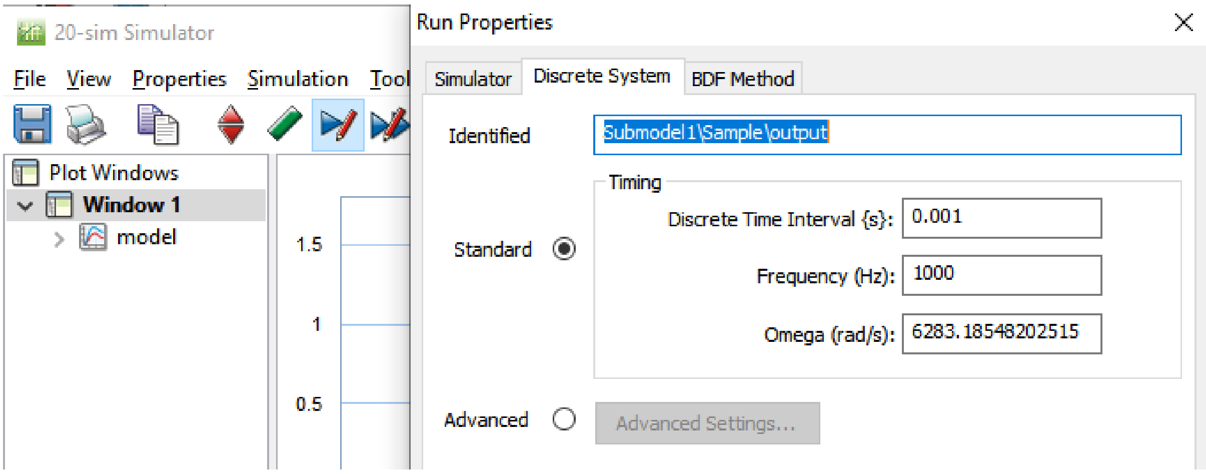


Figure 4.1: Location at which to set the simulation frequency

Assignment: *Add the analog to digital and the digital to analog conversion steps to your model*

Assignment: *Simulate the system and verify that the system is still stable.*

In many cases, the maximum frequency at which the controller can update is limited by the available processing power. Try to simulate the system.

Assignment: *Simulate the system with* 1000 Hz *and* 100 Hz *update frequency and check if the system is still stable.*

Chapter 5

Motor report

5.1 Introduction

Before you get started on modeling and controlling the Segway, you need to perform a final assignment with the Segway motor, and to deliver a report on your work of the first three pre-projects. Mainly on the methods and results of modeling, characterization, and control of the Segway motor.

5.1.1 Grading

As mentioned in the introduction of chapter 1, this lab report will account for 30% of your project grade. This grade is based on a short (4 pages) report together with the simulation files. More specifics on what to hand in can be found in section 5.3.

For more information on how the reports are graded see appendix E and F.

5.2 Segway Motor: Final Goal

As a last entertaining challenge, regarding the Segway motor, you need to perform the following assignment:

Note: This lab requires the completion of all the previous pre-projects.

Assignment: *Make one wheel follow the position of the other wheel: Generate a transfer function that can be used to design your controller. Don't forget that the PWM signal is not volts and the angle sensor output is not in radians.*

Assignment: *Upload and test your controller in the real Segway.*

5.3 What to Report

5.3.1 Modeling

For this section you will need to report at least the following:

- Your finished motor model, including an explanation.
- The decisions to or to not, neglect certain model components.

5.3.2 Characterization

For this section you will need to report at least the following:

- A list of the experiments performed, with a short description.
- What parameters were determined with each experiment?
- Show a table with the final motor parameters.
- Hand in your validation simulation. To hand in your simulation hand in a 20-sim file that when opened can immediately be simulated and will show a correctly formatted validation. Refer to the 20-sim guide for formatting.

5.3.3 Control

For this section you will need to report at least the following:

- Your design, calculations, and results of your best-performing controller, including explanation.
- All the tuning decisions made.
- A comparison between the real and simulated controller.

5.3.4 Submission files

The report file will have to be handed in as **.PDF** and should be named:**groupXX_motor_lab.pdf**. The same goes for the 20-sim files, name them: **groupXX_motor_validation.emz** and **groupXX_controller_validation.emz**.

When handing in 20-sim files that contain data from the file block, please save them in the *.emz* format. Please do not hand your data as an archive (i.e. as a .zip). For more information on how the reports are graded see Appendices E and F.

Part II

Mini-Segway Project

Chapter 6

Pre-project: Sensor Fusion and Advanced Control

6.1 Introduction

During this pre-project you will get to control an inverted pendulum using pole placement and LQR control. [1] The inverted pendulum is a nonlinear system, which characterizes a large part of the behavior of the real segway. Here, we provide a 20-sim model of it. For control design, you will need a linearized state-space representation, which is your first assignment. Afterwards, you will move on to building a full state estimator that uses the gyroscope and angle encoder signals to measure the angle of the pendulum. You are then in a position to apply the controllers mentioned above.

In this manual, you will find a series of questions/commands to guide you through this process. You do not have to hand in your answers, but you should note them down in your journal. When asking questions, you should be able to show your notes.

6.2 System

The motor of the pendulum is controlled using a motor controller that contains a current sensor. The angle of the pendulum is measured using an encoder and the rotation speed is measured using a gyro, which both have a limited bandwidth and contain noise. We will start with a simpler model of this system called `model.emx`. Later we will use a more complicated model of the pendulum called `model_with_coil.emx`.

Assignment: *Determine the input and output of your system. Which variable do you want to control (output)? Which variable can you adjust freely, to achieve this (input)?*

Assignment: *Find the state space representation of this model using the linearisation toolbox of 20-sim.*

Assignment: *How many states does this model have? What is the physical meaning of the states in this state space?*

[1] For details, also check out the Control Engineering course on "Design in the Time Domain for MIMO Systems".

We will use the A and B matrix of this state space representation of the model to design a controller in Matlab. To move the state space from 20-sim to Matlab; click on the MATLAB button, look for the definition of the A and B matrix, and copy them to a Matlab script.

Assignment: *Use Matlab to compute the controllability matrix and check if the system formed by this A and B matrix is controllable.*

6.3 State estimator

The two sensors on the pendulum, both have their strength and weaknesses. The encoder directly measures the position of the pendulum, but its bandwidth is limited to 10 Hz. The gyro has a higher bandwidth, however, to obtain a position, its output has to be integrated. This makes the position output of the gyro sensitive to low frequent drift.

To get to a reliable and highly frequent reading, the sensors' outputs need to be combined. To combine these sensor outputs one could for example use a so-called complementary filter, which is nothing more as two filters with a transfer function with a sum of one.

Assignment: *Combine the output of the encoder and the gyro to obtain one high-frequency position output. The output of the filter can be checked using the "rotation" signal.*

Now we have a good estimate of the angle, we can start to calculate the state vector. Our state vector is a column vector. To make a 20-sim submodel output a vector you either:

1. Right click on the sub-model and go to "edit interface"
2. Click on a signal output
3. Tick the size checkbox
4. Set the number of rows

Or you can use a mux to combine several signals and then use the implode function.

Assignment: *Make a sub-model that outputs the estimated state vector that is being used in the linearisation of 20-sim.*

Before we can continue with the design of the controller, we need to verify that the state that we are calculating estimates the states correctly.

Assignment: *Let the pendulum fall over. Plot state estimates and the actual state from the bond graph and check that they match.*

6.4 Controller

Now we have verified that our state matches the actual state of the system, we can continue by designing a controller. We will start by designing a controller using pole placement. A full state controller for a system with 2 states can be found in `controller.emx`.

Assignment: *Design a full state controller in Matlab using pole placement that is stable and able to rotate the pendulum from* 0 rad *to* 0.1 rad *and back.*

The model that we were using up till now did not include the coil of the motor. We will continue by designing a controller for `model_with_coil.emx`.

Assignment: *Find the state space matrix and build the state estimator for this model*

Assignment: *Can you build a controller for this system that controls just the position using pole placement?*

We will now continue by designing a controller for this system using the LQR design method in Matlab.

Assignment: *Use the LQR design method to design a controller that does not take into account the state of the coil inside its cost function and puts very little emphasis on minimising the output.*

Assignment: *Extend the number of input states of the full state controller in 'controller.emx' to 3 and use the controller that you designed to rotate the pendulum from* 0 rad *to* 0.1 rad *and back.*

Chapter 7

Project: Segway lab

7.1 Introduction

The final and biggest part of the project is to model and control the Segway. **You will notice that the "Assignments" in this chapter provide much less guidance than in previous chapters. We encourage you to explore and experiment on your own!**

The first week will be spent on modeling, and the second week on control. The Segway model has to be shown to your supervisors in a short presentation. To test your control of the Segway, a set of challenges is defined. For more information on these challenges see subsection 7.1.4.

To create your model and controller there is some information and a series of questions have been written down in this manual. You do not have to hand in all the answers to all these questions, but it is advised to write them all down in your journal.

7.1.1 Segway model presentation

Halfway through the project, you have to present your Segway model. This will be on the 20th of January (see also Section 1.1.1). This presentation was done by a short pitch for a set of Supervisors. This pitch takes about 10 minutes and you will show off your model and will have to answer questions about this. You will also get feedback on your model which you can use for the rest of your project.

For this pitch, it is important that:

- Your Segway model is properly formatted (see the 20-sim guide).
- You have a working simulation of your Segway falling over.
- You can visualize the position/angle of the Segway.

7.1.2 Grading

As mentioned in the introduction, this part of the project will account for 70% of your project grade. This grade is based on the report together with the simulation files and measurement results. More specifics on what to hand in can be found in section 7.9. For the

For more information on how the reports are graded see Appendix E

7.1.3 Segway

A Segway will be assigned to your group. Due to limitations, you will have limited time to test on the actual Segway. Your group has been assigned measurement slots on Canvas. This means you will have to create a model to work with, in order to test your controller.

For more information on how the Segway functions, see Appendix C.

7.1.4 Challenges

You will have to complete 3 challenges:

1. Standing still with the Segway.

2. Standing still with a disturbance (either inclined floor or a push).

3. Driving from start to finish, turn around, and go back.

The last two challenges will be done on our test track: it is 2.5m long, it has a starting area, a finish area, and a line along the full length of the track. Each challenge has to be completed within the time limit (will be defined later). These challenges will take place during the last day (24th of January).

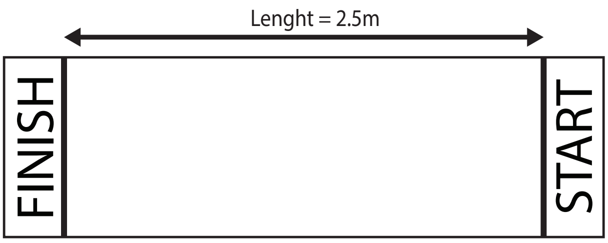


Figure 7.1: An overview of the test track

It is allowed to use the gamepad for the 3rd challenge, however not for the 1st and the 2nd challenge.

In case your Segway has passed the challenges, you qualified for the Segway soccer tournament. In a Segway soccer match your Segway will battle against another Segway. The goal of Segway soccer is to push a beach ball into your opponents' goal. Segway soccer is played on the field described in figure 7.2.

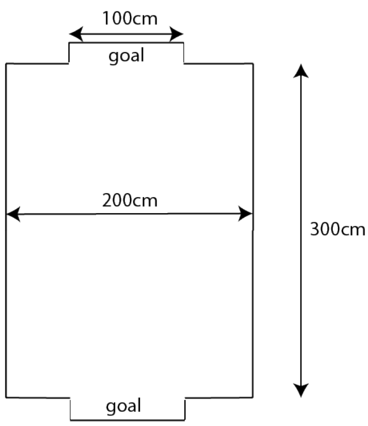


Figure 7.2: Dimensions of the Segway soccer field

7.1.5 Hints and questions

The following sections of this assignment will provide you with a set of subjects to think about, including some information, some hints and some questions. These questions do not need to be answered directly in your report. But you should note the answers and thoughts about these subjects in your Journal.

7.2 Day 1

Assignment: *Carefully read this chapter, and make a rough plan for your project approach. Try to foresee what you will need to successfully complete the challenges. Which parts can you get started on already, and which can wait for later?*

7.3 Segway Model

When creating the model of the Segway, you should take into account a few things. First and foremost, the motors provide their torque between the body of the Segway and the wheels. Second, the wheels can be assumed to be fixed in the vertical direction (contact with the floor).

When making the model you will need to make sure that all the signals that are measured by the sensors are present in the model, as you might need them for your controller later. You might need to construct extra points on the body of the Segway for this.

When linearising the Segway model for controller design, you should linearize this without all the complicated sensor effects. This allows for a simpler controller design. The multiple implementations of sub models feature could help with this.

When linearising into a state space representation of the model, 20-sim will automatically choose the states for us. Simplifying the model and changing the input and outputs of the model will change the states that 20sim selects.

As a full 3D motion is too much for this course. Fortunately, you can model the Segway as a 2D rigid body. This model can then be made in the Forward-Up-Plane. To verify that your model works correctly it is advised to make a 3D simulation in 20sim to be able to easily check if the behaviour of the Segway seems natural.

Last, to better represent the 2D rigid model of the Segway in bondgraphs, multi-bonds are preferred. You may refer to the multi-bond reader (see appendix chapter B) for a quick refresher. For a more in-depth explanation always refer to ESD Materials and Lectures in Canvas.

7.4 Angle sensor

There is no sensor that can directly measure the angle of the Segway body in relation to the world.

To determine this angle you can use both parts of the IMU. The IMU contains an accelerometer that measures the direction of gravity and a gyroscope that measures angular velocity.

Both of these sensors have their advantages and disadvantages.

Assignment: *Determine the advantages and disadvantages of using the gyroscope or* *the accelerometer.*

These disadvantages can be offset using clever tricks or by combining both sensors. For example by intelligently setting the begin value of the gyro. You can also try to combine both sensors. In that case, it is advised to look up complementary filtering.

The most important part of your angle sensor design is that you verify the angle measured by the sensors on your Segway.

Assignment: *Do an experiment to validate the performance of your angle sensor implementation. Don't forget to save the measurement results.*

Is your sensor good enough to use?

After you determine your angle you try to construct your state estimator. Again it is important to verify that your state estimator works correctly. This can be done by doing the same simulation in the simulation and on the real Segway.

Assignment: *Do an experiment to validate that your state estimator is working correctly. Don't forget to save the measurement results.*

7.5 Control

After you obtain a state estimate, full state feedback control can be used to make the Segway stand upright. But first, we will need to answer a couple of questions.

Assignment: *What is the reference state to which the Segway should be controlled for it to stand upright?*

Assignment: *How can the error between the state estimate and the reference be computed?*

Assignment: *Which error in the state should be minimised to get the Segway to stand upright as fast as possible?*

Based on the answers to these questions a full state controller can be designed.

Assignment: *Design and implement a full state controller for the Segway.*

After the Segway stands upright the next step would be to control the velocity of the Segway.

Assignment: *Which elements of the reference state vector have to be changed to make the Segway move forward and backward?*

7.6 Direction

The direction of the Segway is determined by the difference in wheel velocities, which is a signal you can measure.

Assignment: *Can you also control this signal?*

Assignment: *How would you approach this?*

Assignment: *Can you split the different wheel velocities into a forward and rotational velocity?*

7.7 Challenges and control

For the challenges and the Segway soccer tournament, you have to be able to control the Segway using the gamepad.

Assignment: *Where in your control loop will you insert the control signal?*

Assignment: *What is the maximum control signal you can give before the Segway becomes unstable?*

7.8 Safety

When working with the Segway, it might fall over. It is a good idea to stop the Segway when it has fallen. It might even be possible to let it restart when you put it upright again.

Assignment: *How can you detect that you have fallen over?*

Assignment: *How can you shut off the motors when you have fallen over? Look into switches.*

When restarting the Segway you might have to compensate for integrators in your controller that have wound up to large values.

Assignment: *Find a way to prevent windup or to reset your integrators.*

7.9 Hand in

To summarize you have to hand in a report of 6 to 8 pages (excluding your appendix) and some 20-sim files. This report will have to contain, but is not limited to:

- An overview of the Segway model and characterizations

- The design and validation of the Segway model, including sensors and parameters
- The decisions made during the modeling process
- The design and validation of the sensor fusion.
- The design and validation of the state estimator.
- The design and tuning of the controller
- The performance of the real Segway.
- The performance during the challenges.

See these specific hints sections for more content to show. Also, hand in the final model and the CSV files of your measurements. The report file will have to be handed in as **.PDF** and should be named:
groupXX_Segway_project.pdf.
. The same goes for the 20-sim files, name them:
groupXX_Segway_project.emz. When using data that is loaded from a file, please save your 20-sim file as an emz. Also please do not hand your data as an archive (i.e. as a zip), this prevents us from using the speedgrader in Canvas. For more information on how the reports are graded see Appendix E and G

Appendix A

Mechanics Reader

A.1 Dynamical system concepts

Modelling

When modelling real world systems, the goal is to create an abstract mathematical description of a real-world system. In order to do so, someone uses ideal mathematical tools to describe non-ideal behavior. The art of modelling is to create a mathematical model that is sophisticated enough to accurately describe the behavior of the system during intended operation, without making the model overly complicated.

There are many different domains, but in this course translational mechanics, rotational mechanics, electronical systems and hydraulical systems will be treated. In the next chapters, these domains will be introduced along with their corresponding ideal physical models or IPM's. These IPM's are idealized building blocks that are used to model these systems. Important to notice is that an IPM is not the same as a component.

Further reading: Chapter 1 of the lecture notes.

Further reading: Chapter 4 of the lecture notes.

A.2 Mechanical Translation

In this section, one dimensional mechanical translational systems will be explained. Mechanical elements and conventions will be explained.

The classical approach to analyse a mechanical translational system is using the laws of Newton. Although these kind of systems can be perfectly analysed using the laws of Newton, in this chapter an element based approach will be used. This element based approach makes it easier to make the step to techniques used later on in this course, like bond graphs.

Conventions

In the mechanical domain, the generalized effort variable is force. Force (F) is expressed in Newtons [N]. A Newton is the force required to accelerate a mass of 1 kg with 1 $\frac{m}{s^2}$ ($[N] = [\frac{kg \cdot m}{s^2}]$). The generalized flow is velocity. Velocity (v) is expressed in meters per second $[\frac{m}{s}]$

Individual elements in the translational domain are being connected with ideal interconnections. These can be assumed massless and infinitely stiff.

In translational systems, usually a fixed ground is indicated. This ground is the reference velocity and has got a velocity of 0 $\frac{m}{s}$.

Mass

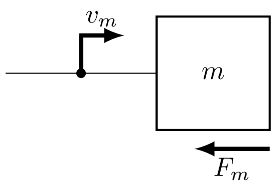


Figure A.1: Mass

A mass (Figure A.1) is a storage element that accelerates when there is an external force applied to it. This also means that it takes an external force to decelerate the mass and when there is no external force applied it will stay at a constant velocity. The element equations of the mass are:

$$F_m = m \cdot \frac{d}{dt} v_m$$

$$v_m = \frac{1}{m} \cdot \int F_m dt$$

F_m is the force exerted by the mass on the rest or the system. E.g. when the mass is being accelerated towards one side, it tries to counteract this by exerting a counter force on the rest of the system.

Note that the orientations of velocity and force are being defined in opposite directions. When defined in this way, the mass has got an ingoing power orientation. Thus, the power P_m (in Watts) that is being stored in a mass can be described by: $P_m = F_m \cdot v_m$.

Spring

A spring (Figure A.2) is a storage element that exerts a force when it is compressed or expanded. The further the spring is expanded or compressed, the more force it exerts on

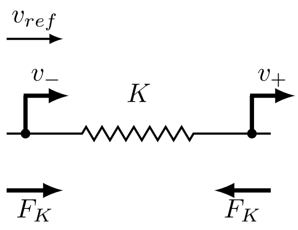


Figure A.2: Spring

the rest of the system. The element equations of a spring are:

$$F_K = K \cdot \int v_k dt$$

$$v_K = \frac{1}{K} \cdot \frac{d}{dt} F_K$$

The velocity v_K is the velocity at which the spring is expanding. This means that $v_K = v_+ - v_-$. The force F_K is the force that the spring exerts at both ends. A positive value of F_K means that it is pulling both sides inwards. When defined in this way, the spring has got an ingoing power orientation.

Damper

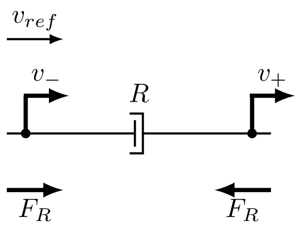


Figure A.3: Damper

A damper (Figure A.3) is a dissipative component, often caused by some kind of friction. The component value of a damper is the damping coefficient which often notated with R or c. It requires a force to keep it at a constant velocity. The element equations of a damper are:

$$F_R = R \cdot v_R$$

$$v_R = \frac{1}{R} \cdot F_R$$

The velocity v_R is the velocity at which the damper is expanding. In the example, this means that $v_R = v_+ - v_-$. The force F_R is the force that the damper exerts at both ends. A positive value of F_R means that it is pulling both sides inwards. When defined in this way, the damper has got an ingoing power orientation.

Gravity

Gravity is an example of an external force input. Gravity acts on all masses and exerts a force $F_g = m \cdot g$. m is the mass of the object in kg and g is the gravitational acceleration. The value of g is sort of constant on earth: $g \approx 9.81[\frac{m}{s^2}]$. When there is gravity acting on a system, usually there is an arrow in the schematic pointing towards the direction which the gravity is working in. Often, when a mass is moving in a horizontal direction

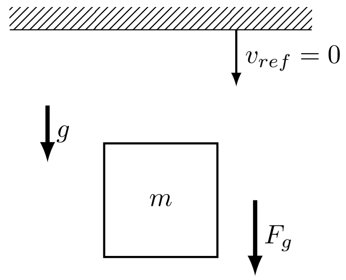


Figure A.4: Gravity on mass

without gravity acting on it, it will be drawn as a cart on wheels. No external forces are acting on this mass in this case.

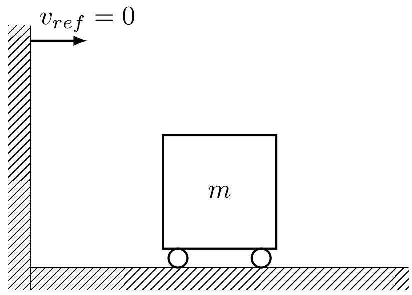


Figure A.5: Gravity acting on mass on wheels

Examples

Example 1

Question

Given the system shown in Figure A.6. Derive the equation of motion as a function of the velocity of the mass.

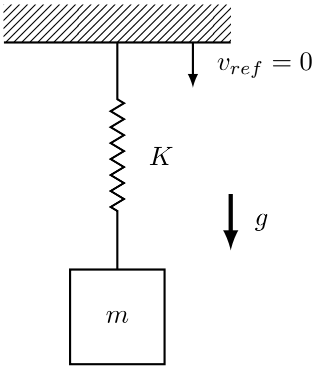


Figure A.6: Mass and spring connected to ceiling

Answer

A drawing of the system is shown in Figure A.6. Do not forget the gravity that is acting on the mass. There are two velocities: v_0 and v_1. See the schematic in Figure A.7. Note that $v_0 = v_{ref} = 0$. The velocity of the mass is the same as v_1:

$$v_m = v_1$$

The velocity of the spring is derived as follows:

$$v_K = v_1 - v_0 = v_1 - 0 = v_1$$

The forces that act on the system are shown in Figure A.8. The summation of forces at

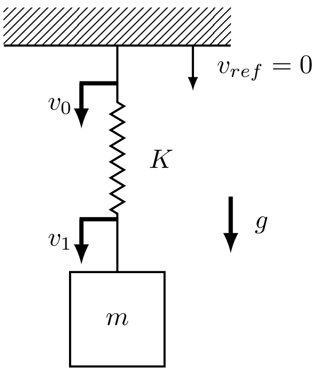


Figure A.7: Mass and spring connected to ceiling

v_1 yields:

$$F_K + F_m - F_g = 0$$

. Note: the spring is also pulling on the reference ground with force F_K, but this force is not relevant for deriving the differential equation.

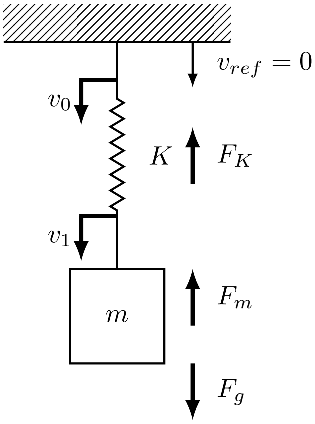


Figure A.8: Mass and spring connected to ceiling

The force equations are:

$$F_K = K \cdot \int v_K = K \cdot \int v_m$$

$$F_m = m \cdot \frac{d}{dt} v_m$$

$$F_g = m \cdot g$$

Combining these equations gives:

$$K \cdot \int v_m + m \cdot \frac{d}{dt} v_m - m \cdot g = 0$$

$$\frac{d}{dt} v_m + \frac{K}{m} \cdot \int v_m = g$$

Example 2

Question

Given the system in Figure A.9. Derive the equation of motion as a function of the velocity of the mass.

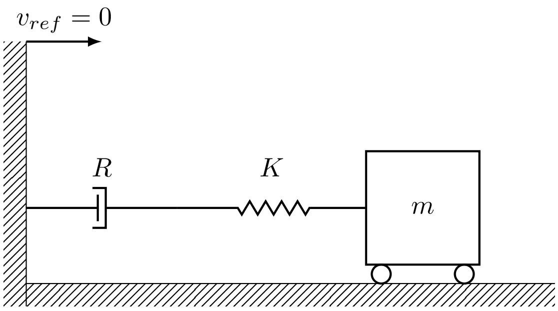


Figure A.9: Mass spring system

Answer

A drawing of the system is shown in Figure A.9. The velocities are shown in Figure A.10. The corresponding equations are:

$$v_R = v_1 - v_0 = v_1$$

$$v_K = v_2 - v_1$$

$$v_m = v_2$$

The forces are given in Figure A.11. The corresponding equations are:

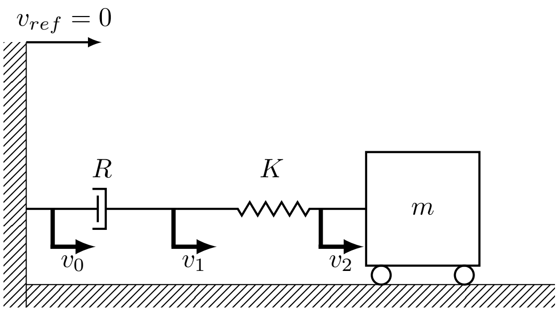


Figure A.10: Mass spring system

at v_1:

$$F_R - F_K = 0$$

at v_2:

$$F_K + F_m = 0$$

At v_1:	At v_2:	
$F_R - F_K = 0$	$F_K + F_m = 0$	
$R \cdot v_R - K \cdot \int v_K = 0$	$K \cdot \int (v_2 - v_1) + m \cdot \frac{d}{dt} v_2 = 0$	
$R \cdot v_1 - K \cdot \int (v_2 - v_1) = 0$	$K \cdot \int v_2 - K \cdot \int v_1 + m \cdot \frac{d}{dt} v_2 = 0$	Combining these
$R \cdot v_1 + K \cdot \int v_1 - K \cdot \int v_2 = 0$	$K \cdot \int v_1 = K \cdot \int v_2 + m \cdot \frac{d}{dt} v_2$	
	$v_1 = \frac{1}{K} \cdot \frac{d}{dt}(K \cdot \int v_2 + m \cdot \frac{d}{dt} v_2)$	
	$v_1 = v_2 + \frac{m}{K} \frac{d^2}{dt^2} v_2$	

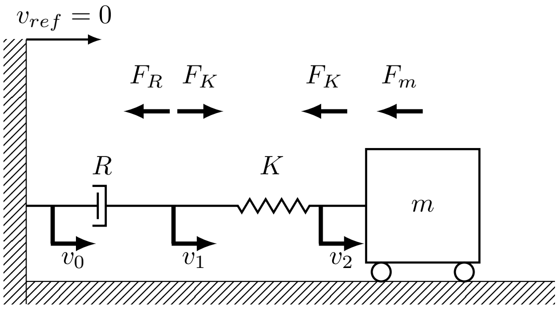


Figure A.11: Mass spring system

equations:

$$R \cdot (v_2 + \frac{m}{K} \cdot \frac{d^2}{dt^2} v_2) + K \cdot \int (v_2 + \frac{m}{K} \cdot \frac{d^2}{dt^2} v_2) - K \cdot \int v_2 = 0$$

$$\frac{Rm}{K} \cdot \frac{d^2}{dt^2} v_2 + m \cdot \frac{d}{dt} v_2 + R \cdot v_2 = 0$$

$$\frac{d^2}{dt^2} v_2 + \frac{K}{R} \cdot \frac{d}{dt} v_2 + \frac{K}{m} \cdot v_2 = 0$$

$$\frac{d^2}{dt^2} v_m + \frac{K}{R} \cdot \frac{d}{dt} v_m + \frac{K}{m} \cdot v_m = 0$$

A.3 Mechanical Rotation

In this section rotational mechanics will be explained. Rotational mechanics are similar to translational mechanics, however instead of moving in a straight line, objects are rotating around a certain point or axis. Rotational mechanics also have three basic components: a rotational damper, a rotational spring and a rotational inertia. These components are connected to each other with stiff, massless and lossless rotating axis. In this chapter, rotational elements and conventions will be explained.

Conventions

In the rotational domain, the effort variable is torque and the flow variable is angular velocity. Torque (T) is expressed in Newton-meters [Nm]. Torque is calculated using: $T = r \cdot F_{\perp}$, where r is the distance (in meters) between the point where the force is acting on and the centre where it rotates around and $F_{\perp}$ (in Newtons) is the force applied on an object perpendicular to the radius. Angular velocity (ω) is measured in radians per second $[\frac{rad}{s}]$. In other literature, sometimes also angular displacement θ $[rad]$ or angular acceleration α $[\frac{rad}{s^2}]$ is used. In the rotational domain, a positive orientation of rotation is always defined.

Rotational damper

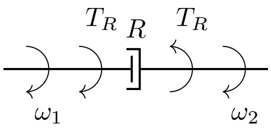


Figure A.12: Rotational damper

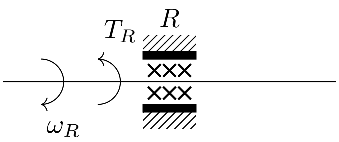


Figure A.13: Rotational friction

A rotational damper (Figure A.12 and Figure A.13) is a mechanical component that dissipative component, just like the translational damper. The element equations of the rotational damper are:

$$T_R = R \cdot \omega_R$$

$$\omega_R = \frac{1}{R} \cdot T_R$$

The velocity $\omega_R = \omega_2 - \omega_1$. The torque T_R is exerted at both sides of the damper in the opposite direction of $\omega_R l$.

When ω_0 is connected to the fixed world, often the symbol at the right is used. The element equations stated before are still valid.

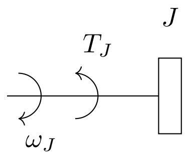


Figure A.14: Rotational inertia

Rotational inertia

A rotational inertia or flywheel (Figure A.14) is a mechanical component comparable to a mass. The element equations of the rotational inertia are:

$$T_J = J \cdot \frac{d}{dt}\omega_J$$

$$\omega_J = \frac{1}{J} \cdot \int T_J$$

The value of the inertia (J) is measured in $[kg \cdot m^2]$. When a flywheel is being accelerated in one direction, it will produce a counter-torque in the opposite direction.

How to derive the inertia of an arbitrary object

The inertia of a point mass that is rotating around a central point equals the mass multiplied by the square of the distance to the central point:

$$J = m \cdot r^2$$

However when not having a point mass this formula cannot be used. When having more point masses that rotate around one central point, the different inertia's of the separate masses can be added up:

$$J = \sum_i m_i \cdot r_i^2$$

For distributed masses, this summation can be made continuous. When having a 2D rotating around a central point, the inertia can be calculated using:

$$J = \iint \sigma \cdot r^2 dA$$

In this formula σ stands for the weight per square meter of the object and r is the distance to the central point. When having a 3D object, the inertia can be calculated using:

$$J = \iiint \rho \cdot r^2 dV$$

The variable ρ is the mass density of the object in $\frac{kg}{m^3}$ and r is the distance to the rotating axis.

Example: Calculating the inertia of a disc with uniform mass density

$$J = \iint_A \sigma \cdot r^2 dA = \int_0^r \int_0^{2\pi} \sigma r^2 r d\phi dr = \int_0^r 2\pi\sigma r^3 dr = \frac{1}{2}\pi\sigma r^4$$

To express this inertia as a function of it's mass, first the mass of a disc must be calculated using:

$$m = \iint_A \sigma dA = \int_0^r \int_0^{2\pi} \sigma r d\phi dr = \int_0^r 2\pi\sigma r dr = \pi\sigma r^2$$

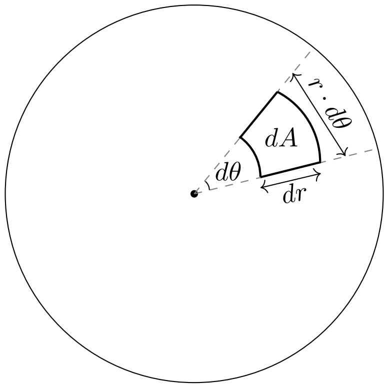


Figure A.15: Disc with uniform mass density

Now by multiplying the inertia found above with a well chosen one, it can be rewritten as follows:

$$J = \frac{1}{2}\pi\sigma r^4 \cdot \frac{m}{\pi\sigma r^2} = \frac{1}{2}mr^2$$

Rotational spring

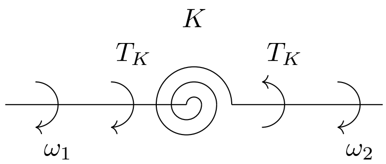


Figure A.16: Rotational spring

A rotational spring (Figure A.16) is comparable to a translational spring. The element equations are:

$$T_K = K \cdot \int \omega_K$$

$$\omega_K = \frac{1}{K} \cdot \frac{d}{dt}T_K$$

Just like the rotational damper $\omega_K = \omega_2 - \omega_1$. The torque T_K is applied at both sides of the spring in the opposite direction the spring is extended in.

Gears

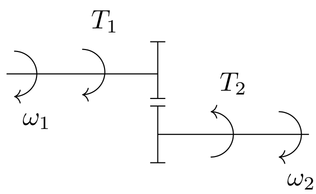


Figure A.17: Gears

Often used in rotational dynamics are gears (see Figure A.17). The element equations of gears are:

$$T_1 = -\frac{r_1}{r_2} \cdot T_2$$

$$\omega_1 = -\frac{r_2}{r_1} \cdot \omega_2$$

Note that the input- and output axle's are rotating in opposite directions. r_1 is the radius of the input gear and r_2 is the radius of the output axle.

Example

Question

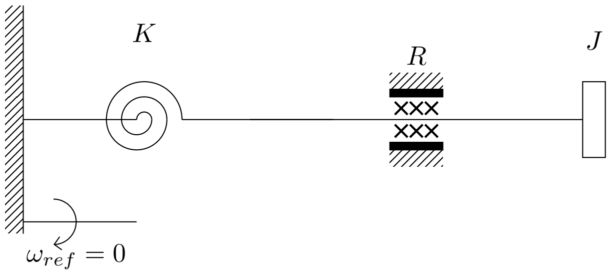


Figure A.18: Mechanical system

Given the system shown in Figure A.18. Derive an equation of motion as a function of the angular velocity of the inertia.

Answer

The drawing of the system is given in Figure A.18.

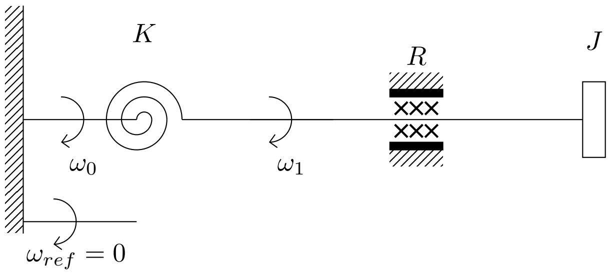


Figure A.19: Mechanical system

There are two velocities: ω_0 and ω_1, shown in Figure A.19. The corresponding equations are:
$\omega_K = \omega_1 - \omega_0 = \omega_1$
$\omega_R = \omega_1$
$\omega_J = \omega_1$

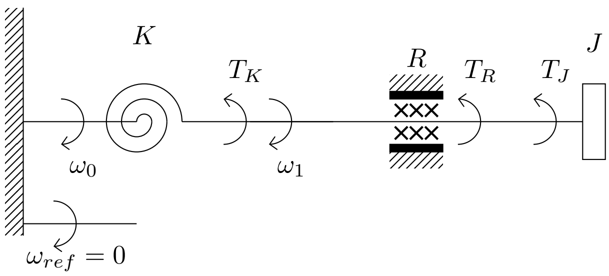


Figure A.20: Mechanical system

The torques are given in Figure A.20. The corresponding equation at ω_1 is:

$$T_K + T_R + T_J = 0$$

Combining these equations:

$$K \cdot \int \omega_K + R \cdot \omega_R + J \cdot \frac{d}{dt}\omega_J = 0$$

$$K \cdot \int \omega_1 + R \cdot \omega_1 + J \cdot \frac{d}{dt}\omega_1 = 0$$

$$\frac{d}{dt}\omega_J + \frac{R}{J} \cdot \omega_J + \frac{K}{J} \cdot \int \omega_J = 0$$

A.4 Electric domain

In this section, electrical systems will be explained. Although this domain should be prerequisite knowledge, it will be shortly be treated. First, electrical elements and conventions will be explained.

Conventions

In the electrical domain, the effort variable is voltage and the flow variable is current. Voltage (U) is expressed in Volts [V] and current (I) is expressed in Ampère's [A]. In electrical schematics, the individual elements are being connected with ideal wires. In electric circuits, usually a ground is indicated. These wires do have zero resistance. This ground is the reference voltage and is defined as zero volts.

Capacitor

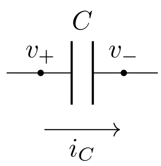


Figure A.21: RL circuit

A capacitor (Figure A.21) is a storage element that stores electrical charge. The element equations of the capacitor are:

$$I_C = C \cdot \frac{d}{dt} U_C$$
$$U_C = \frac{1}{C} \cdot \int I_C dt$$

The voltage U_C in this equation is being defined as: $U_C = v_+ - v_-$. When defined in this way, the capacitor has got an ingoing power orientation.

Inductor

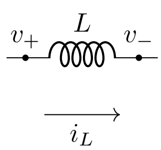


Figure A.22: RL circuit

An inductor (Figure A.22) is an electrical storage element. The element equations of the inductor are:

$$U_L = L \cdot \frac{d}{dt} I_L$$
$$I_L = \frac{1}{L} \cdot \int U_L dt$$

The voltage U_L in this equation is being defined as: $U_L = v_+ - v_-$. When defined in this way, the inductor has got an ingoing power orientation.

Resistor

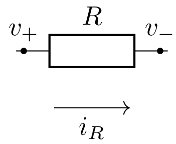


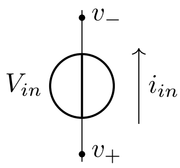
Figure A.23: RL circuit

An R-element (Figure A.23) describes an electrical resistance. The element equations of the resistor are:

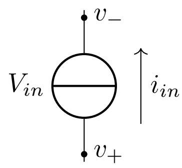
$$U_R = R \cdot I_R$$
$$I_R = \frac{1}{R} \cdot U_R$$

The voltage U_R in this equation is being defined as: $U_R = v_+ - v_-$. When defined in this way, the resistor has got an ingoing power orientation.

Sources



(a) Voltage source



(b) Current source

Figure A.24: Electrical sources

In the electrical domain there are two source elements: voltage sources (Figure A.24a) and current sources (Figure A.24b). These elements supply a defined voltage or current. For example, the voltage source supplies a given voltage whilst its current is undefined and being determined by the rest of the system. Both these elements have got an outgoing power orientation.

A.5 Hydraulics

In this section, hydraulic systems will be explained. First, hydraulic elements and conventions will be explained and afterwards a systematic method will be proposed to describe the behavior of these systems.

Conventions

In the hydraulic domain, the effort variable is pressure and the flow variable is just flow. Pressure (p) is expressed in Pascal [Pa] and flow (φ) is expressed in cubic meters per second [$\frac{m^3}{s}$]. A Pascal is defined as one newton per square meter [$\frac{N}{m^2}$].

Hydraulic capacitor

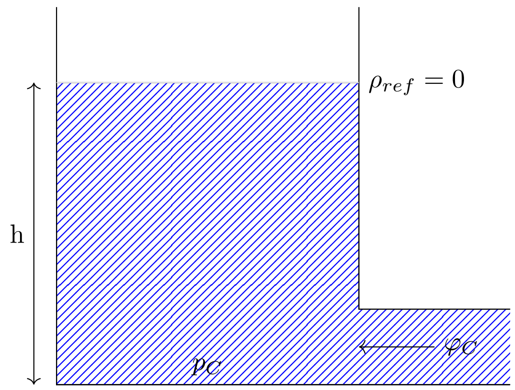


Figure A.25: Example of a hydraulic capacitor

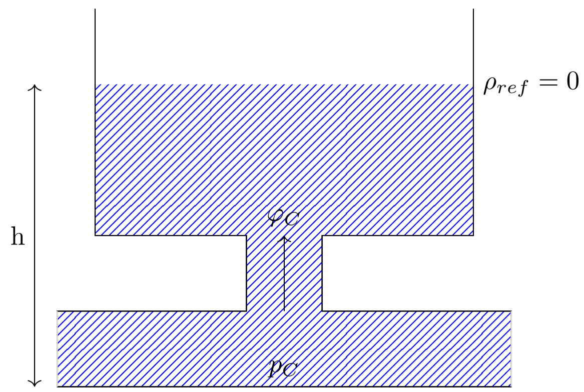


Figure A.26: Example of a hydraulic capacitor

An hydraulic capacitor (Figure A.25 and Figure A.26) is a storage element that stores hydraulic fluid. The element equations of the hydraulic capacitor are:

$$p_C = \frac{1}{C} \cdot \int \varphi_C dt$$

$$\varphi_C = C \cdot \frac{d}{dt} p_C$$

The pressure p_C is the pressure at the bottom of the container. When having an hydraulic storage with uniform surface area, the element value C can be calculated using the following equation:

$$C = \frac{A}{\rho \cdot g}$$

Where A is the surface area of the hydraulic capacitor, ρ is the density of the fluid used ($\rho \cong 1\,\mathrm{kg/l}$ for water) and g is the gravitational constant ($g \approx 9.81\,\mathrm{m/s^2}$ on earth).

Hydraulic inductance

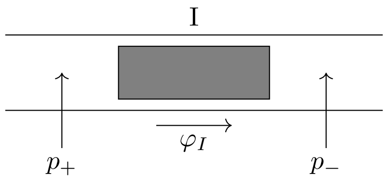


Figure A.27: Example of hydraulic inductance

An hydraulic inductance (Figure A.27) is a hydraulic storage element. Just like the inertia of a mass in the translational or rotational mechanics, moving liquids also have inertia. The element equations of the hydraulic inductance are:

$$p_I = I \cdot \frac{d}{dt}\varphi_I$$

$$\varphi_I = \frac{1}{I} \cdot \int p_I dt$$

The pressure difference p_I in this equation is being defined as: $p_I = p_+ - p_-$. When defined in this way, the inductance has got an ingoing power orientation. When having a tube with a constant cross-section area A and a given length l, the element value I can be calculated using:

$$I = \frac{\rho \cdot l}{A}$$

In this equation the variable ρ again represent the density of the fluid used ($\rho \cong 1$ in the case of water).

Hydraulic resistance

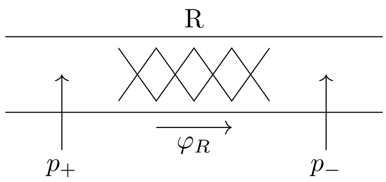


Figure A.28: Example of a hydraulic resistor

An hydraulic resistance (Figure A.28) is a dissipative hydraulic element. The element equations of the resistor are:

$$p_R = R \cdot \varphi_R$$

$$\varphi_R = \frac{1}{R} \cdot p_R$$

The pressure difference p_R in this equation is being defined as: $p_R = p_+ - p_-$. When defined in this way, the resistance has got an ingoing power orientation.

Pressure sources

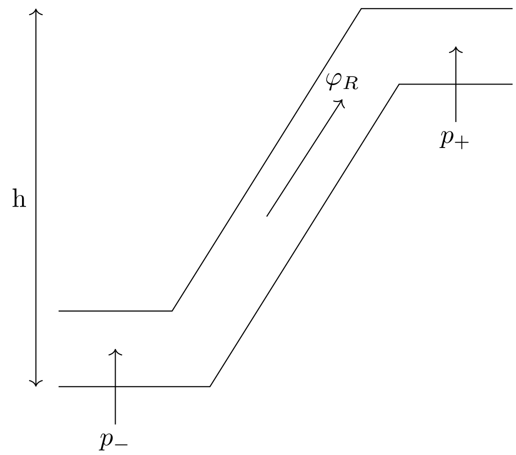


Figure A.29: Example of a pressure source

A pressure source (Figure A.29) supplies a given pressure to the system. In the hydraulical domain there are often pressure differences caused by height differences: these are caused by gravity. The pressure p_{in} caused by a certain height difference h can be calculated using:

$$p_{in} = \rho \cdot g \cdot h$$

Where ρ is the density of the liquid ($\rho \cong 1$ in the case of water) and g is the gravitational constant ($g \approx 9.81$ on earth).

The pressure difference p_{in} in this equation is being defined as: $p_{in} = p_+ - p_-$. When defined in this way, this element has got an outgoing power orientation. The flow φ_{in} is being determined by the rest of the system.

Flow source

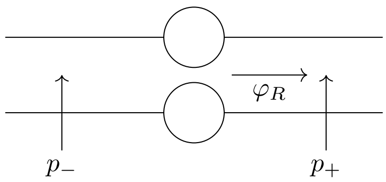


Figure A.30: Example of a flow source

A flow source (Figure A.30) supplies a given flow to the system. This element has got an outgoing power orientation. The pressure p_{in} supplied by the flow source is undefined and determined by other elements in the system, but is defined as: $p_{in} = p_+ - p_-$.

Example

Question

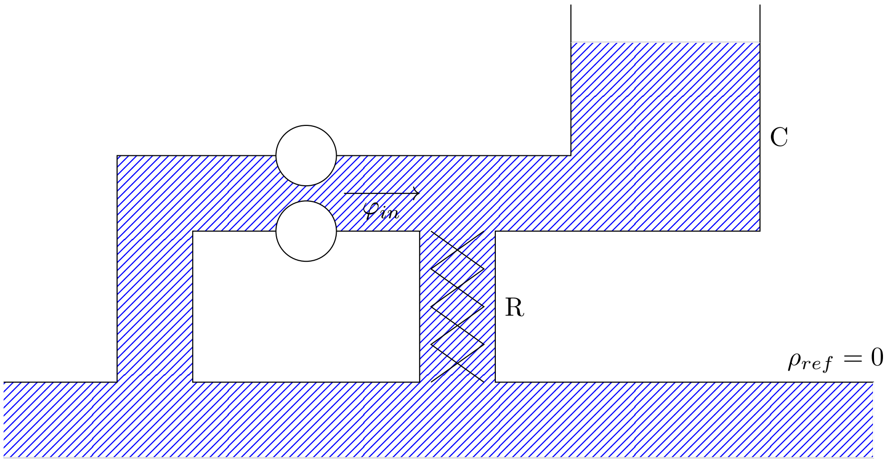


Figure A.31: Hydraulic system of example 1

Given the system shown in Figure A.31, derive a differential equation describing the behavior of the system as a function of the pressure at the bottom of the hydraulic storage tank.

Answer

A symbolic drawing is already given in Figure A.31. There is only one pressure in this

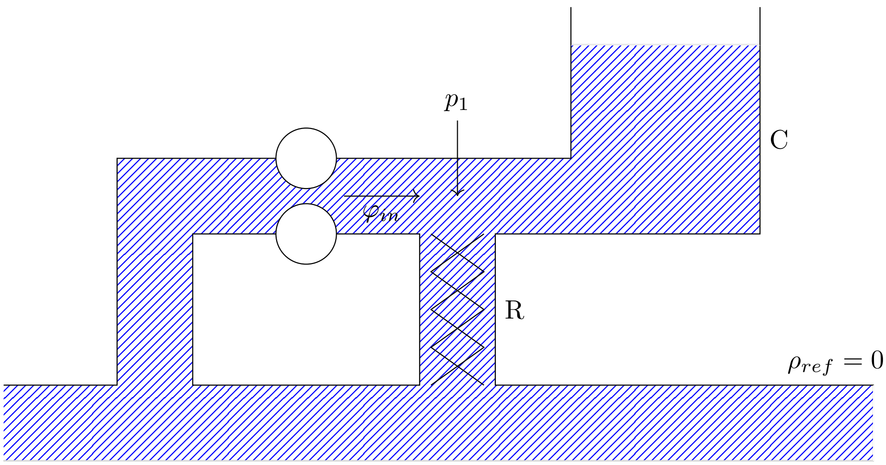


Figure A.32: Hydraulic system of example 1

system apart from the reference pressure as indicated in Figure A.32. This means the following holds: $p_R = p_C = p_{in} = p_1$. The different flows in the system are shown in Figure A.33. The summation of the flows at the junction is: $\varphi_{in} - \varphi_C - \varphi_R = 0$. Combining the equation above with the element equations, which can be expressed as functions of p_C:

$$\varphi_C = C \cdot \frac{d}{dt} p_C$$

$$\varphi_R = \frac{1}{R} \cdot p_R = \frac{1}{R} \cdot p_C$$

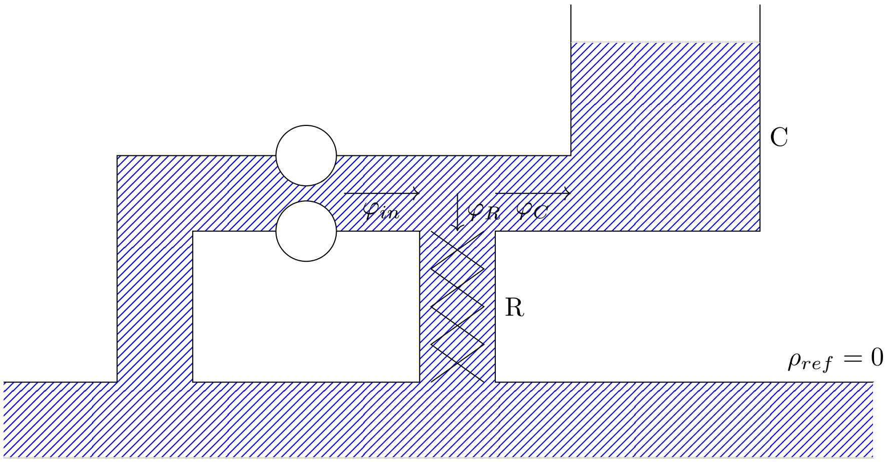


Figure A.33: Hydraulic system of example 1

Filling these in yields:

$$\varphi_{in} - C \cdot \frac{d}{dt} p_C - \frac{1}{R} \cdot p_C = 0$$

Rewriting gives the differential equation:

$$\frac{d}{dt} p_C + \frac{1}{RC} \cdot p_C = \frac{1}{C} \cdot \varphi_{in}$$

Appendix B

Multibond reader

Multibond bond graphs are often used to describe multi-dimensional systems such as 2D or 3D mechanics in a more compact way. Multibonds are the multivariable equivalent of regular bond graphs. Instead of a single bond with one effort and one flow, multibonds represent a vector of efforts and flows. Multibond elements have a square matrix as an element value, except effort an flow sources which have a vector. All equations from single bonds are also valid for multibonds, when replacing all variables with corresponding vectors and matrices, as can be seen in Table B.1. The only exception are the transformer and gyrator, where one of the transformer and gyrator matrices need to be transposed in order to guarantee power continuity.

A multibond bond graph can always be expanded into single bond notation. In order to do so, all elements in the bond graph need to be replaced with a single bond equivalent. This is easy for bonds, sources and junctions, as they need to be replaced by multiple regular single bond versions, but can be more comprehensive for the other elements.

An example would be expanding a double bond resistor with the component value $\begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}$, as can be seen in Figure B.1. Writing down the element equations of this

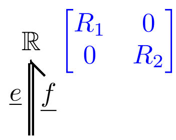


Figure B.1: Double bond resistor

resistor, one obtains:

$$\underline{e} = \mathbb{R} \cdot \underline{f}$$

$$\begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \cdot \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

$$\begin{cases} e_1 = R_1 \cdot f_1 + 0 \cdot f_2 = R_1 \cdot f_1 \\ e_2 = 0 \cdot f_1 + R_2 \cdot f_2 = R_2 \cdot f_2 \end{cases}$$

It can be seen that this is equal to the element equation of two individual resistors: one with the value R_1 and one with the value R_2, like in Figure B.2.

I and C elements have similar behaviour: when all values that are not on the diagonal are zero, the element can be replaced by multiple single-bond elements.

	Singlebond:		Multibond:	
0	$e_2 \,\lvert\, f_2$; $\xrightarrow[f_1]{e_1} \mathbf{0} \xrightarrow[f_3]{e_3}$	$e_1 = e_2 = e_3$ $f_1 = f_2 + f_3$	$\underline{e}_2 \,\Vert\, \underline{f}_2$; $\xRightarrow[\underline{f}_1]{\underline{e}_1} \underline{\mathbf{0}} \xRightarrow[\underline{f}_3]{\underline{e}_3}$	$\underline{e}_1 = \underline{e}_2 = \underline{e}_3$ $\underline{f}_1 = \underline{f}_2 + \underline{f}_3$
1	$e_2 \,\lvert\, f_2$; $\xrightarrow[f_1]{e_1} \mathbf{1} \xrightarrow[f_3]{e_3}$	$e_1 = e_2 + e_3$ $f_1 = f_2 = f_3$	$\underline{e}_2 \,\Vert\, \underline{f}_2$; $\xRightarrow[\underline{f}_1]{\underline{e}_1} \underline{\mathbf{1}} \xRightarrow[\underline{f}_3]{\underline{e}_3}$	$\underline{e}_1 = \underline{e}_2 + \underline{e}_3$ $\underline{f}_1 = \underline{f}_2 = \underline{f}_3$
Se	$\mathbf{Se} \xrightarrow[f]{e}$	$e = e_{in}$	$\mathbb{S}e \xRightarrow[\underline{f}]{\underline{e}}$	$\underline{e} = \underline{e}_{in}$
Sf	$\mathbf{Sf} \xrightarrow[f]{e}$	$f = f_{in}$	$\mathbb{S}f \xRightarrow[\underline{f}]{\underline{e}}$	$\underline{f} = \underline{f}_{in}$
R	$\mathbf{R}$; $e \,\lvert\, f$	$e = R \cdot f$	$\mathbb{R}$; $\underline{e} \,\Vert\, \underline{f}$	$\underline{e} = \mathbb{R} \cdot \underline{f}$
C	$\mathbf{C}$; $e \,\lvert\, f$	$e = \frac{1}{C} \cdot \int f dt$	$\mathbb{C}$; $\underline{e} \,\Vert\, \underline{f}$	$\underline{e} = \mathbb{C}^{-1} \cdot \int \underline{f} dt$
I	$\mathbf{I}$; $e \,\lvert\, f$	$f = \frac{1}{I} \cdot \int e dt$	$\mathbb{I}$; $\underline{e} \,\Vert\, \underline{f}$	$\underline{f} = \mathbb{I}^{-1} \cdot \int \underline{e} dt$
TF	$\xrightarrow[f_{in}]{e_{in}} \mathbf{TF} \xrightarrow[f_{out}]{e_{out}}$	$f_{out} = T \cdot f_{in}$ $e_{in} = T \cdot e_{out}$	$\xRightarrow[\underline{f}_{in}]{\underline{e}_{in}} \mathbb{TF} \xRightarrow[\underline{f}_{out}]{\underline{e}_{out}}$	$\underline{f}_{out} = \mathbb{T} \cdot \underline{f}_{in}$ $\underline{e}_{in} = \mathbb{T}^T \cdot \underline{e}_{out}$
GY	$\xrightarrow[f_{in}]{e_{in}} \mathbf{GY} \xrightarrow[f_{out}]{e_{out}}$	$e_{in} = r \cdot f_{out}$ $e_{out} = r \cdot f_{in}$	$\xRightarrow[\underline{f}_{in}]{\underline{e}_{in}} \mathbb{GY} \xRightarrow[\underline{f}_{out}]{\underline{e}_{out}}$	$\underline{e}_{out} = \mathbb{R} \cdot \underline{f}_{in}$ $\underline{e}_{in} = \mathbb{R}^T \cdot \underline{f}_{out}$

Table B.1: Multibond elements

In the case of a double resistor with the value $\begin{bmatrix} R_1 & R_x \\ R_x & R_2 \end{bmatrix}$, there are values that are not on the diagonal and not zero. When writing down the element equations of this resistor, one obtains:

$$\underline{e} = \mathbb{R} \cdot \underline{f}$$

$$\begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = \begin{bmatrix} R_1 & R_x \\ R_x & R_2 \end{bmatrix} \cdot \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

$$\begin{cases} e_1 = R_1 \cdot f_1 + R_x \cdot f_2 \\ e_2 = R_x \cdot f_1 + R_2 \cdot f_2 \end{cases}$$

Writing these equations back into a bond graph is less obvious than in the previous example. In this case it is more clear when it is first drawn as a block diagram and then converted to a bond graph. The blockdiagram is given in Figure B.4. From this diagram, it appears that this 'resistor' appears to show gyrator-behavior. However, when taking into account the symmetry of the R matrix, thus $R_{12} = R_{21}$ this element could also be constructed using a resistor and a rotating transformer. An example of this is shown in Figure B.3.

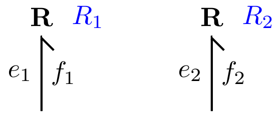


Figure B.2: Single bond representation of double bond resistor

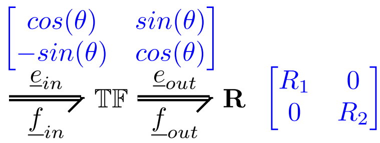


Figure B.3: A multi-bond resistor after a Transformer

When one would simplify the graph of Figure B.3 to a single element. The resulting element is a single R element with the element matrix:

$$\begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix}^T \cdot \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \cdot \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} =$$

$$\begin{bmatrix} cos^2(\theta) \cdot R1 + sin^2(\theta) \cdot R2 & cos(\theta)sin(\theta) \cdot R1 - cos(\theta)sin(\theta) \cdot R2 \\ cos(\theta)sin(\theta) \cdot R1 - cos(\theta)sin(\theta) \cdot R2 & cos^2(\theta) \cdot R2 + sin^2(\theta) \cdot R1 \end{bmatrix}$$

Only when the element matrix is not symmetric, thus $R_{12}! = R_{21}$ is a gyrator added to the system. However, this only happens when a storage element is moved through a modulated storage. For more information on this, see the 2D mechanics handout.

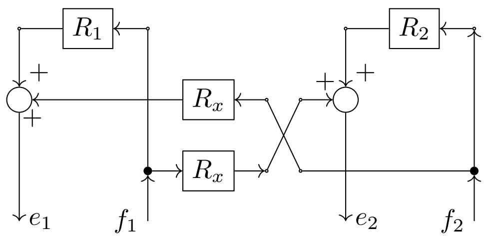


Figure B.4: Block diagram representation

Example question 1

Question

Given the bond graph in Figure B.5. Expand the bond graph into single bond notation.

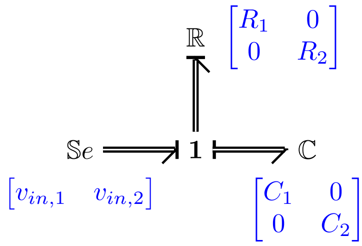


Figure B.5: Bond graph.

Solution

The multibond bond graph given in Figure B.5 can be expanded in single bond notation. The result can be found in Figure B.6.

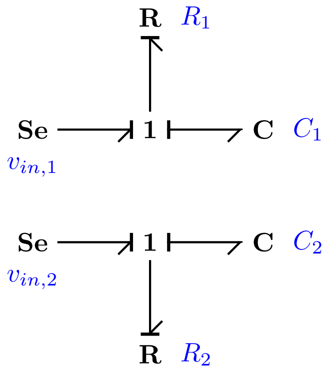


Figure B.6: Single bond graph representation of bond graph given in Figure B.5.

Example question 2

Question

Expand the transformer shown in Figure B.7 into single bond notation.

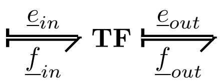


Figure B.7: Bond graph.

Solution

The element equations are:

$$\begin{cases} \underline{f}_{out} = \mathbb{T} \cdot \underline{f}_{in} \\ \underline{e}_{in} = \mathbb{T}^{\top} \cdot \underline{e}_{out} \end{cases}$$

When filling this in:

$$= \begin{cases} \begin{pmatrix} f_{out,1} \\ f_{out,2} \end{pmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \cdot \begin{pmatrix} f_{in,1} \\ f_{in,2} \end{pmatrix} \\ \begin{pmatrix} e_{in,1} \\ e_{in,2} \end{pmatrix} = \begin{bmatrix} T_{11} & T_{21} \\ T_{12} & T_{22} \end{bmatrix} \cdot \begin{pmatrix} e_{out,1} \\ e_{out,2} \end{pmatrix} \end{cases} = \begin{cases} f_{out,1} = T_{11} \cdot f_{in,1} + T_{12} \cdot f_{in,2} \\ f_{out,2} = T_{21} \cdot f_{in,1} + T_{22} \cdot f_{in,2} \\ e_{in,1} = T_{11} \cdot e_{out,1} + T_{21} \cdot e_{out,2} \\ e_{in,2} = T_{12} \cdot e_{out,1} + T_{22} \cdot e_{out,2} \end{cases}$$

When drawn into a block diagram, the result can be found in Figure B.8.

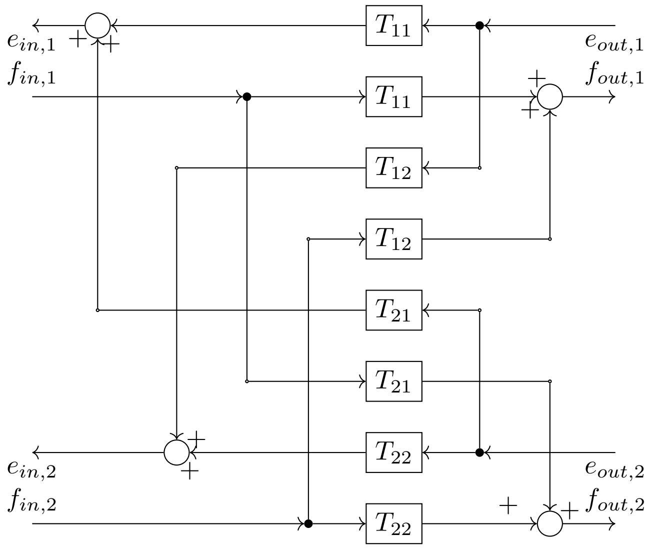


Figure B.8: Block diagram representation

This block diagram can be converted to a bond graph, see Figure B.9.

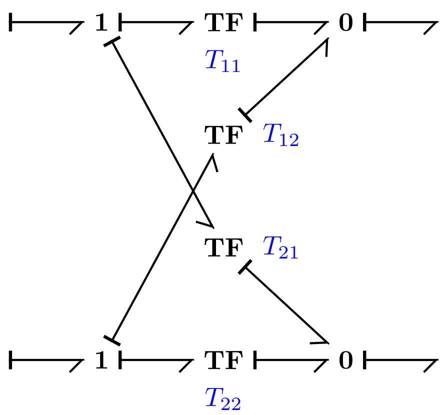


Figure B.9: Single bond graph representation of bond graph given in Figure B.7.

Appendix C

Segway Specifications

The Segway used for these labs is a custom-made platform based on a Raspberry Pi 3B. An image of this segway is shown in Figure C.1. This segway contains 2 motors, with current and angle sensors, an IMU, and a line sensor.

All of these motors and sensors are connected to the controller in such a way that they can be used in 20-sim 4C. How the signals are 20-sim4C translated to the real world, is described in the specific sections. The Segways are accessible by connecting to *ramsegwayXX.roaming.utwente.nl* where XX is the number of the segway. The frequency at which the controller on the Segway can run depends on the model but is approximately 1 kHz

With the segway you are supplied a box of measurement instruments, this box contains a measurement disk, a spring balance, and various tools to perform measurements on the segway.

C.1 Mechanical

The Segway without wheels weighs 2.120 kg. A single wheel weights 140 g.

C.2 Motors

The segway contains two brushed DC motors, controlled by two MC33926PNB motor driver chips. These motor drivers are powered by 24 V and controlled with a PWM signal. The signals from the controller are mapped from -1 to 1 for a duty cycle of -100% to 100%. Thus resulting in an effective mapping from -1 to 1, to $-24\,\mathrm{V}$ to 24 V.

For example, a signal of 0.4 will result in a duty cycle of 40% and thus an effective voltage of 9.6 V.

Take note that both motors are mirrored, thus the motors will turn in the opposite direction (forward and backward) when the same signal is applied.

Every motor has 2 sensors, a current sensor, and an angle sensor.

C.2.1 Current sensor

The motor driver outputs a current signal. This current signal is sampled by a AD7091R-5 and sent to the controller. The current sensor can only sense the magnitude of the current, not the way the current is flowing. The bandwidth of the currents sensor is also limited to 80 Hz and the current sensor does not measure currents below 0.2 A and is not linear below 0.3 A, see figure Figure C.2. After 0.3 A the current (I) can be calculated from the current sensor output (O) using the following formula:

$$I = 4.7 \cdot O - 0.09 \tag{C.1}$$

For example, both a current of $-1.7\,\mathrm{A}$ and $1.7\,\mathrm{A}$, result in the signal 0.38 in the controller. Also both a current of $0.05\,\mathrm{A}$ and a current of $0.1\,\mathrm{A}$ will result in an output of 0. When using this sensor for parameter identification, make sure it is in the linear range. For example, use a sinewave with an offset, instead of just a sine.

C.2.2 Angle sensor

There is a magnet connected to the back of the motor axle. The orientations of this magnet are measured with an AS5048B angle sensor, which measures the direction of the magnetic field. This means that after a complete revolution of the motor, the angle sensor will output the same value.

The angle of the motor is mapped in such a way that the angle of 0 to $2\,\pi$ corresponds to a signal from 0 to 1. The sampling frequency of the angle sensor is equal to the frequency at which the controller is running but has one sample delay.

For example, if the motor is at the angle π (signal 0.5) and turns a quarter turn the resulting signal will be 0.25 or 0.75 depending on the direction of turning.

C.3 IMU

The segway can try to measure its orientation by using an Inertial Measurement Unit (IMU), more specifically the BMX055. This chip can measure both acceleration and angular velocity. The chip is connected to the controller via a digital interface in such a way that:

Acceleration in the X, Y, and Z directions is mapped from -2 to 2 g to a signal of -1 to 1. This means that the acceleration felt due to gravity is mapped from $9.81\,\mathrm{m\,s^{-2}}$ to a signal of 0.5.

Rotational velocity around the X, Y, and Z axis is mapped from $-500\,^{\circ}\,\mathrm{s^{-1}}$ to $500\,^{\circ}\,\mathrm{s^{-1}}$ to -1 to 1. For example, a rotation around the Z axis of 50 degrees per second, will result in a signal of 0.1.

The sample frequency of the both accelerometer and gyro is equal to the frequency at which the controller is running but has one sample delay.

The IMU uses the coordinate system that is defined in figure C.1. Rotations are defined as a rotation around a certain axis. So for example a rotation around y would mean that the segway is steering.

C.4 Line sensor

The line sensor is based on a camera at the bottom of the segway. This camera is connected to a bit of image processing software to determine the line location. Based on the image the camera sees it will provide the following information on the position and angle of the line seen.

This information is presented as 4 points. These 4 points are the locations of the left and right side of the line, on the top and the bottom of the camera view. Here a signal of 0 means completely on the left of the screen, and 1 means completely to the right. For details on the names of the points, see Figure C.3

It also presents the color of the line seen, expressed as the angle in the YUV plane, mapped from 0-1. It is advised to calibrate these color values.

C.5 Gamepad

The segway is paired with a bluetooth gamepad. The 20-sim target is configured to read the buttons and analog channels of this gamepad.
The buttons are either 0 or 1, and the analog channels go from -1 to 1.
For the specific positions of the different buttons/channels, refer to Figure C.4.

C.6 Measurement instruments

There is a measurement disk provided with the segway. This disk has a set of hooks to attach weights and can be connected to the motor axle. A diagram of the disk is found in fig C.5. The disk has a moment of inertia of $\approx 4.694 \times 10^{-5}\,\mathrm{kg\,m^2}$. And a mass of 119,6 grams.

Bolts can be added to the disk to increase the moment of inertia. The bolts and nuts weigh 16.5 g to 4.6 g respectively.

Furthermore a digital scale, a set of weights (50 g each), a lever arm and a wrench are provided. The digital scale has a tare feature that can be used to zero the scale.

The lever arm has a length of 137.5 mm from the point of the rotation to the contact point.

Make sure to tighten the wing-nut when the measurement disk is attached.

Figure C.1: The segway used for these labs, with the coordinate system used by the IMU.

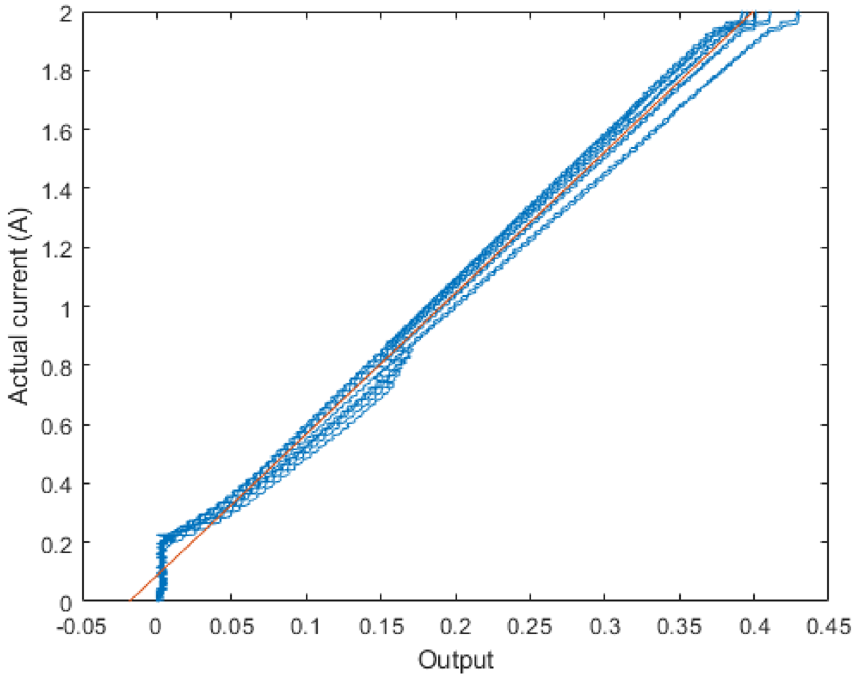


Figure C.2: The actual current versus the output of the current sensor.

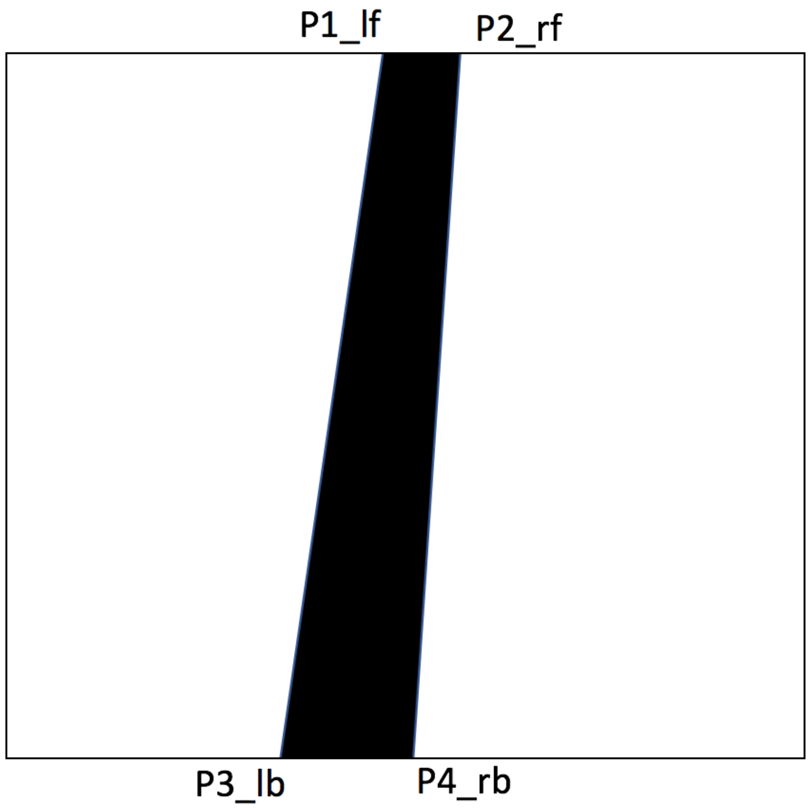


Figure C.3: The crossover points of the linesensor.

Figure C.4: The gamepad provided with the Segway

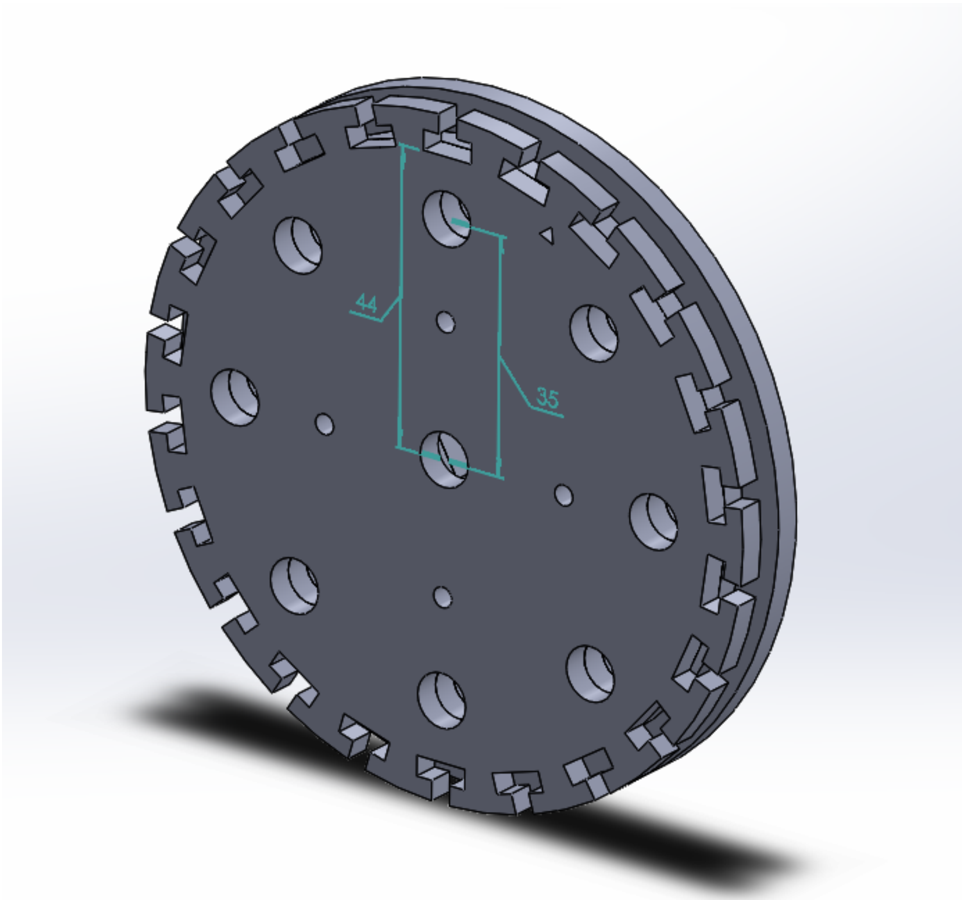


Figure C.5: Overview of the dimensions of the measurement disk (in mm)

Appendix D

Segway troubleshooting

There are two possible errors that can be generated by the 20-sim 4C connection. How to fix these errors is described in this section. Furthermore this section contains some information on the batteries of the segway and restarting it.

D.1 Possible errors

This section contains lists that you may encounter during your experiments as well as the solutions to these errors.

Error: Target could not be initialized
Solution: Make sure your segway is on and you have and that you are connected to eduram. If this is the case this error is likely caused by the fact that 4C cannot find the segway device and does not retry the connection. To fix this, make sure your hostname is correct in the target section of 4C. Save your project and restart 4C.

Error: ipcv library could not be loaded
Solution: This error is usually caused by connecting a new instance of 4C without properly closing the previous connections. Or if the Segway fell way too hard. To fix this, restart the segway, see section D.2.

Error: tokenparser.xml not found
Solution: You unzipped your target in the wrong directory. The folder C:/program files (x86)/20 sim 4C/target/ should contain a folder a called raspberry-pi-Linux.

Error: 20-sim-4c does not have a target called Raspberry Pi Linux.
Solution: Make sure you unzipped the target that can be downloaded from canvas into the folder C:/program files (x86)/20 sim 4C/target/

Error: 20 sim-asks for a license
Solution: Click on activate and use the license code that is supplied via canvas.

Error: I plugged in the charger of my segway but the led on the power shield stays green (means not charging).
Solution: Turn on the segway.

Error: error 14: ivcCommand.getValuesBinary –¿ Application is not running..

Solution: Turn on the segway.

D.2 Restarting the segway

To restart the Segway follow the following steps:

1. Unplug the adapter
2. Hold the button on the segway until the red light on the battery back blinks and the lights turn off.
3. Press the button again.

Note that when the segway is starting up, it cannot be shutdown. To shut it down you have to wait for about 60 s.

D.3 Batteries

The segway has a set of LEDs on the motor shield on top of Raspberry Pi. The colors of the lights have certain meanings in relation to the batteries:

- **BLUE** the batteries are fully charged
- **RED** the batteries are charging.
- **GREEN** the segway is on but isn't charging.

When none of these light are on, the segway is off. The segway should be able to run for about **30 minutes** on a full charge.
Note 1: If GREEN LED is flashing while charging, this is OK (LED functionality fluctuates between battery detection and charging cycle).
Note 2: For the new shields, consider GREEN and BLUE LED functionality swapped.

There also is a set of LEDs on the battery pack inside the segway. The LEDs on the battery pack will blink **RED** on both start-up and shutdown. Besides that the colors of the lights have the following meaning

- **GREEN** The battery pack is on and functioning normally
- **RED** The batteries are too full.
- **YELLOW** The batteries are too empty.

Note that the green LED is very dim.

Appendix E

Report criteria

You have to hand in a report for both of the projects. Make sure to mention the contribution of each student to the work done throughout the project to make sure everyone participated. For the report, we provided you with an overleaf template on: `https://www.overleaf.com/read/bvkvzbwqtcjc`

E.1 General Report Content

To make it easy for people to understand your report it is good practice to follow a standard report structure. In this case you could follow the following structure:

Introduction Describe what the reader can expect while reading your report and what you are trying to achieve.

Analysis Explain what you expect and what information you require to do the measurements.

Methodology contains what you are going to measure and how you are going to measure it.

Results show the outcome of your (processed) outcome of the measurements. It is not required to prove your measurement. However, it is important that the reader can reproduce your results.

Conclusion Finish up your report. Reflect on your introduction and discuss if you succeeded or not.

Discussion Discuss what went wrong, why it went wrong, and what can be improved for future work.

The final form of this basis can be changed. However, make sure that the content of this list is still in your report.

E.2 Checklist Report

Any scientific report has to meet a list of requirements, which make sure that your report can only be interpreted in one way.

- Did you use the correct units for each value?
- Does every figure have a caption?

- Does every figure have units on the x and y axis?
- Can the axis of each figure be read without zooming in? If not rescale your 20-sim simulation window such that is not full screen anymore.
- Can your diagrams be read without zooming in?
- Do you refer at least once to each figure in your text?
- Is it clear what each component in your model represents?
- Did you use citations where necessary?
- Are your citations formatted correctly?

E.3 Readable figures

Having readable figures that explain your data is essential for a good report. In this section, we will explain how to obtain the perfect figures.

E.3.1 Saving and exporting your data

The first step in obtaining good figures is to save your measurement data such that you can look at this data later. It is smart to use a proper name convention for these files, to find the data easier. Once you start writing your report and you need a figure of the angular velocity, it won't be easy if you name your measurements: '*measurement1.csv*', '*measurement2.csv*', '*measurement3.csv*'.

If you save your data as a comma-separated file (.csv), you are still able to zoom in and arrange your data at a later stage. If you save it as a figure, you are **not** able to scale or zoom without significantly degrading the quality of the plot. The .csv files can also be loaded into MATLAB in order to create nice figures.

> **Tip:** You can change the name of the sub-model file every time you do a simulation, and use this same name for the simulation data. This will save you a lot of time later when writing your report and will allow you to save your progress in the meantime. If you ever have a problem, you can easily revert to an old version.

E.3.2 Visibility

The most important rule of thumb is to keep in mind what you want to show the reader of your report with your figure. The reader should not need more than a few seconds to look at the figure to understand the message. This means that the reader should not need to zoom-in or think too much about the figure. It should act as a proof or additional graphical information to accompany the text. If the reader needs a lot of time to understand the figure, it might have been better just to write it down instead.

When plotting data in a figure also make sure the axes are clearly visible and have correct units. The same holds for the labels of the data if more than one signal is plotted. The caption below the figure should be a concise description of what the reader should learn from the figure.

If more than one signal is plotted, make sure that the data is related to each other and easily distinguishable. If this is not the case it might be better to create two sub-figures. No one will be happy with a cluttered figure with too much data.

Small checklist:

- Can you explain the figure within a few seconds? If this is not the case, try to simplify the image or create multiple figures.
- Are the labels and the axis readable without having to zoom-in?
- Do all the axes have the proper units?
- Does the figure have a clear description in the caption?
- If you have multiple signals in your plot, are all the signals easily distinguishable? If not, try to create multiple plots.

E.3.3 A few examples

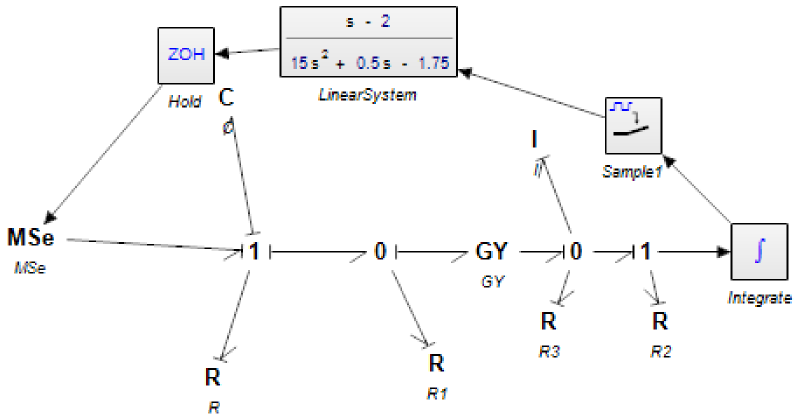
Figures

In figure E.1, two versions of a figure containing a bondgraph and a controller can be seen. The bondgraph as in figure E.1a can be described as a chaotic figure that is difficult to read. The bondgraph consists of several parts that have different functions. In figure E.1b it can be seen that these sections are highlighted which makes it a lot easier to reference. The reader immediately recognizes the part of the bondgraph that belongs to the motor model and the part that belongs to the body model.

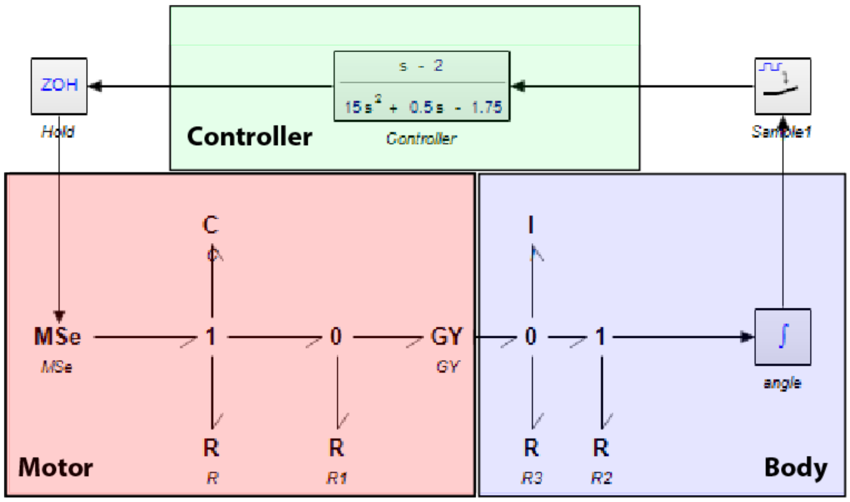
Keep in mind that if the model is larger, the figure can get too busy, even if you use different sections as shown in figure E.1b. It is then recommended to implode different sub-models and create different figures to show the contents.

Plots

If you want to plot several measurements into one graph, check whether your data is actually visible and whether the plot actually conveys the information you want. In figure E.2a, an example is shown of a plot containing four different measurements with four different parameters. The blue line clearly dominates in the figure and causes the other lines to be barely visible. One option is here to separate all the lines into four different plots as shown in figure E.2b. This is not always possible. In that case evaluate whether you actually need the first measurement in the first place, or that it might be sufficient to only describe what happens. This way the other measurements can be plotted into the same graph, if they still remain distinguishable.

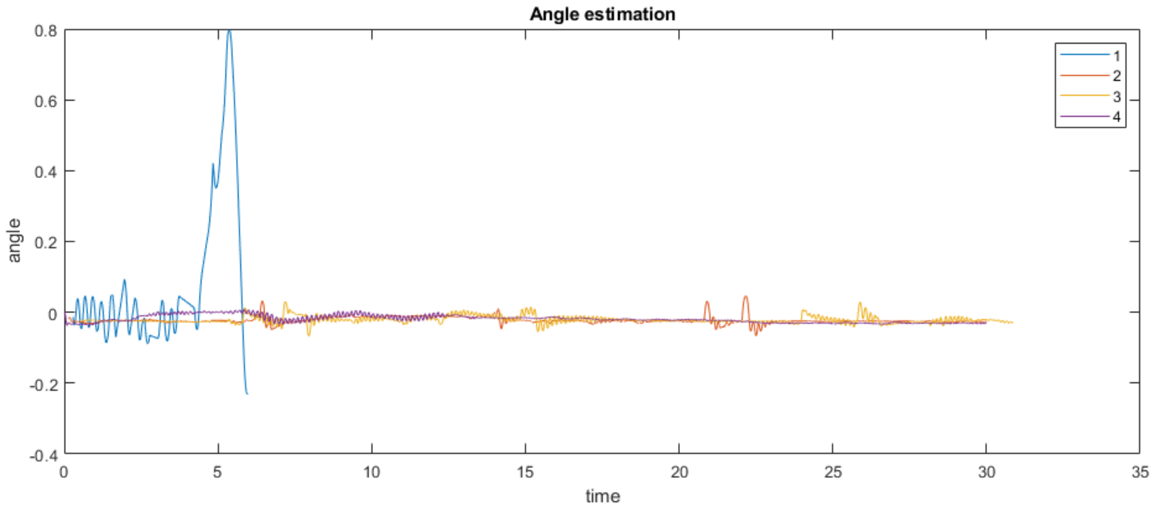


(a) A chaotic figure of a bondgraph and controller

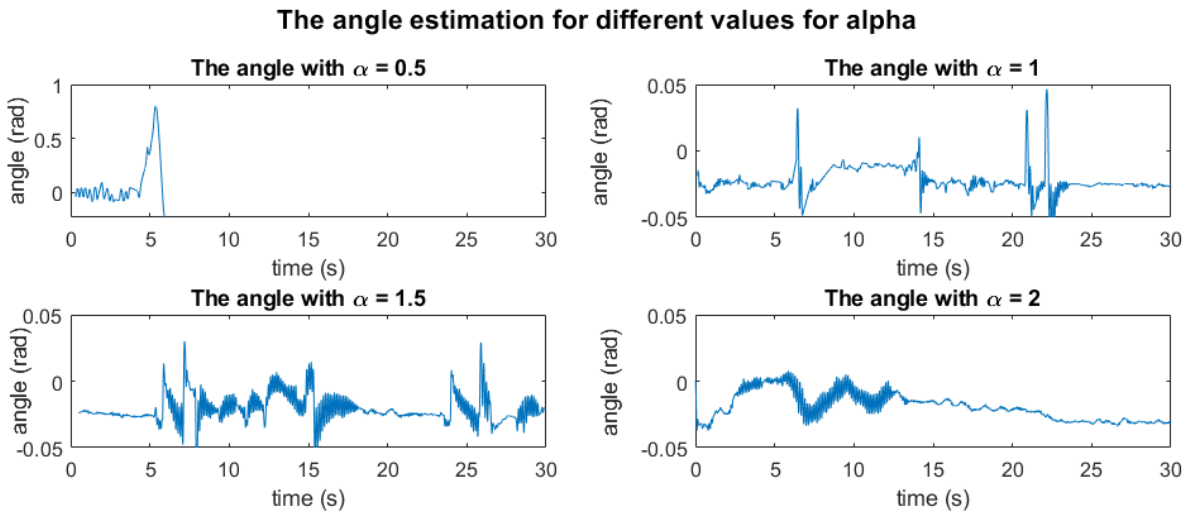


(b) A clear figure of a bondgraph and controller

Figure E.1: A good example and a bad example of a figure containing a bondgraph and controller.



(a) A bad figure containing data



(b) A clear figure containing data

Figure E.2: A good example and a bad example of a figure containing several measurements

Appendix F

Motor lab grading form

Group:

Writing and structure (15% of final grade)		
Level of writing		
grammar		
sentence structure		
references used		
formal writing		
Quality of the used figures		
Readability		
Units on axis		
Figure captions		
Structure of the report		
Follows a logical struture (seperation of method and execution)		
Reader is guided through the report		
Equations and units		
Equations are clear		
Variables are clearly defined and consistently used		
The correct units have been used		
Total		
Introduction and conclusion (10% of final grade)		
Introduction that introduces the problem and sets well defined goals		
Conclusion reflects back to goals in the introduction		
Total		
Modelling (20%)		
Qualitative description of the physical system		
Explanation of simplifying assumptions / effects neglected in the model		
Model is correct and briefly explained		
Model is presented as a bond-graph		
Essential diifferential equations describing the model are briefly derived		
Total		
Characterisation (25%)		
Overview of conducted experiments		
Explanation of individual experiments		
What is the experiment?		
Which parameters are determined?		
Necessary equations to identify parameters		
Experiment can be reproduced from explanation		
Characterisation of the model parameters		
Correct values and units, possibly error bars		
Computations are transparent and reproducable		
Parameters are validated using further experimental data		
The simulated model behaves similarly to the real system		
Differences are explained and corrected		
Total		
Control (20%)		
Transfer function of the plant is identified		
The controller is designed & tuned in a systematic way (TF of controller)		
Control objective is determined (quantitative criteria are up to you!)		
Clear explanation of control architecture and tuning decisions		
Controller performance is validated on the real plant		
Error is plotted over time		
Controller is succesful		
Limitations of controller are explained		
Total		
Discussion (10%)		
Discussion on what went well, what not and what could be improved		
	Final grade:	

Appendix G

Segway lab grading form

Group:

Writing and structure (15% of final grade)	
Level of writing	
grammar	
sentence structure	
references used	
formal writing	
Quality of the used figures	
Readability	
Units on axis	
Figure captions	
Structure of the report	
Follows a logical struture (seperation of method and execution)	
Reader is guided through the report	
Equations and units	
Equations are clear	
Variables are clearly defined and consistently used	
The correct units have been used	
Total	
Introduction and conclusion (10% of final grade)	
Introduction that introduces the problem and sets well defined goals	
Conclusion reflects back to goals in the introduction	
Total	
Modelling (20%)	
Qualitative description of the physical system	
Competent graphical overview	
Model is correct and briefly explained	
Very briefly: motor and wheels	
Reference frames and transformations between them are clear	
Interconnections between model parts are clear	
Explanation of simplifying assumptions / effects neglected in the model	
Model is presented as a bond-graph	
Model is visualized in a 3D animation to show that it behaves as expected	
Total	
State estimate (20%)	
Determine goal of state estimator: e.g., states to be estimated	
Model sensors in simulation	
Include realistic sensor noise in simulation	
Explain the method used for state estimation	
E.g., complementary filters, observers, etc.	
Validation of the state estimator	
Validation within simulation by comparing estimated state and true state	
Implementation on real system and reflect on differences to simulation	
Total	
Characterisation (10%)	
Overview of conducted experiments, summary of reused old results	
List of required parameters, and how they relate to the experiments	
Explanation of individual experiments	
Setup of experiment	
Which parameters are determined?	
Necessary equations to identify parameters	
Experiment can be reproduced from explanation	
Parameters are validated using further experimental data	
The simulated model behaves similarly to the real system	
Differences are explained and corrected	
Total	
Control (20%)	
Transfer function or state-space of the plant is identified	
The controller is designed & tuned in a systematic way	
Control objective is explained (motivate criteria you choose!)	
Clear explanation of control architecture and tuning decisions	
Explanation of how control using the gamepad is implemented	
Controller performance is validated:	
Error is plotted over time	
Controller is succesful in simulation	
Controller is succesful in reality	
Limitations of controller are explained	
Total	
Discussion (5%)	
Discussion on what went well, what not and what could be improved	

Final grade: 0