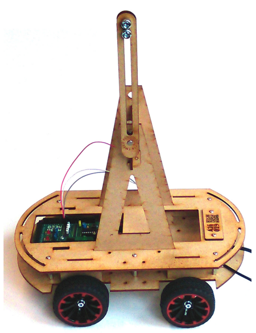
# Control Theory – Assignments

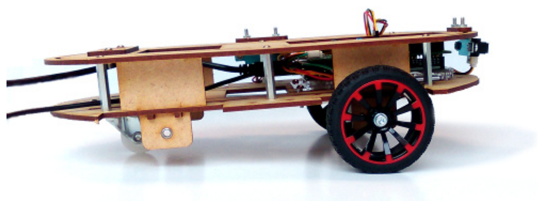
## 1 Introduction

The evaluation of the course Control Theory is based on practical assignments in which you will apply several techniques taught during the course in order to control the cart presented in Figure 1. All carts have two separately driven wheels in the front. At the back, one type of cart is equipped with a free axle with two wheels, for moving on a straight line, while the other carts have a swivel wheel to allow rotation. The carts are controlled by an Arduino Mega, running MicroOS as software framework. For more information on this system, you are referred to the tutorial session.

You work in groups of 2 students. Be sure to **register your team by Wednesday November 03, 2021** in the provided Tolinto module. After the registration, you will get a cart assigned to your group. If your cart has a swivel wheel, you must perform assignments 1, 2, 3 and 5. Students with a 4-wheeled cart perform assignments 1, 2, 3 and 4. If for some reason you are alone in a group (e.g. your team partner stopped following the course), you still perform the same assignments as a group of 2 students.



(a) 4-wheeled cart with pendulum



(b) Cart with swivel wheel

Figure 1: Setup used during the assignments.

### 1.1 Instructions for the evaluation

You write your answers to the assignments down in a compact report. In addition, we ask for the source codes of your implementations as well as a movie of one of your experiments. In a second step, you will be invited to for an oral (off-campus) authenticity check of your work during the examination period in January. You will be able to register for a time slot that fits you best through a Tolinto module. At this point, the examination schedule is unfortunately not known yet. More details are therefore communicated later.

The **deadline** for handing in your report plus source files and movie is **Friday December 24, 2021** (11.59 AM). At that time a submission module will be made available on Toledo, in the menu "Assignments", for uploading the required files.

Please read the following instructions carefully and comply with them!

#### Instructions for the report

- Make a separate PDF file for every assignment. Make sure that your team number and your names are clearly mentioned in every file. Save as `teamXX_2021_report_assignment_YY.pdf`, with XX your team number and YY the assignment number.
- Answer all questions of the assignments. All questions are assessed.
- Provide answers in a clear way. Be complete as well as concise. In order to help you with this, we have provided a structured blueprint of what your answer is supposed to look like. Keep the following guidelines in mind:

- if you use symbols in equations or formulas, make sure you *declare* what every symbol means;
- if we ask for a plot, we **also always** expect a brief explanation of *what* we see in this plot and *why* we see this; remove irrelevant signals to avoid confusion;
- if we ask for a numerical value or an explicit symbolic expression, we **also always** expect a brief *interpretation* on how this value relates to the behavior of the system;
- if you make a design choice (filter parameters, controller parameters, uncertainties, ...), we **also always** expect a brief *motivation*.

In all cases **a few lines** of explanation (max. 10) should suffice. We know the assignments and their context, so you can answer straight to the point. Since there is no possibility for an extensive oral discussion on your results, we would like to emphasize once more that your report should be **complete and self-explanatory**.

- Provide clear figures: all lines and colors must be visible, and axes must have readable labels with physical units. Also numerical values have units. The quality of the report (completeness, notation, units, ...) will be accounted for in the overall evaluation!
- The report may be written in English or in Dutch, as you prefer.
- It is important to notice that, even though we encourage you to discuss your findings with other students (see further), every team has to **perform its own experiments and to write its own report**. Submitted reports are checked for similarity as well as plagiarism through Turnitin. Its database also includes the reports of your predecessors who took the Control Theory course earlier.

**Instructions for the movie**

- Film your most advanced/successful experiments with a camera, tablet, or smartphone.
- Make sure the platform ID as well as one of the team members' student card is clearly visible in the movie.
- Make sure your movie is compressed (web-optimized). Have a look at HandBrake, for example.
- Save in MP4 format: `teamXX_2021_movie.mp4`, with XX your team number.

**Instructions for the source code**

- Compress your MATLAB files as well as the entire Arduino sketchbook folder into one zip file: `teamXX_2021_source.zip`, with XX your team number.
- Make sure the code is stand-alone executable (include all "helper" files), and to select logical names for the main routines.

**Instructions for the authenticity check**

- There is no extensive oral defense. Instead we organize an authenticity check that only comprises 5 minutes of limited questioning on randomly selected topics of your report by the examiners. During these 5 minutes, there is no time for an in-depth discussion.
- This check will be in English or in Dutch, according to your preference. The specific modalities will be communicated later.
- Assignments that are not treated in the report will also not be addressed during this oral discussion.

## 1.2 Feedback sessions and support

You can pick the platform up at the electronic equipment service point ('Uitleendienst elektronica', C300 01.153) of the mechanical engineering department during office hours (8 AM - 12 PM, 1 PM - 6 PM, unless explicitly stated otherwise). Similarly, you have to return it there, at the latest on the deadline for handing in the reports.

In case there are questions, we strongly encourage you to consult with your fellow students. Discuss your findings and questions amongst yourselves and try to come up with an explanation or an answer. A dedicated forum is also made available on Toledo for this purpose. In case you cannot get to a consensus, you can ask the teaching assistants to help you out. They will be available for questions **during plenary feedback sessions only**.

- **assignment 1** (FB1): Friday November 12, 4.00 - 6.00 PM, Aud. D
- **assignment 2** (FB2): Friday December 03, 4.00 - 6.00 PM, Aud. B
- **assignment 3** (FB3): Friday December 10, 3.00 - 5.00 PM, Aud. E
- **assignment 4, 5** (FB45): Friday December 17, 4.00 - 6.00 PM, Aud. A
- **all assignments** (FB*): Wednesday December 22, 10.30 - 12.30 AM, Aud. D

In the first four sessions, only questions concerning the specific assignments that are mentioned will be treated. Therefore, and as stated earlier, make sure you have finished the previous assignments before! During the 5th session you can ask questions about any of the assignments. Keep in mind, furthermore, that the assignments serve as your examination, so do not expect the assistants to solve them on your behalf.

In case your cart is broken, please don't hesitate to pass by the teaching assistants (rooms 01.17 and 01.19) as soon as possible such that we can fix whatever is broken. Do not return broken platforms without further notice!

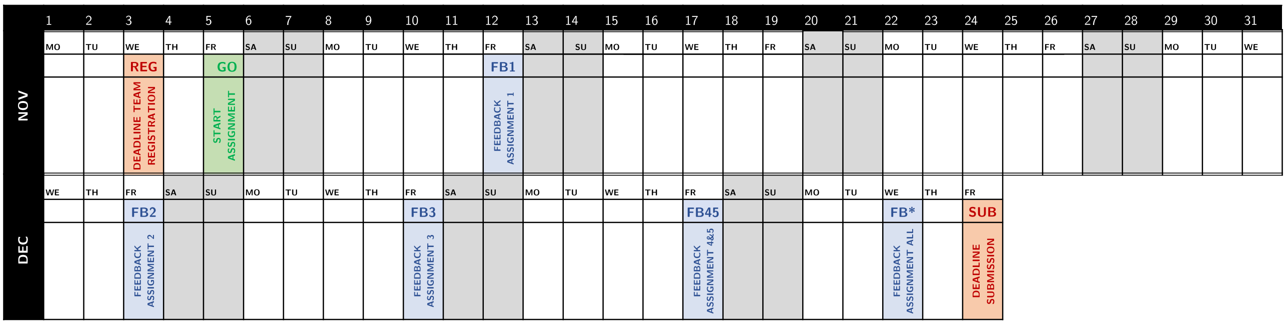



Figure 2: Timeline overview of deadlines and feedback session of the exam assignments.

# 2 Assignments

If your cart has a swivel wheel, you must perform assignments 1, 2, 3 and 5. Students with a 4-wheeled cart perform assignments 1, 2, 3 and 4.

## 2.1 Assignment 1: identification of the motors

1. Select an appropriate model structure for the DC motors.

   (a) Select a discrete-time model structure that you will used for identification of the relation between motor voltage to rotational wheel velocity.

   STRUCTURE YOUR ANSWER AS FOLLOWS:

   - Present the discrete-time transfer function while clearly indicating the order of the numerator(s), the order of the denominator and the number of delays. This selection is based on the physical laws and continuous-time transfer function describing the behavior of your system, simplifications you assume, the sampling process (zero-order-hold) and potential delay(s) introduced by the software framework MicroOS. Do not derive the physical laws but just present and discuss them (which effects play a role).
   - Briefly (max. 10 lines) motivate your choice(s). If you plan to try out more than one model structure, explain your motivation for this as well, but limit the number of different model structures to three.

   (b) What are the input(s) and output(s)? And what are their physical units?

   STRUCTURE YOUR ANSWER AS FOLLOWS:

   - What is the physical meaning of your model's input(s) and what is/are the unit(s)?
   - What is the physical meaning of your model's output(s) and what is/are the unit(s)?

2. Identify each 'DC motor plus wheel' unit by exciting the motor while holding the cart in the air.

   (a) Which excitation signal do you apply? Why?

   STRUCTURE YOUR ANSWER AS FOLLOWS:

   - How do you design your excitation signal?
   - Plot your excitation signal. Do you apply it just once or do you repeat it? Why?

   (b) How do you obtain your system parameters? Write down the recursion expression (difference equation) and criterion that you used to estimate the parameters.

   STRUCTURE YOUR ANSWER AS FOLLOWS:

   - Write down the recursion expression (difference equation), specifically for your model, that you solve to find the parameters, and write down the error criterion that this expression minimizes.
   - Calculate the location(s) of the pole(s) and/or zero(s) of your model and express them as a frequency (in rad/s or Hz).

   (c) Can you use filtering to improve the identification? Why? Which filter do you choose, and how do you select its parameters? Which signals do you filter?

   STRUCTURE YOUR ANSWER AS FOLLOWS:

   - Why do you prefer to filter your signals before fitting a model? Which signals do you filter before fitting? Explain.
   - How do you design the filter that you use?
   - Write down the characteristics of the filter that you use (type, order, cut-off frequency, ...).
   - Redo the identification after filtering your data and calculate again the location(s) of the pole(s) and/or zero(s) of your model, expressing them as a frequency (in rad/s or Hz). Have their location(s) changed w.r.t. 2(b)?
   - Make one pole-zero map with both the pole(s) and/or zero(s) of your first identified model (without filtering) and your final identified model (with filtering).

(d) Validate your identified model experimentally. Plot the difference in response of the simulated model and the real system. As any physical system, the 'DC motor plus wheel' is in practice nonlinear. Prove this experimentally by checking for violations of the superposition principle that holds for linear systems.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Plot and compare the measured response of your system on a step input with the simulated response of all identified models. Also make a (separate) plot of the differences. Discuss this comparison. Discuss also the characteristics of the model and the differences between models.
> - Which model would you select as being the best considering balance between accuracy and complexity. Motivate your choice.
> - What is the superposition principle? Express the principle as a formula, specifically in terms of the input and output of your system. Then, check experimentally for violations of the principle for your system.
> - What are the causes of the violations of the nonlinearity of your specific system?

3. Identify the cart.

(a) Place the cart on the ground and validate the identified models of the two DC motors again. Are there differences? Explain them.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Put the cart on the ground and apply a step input to both motors. Plot their measured response together with the simulated response of the model you selected in 2(d). Also make a (separate) plot of the difference.
> - Is the response similar as in assignment 2(b)? Explain the differences.

(b) Identify the motors again with the cart placed on the ground. Are the model parameters different? Has the model accuracy improved?

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Re-identify the behavior of the motors when the cart is placed on the ground. You only have to re-identify the parameters of the model structure that you selected as the best one in 2(d). Write down the numerical values of the model parameters and compare them with your previously identified parameters. Calculate the location(s) of the pole(s) and zero(s) of your model and express them as a frequency (in rad/s or Hz).
> - How did you expect the model parameters to change qualitatively when putting the cart on the ground? Does this match with the numerical values?

## 2.2 Assignment 2: velocity control of the cart

1. Design for each DC motor a velocity controller that yields zero steady-state error on a constant velocity reference. Design the controller using frequency response methods. Use the identified models of assignment 1.

   (a) What type of controller do you choose? Why?

   > Structure your answer as follows:
   >
   > - Write down the requirements/specifications that you would like to obtain from your controlled system. Which controller types can realize those?
   > - Select one of the possible controller types and motivate your choice.

   (b) Explain the design process and all choices of design parameters (phase margin, cut-off frequency, integration time, ...).

   > Structure your answer as follows:
   >
   > - Write down the formulas you use to compute the design parameters of the selected controller type in order to meet the requirements you listed in (a). Also write down the numerical values you enter in, and obtain with these formulas.
   > - Briefly explain the design trade-offs involved in the controller design parameters. That is: for every parameter explain the advantage/disadvantage of increasing/decreasing its value.
   > - Plot the open loop (= serial connection of your controller and your model) on a Bodediagram and clearly indicate *on this plot* where you see the characteristics that affect the choice of your design parameters.
   > - Make a Bodeplot of the closed-loop frequency response.

   (c) Is there a theoretical limitation on the closed-loop bandwidth that can be achieved? Is there a practical limitation?

   > Structure your answer as follows:
   >
   > - Is it theoretically possible to design a controller that yields a higher bandwidth? If not, explain the corresponding theoretical bandwidth limitations.
   > - Do you expect / experience practical problems when increasing the closed-loop bandwidth of the controller? If so, explain the corresponding practical bandwidth limitations.
   > - Does the way you implement your controller in practice (software + microcontroller) impose theoretical and/or practical constraints on the achievable bandwidth?

2. Validate the designed controller experimentally.

   (a) Implement the controller for both wheels on the Arduino and apply a step input for the reference velocity. Plot the error between the reference and measured velocity, and plot the corresponding actuator signals.

   > Structure your answer as follows:
   >
   > - Plot, on one figure, (i) the step reference, (ii) its measured closed-loop response and (iii) its simulated closed-loop response.
   > - Plot the measured tracking error of the step reference together with the simulated tracking error.
   > - Plot the measured control signal (= voltage) of the step reference together with the simulated control signal.

   (b) Find a way of applying a constant force disturbance to the cart. Is your controller still following the velocity setpoint? Why?

   > Structure your answer as follows:
   >
   > - How can you apply a constant force disturbance to the cart? Draw the block diagram of your control configuration and indicate where the disturbance is entering the loop.
   > - Apply the same velocity setpoint as in (a) and apply the constant force disturbance. Make the same plots as in (a).
   > - Is the controller still tracking the reference despite the disturbance? Could you think of other disturbances that do (not) affect the steady-state tracking error?

## 2.3 Assignment 3: state feedback and state estimation

In this assignment you will control the position of the cart while driving on a straight line (i.e. both velocity controlled motors should get the same velocity setpoint $v$). To this end, a position control loop is added on top of the velocity controllers designed in assignment 2. For the position control you will design a state feedback controller. In order to retrieve an estimate of the state, you will implement a Kalman filter. This filter uses distance measurements from the frontal infrared sensor to correct its estimate. This infrared sensor measures the distance to a wall in front of the cart. The range in which this sensor is accurate is limited (approximately 5-30 cm), so make sure the cart does not exceed those limits during your experiments to avoid strange results. The origin is chosen on the wall. As the cart is positioned at the negative side of the origin, the infrared sensor measures $-x$, which is a positive value. To model the cart, you can assume the velocity control loop is ideal, i.e. you assume that the desired velocity is equal to the real velocity of the cart.

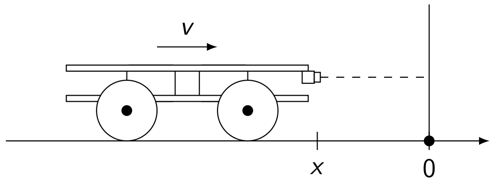



Figure 3: Cart measuring its distance to a wall.

For this assignment you can use the Arduino code provided on Toledo: `CT-KalmanFilter`. This code implements a working Kalman filter and applies a reference velocity signal to the motors to actuate the cart. You should extend the code by implementing the speed controller you designed in assignment 2. The comments in `robot.cpp` provide the body of an implementation and guide you where to add calls to your speed control routines. For your convenience, the position controller is already implemented, you should however adapt the control gain $K$. Likewise, in the Kalman filter, the process noise covariance $Q$ and the measurement noise covariance $R$ need to be tuned, which can be found in `kalman_filter.cpp`. Throughout the assignment you can assume a constant $Q$ and $R$.

1. Design a state feedback controller.

   (a) Write down the state equation in discrete form. The velocity of the cart can be seen as input. Use a forward Euler method as discretization scheme.

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   >
   > - Write down the continuous-time state equation and show how you discretize it using the forward Euler discretization scheme. What are the $A$ and $B$ matrices of your state-space model?

   (b) Derive an expression for the pole of the closed-loop system as a function of the sample time $T_s$ and the state feedback gain $K$. What happens with the pole when you increase or decrease $K$? What happens with the closed-loop response? Can we choose $K$ such that the system becomes unstable?

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   >
   > - Express the pole of the discretized closed-loop system symbolically. How does it change as a function of $K$?
   > - Make a pole-zero map and draw how the pole location changes with varying $K$.
   > - How does the response speed depend on $K$? Can we choose $K$ such that it becomes unstable?

2. Validate the principles of the Kalman filter on your system.

   (a) Write down the measurement equation.

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   >
   > - Write down the measurement equation. Use the $x$-axis convention of Fig. 3. What are the $C$ and $D$ matrices of your state-space model?

(b) Derive an expression for the (time-varying) Kalman gain $L_{k+1}$ as a function of the state estimate covariance $\hat{P}_{k|k}$, the process noise covariance $Q$ and measurement noise covariance $R$. How does $L_{k+1}$ evolve when $Q \to \infty$? How does $L_{k+1}$ evolve when $R \to \infty$? Is this what you expect?

> STRUCTURE YOUR ANSWER AS FOLLOWS:
> - Derive an explicit form of the requested symbolical expression, specifically for this model, that is, you can assume $Q$, $R$, $\hat{P}$ and $L$ to be scalars, and directly fill in the state-space matrices.
> - Calculate the limit of this expression for $Q \to \infty$.
> - Calculate the limit of this expression for $R \to \infty$.
> - What does it mean when the Kalman gain takes the limit values that you just calculated?

(c) Derive an expression for the next state estimate covariance $\hat{P}_{k+1|k+1}$ as a function of the previous state estimate covariance $\hat{P}_{k|k}$, the process noise covariance $Q$ and measurement noise covariance $R$. How does $\hat{P}_{k+1|k+1}$ evolve when $Q \to \infty$? How does $\hat{P}_{k+1|k+1}$ evolve when $R \to \infty$? Is this what you expect?

> STRUCTURE YOUR ANSWER AS FOLLOWS:
> - Derive an explicit form of the requested symbolical expression, specifically for this model, that is, you can assume $Q$, $R$, $\hat{P}$ and $L$ to be scalars, and directly fill in the state-space matrices.
> - Calculate the limit of this expression for $Q \to \infty$.
> - Calculate the limit of this expression for $R \to \infty$.
> - How does the uncertainty on the state estimate evolve in these two limit cases? Is this in accordance with the normal operation of a Kalman filter? Explain.

(d) Derive an expression for the steady-state covariance $\hat{P}_{\infty}$ and related Kalman gain $L_{\infty}$ as a function of $Q$ and $R$. This gain should be equal to gain of a Linear Quadratic Estimator (LQE). Check this for some numerical value for $Q$ and $R$ using the `dlqr` command in MATLAB.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
> - Derive an explicit form of the 2 requested symbolical expressions, specifically for this model, that is, you can assume $Q$, $R$, $\hat{P}$ and $L$ to be scalars, and directly fill in the state-space matrices.
> - Choose several numerical values for $Q$ and $R$ and verify that your expression for $L_{\infty}$ yields the same result as `dlqr(A',(A'*)C',Q,R)'` in MATLAB.

(e) Derive an expression for the closed-loop pole of the Linear Quadratic Estimator (LQE) as a function of $\frac{Q}{R}$. What happens with the pole when you increase or decrease $\frac{Q}{R}$? Is this what you expect? Can we choose $\frac{Q}{R}$ such that the system becomes unstable?

> STRUCTURE YOUR ANSWER AS FOLLOWS:
> - Derive an explicit form of the requested symbolical expression, specifically for this model, that is, you can assume $Q$, $R$, $\hat{P}$ and $L$ to be scalars, and directly fill in the state-space matrices.
> - How does the pole location change with $Q/R$?
> - Is it possible to make the system unstable by choosing an appropriate $Q/R$ value?

3. Implement a state estimator and a state feedback controller.

(a) Choose appropriate values for the control gain $K$, process noise covariance $Q$ (see the `PredictionUpdate`-routine in `kalman_filter.cpp`) and measurement noise covariance $R$ (`CorrectionUpdate`-routine in `kalman_filter.cpp`). Adapt these values in the C++ code provided on Toledo. Also, make sure to set the initial state estimate and its covariance appropriately in the `resetKalmanFilter`-routine in `robot.cpp`.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
> - What are the potential sources of process noise and measurement noise in this system?
> - How do you choose appropriate values for $K$, $Q$, $R$ and the initial state covariance matrix $P_{0|0}$?
> - Give the numeric values of $K$, $Q$, $R$ and $P_{0|0}$ that you have chosen (don't forget the units).

(b) Apply a step signal as position reference for different values of $K$. Illustrate the different step responses in one plot, and do the same for the corresponding control signals. Explain the plots using the analysis made in 1(b).

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Plot, on one figure, (i) the (position) step reference and (ii) its measured response for different values of $K$. Explain the relation with the results of 1(b).
> - Plot, on one figure, the low-level control signals (= voltage) that correspond to the responses of the previous figure. How do they depend on the value of $K$? Explain.
> - Are there theoretical and/or practical limitations on the choice of a feasible $K$? Which one(s)?

(c) Apply a step-signal as position reference for different values of $Q$ and $R$ (e.g. values between $10^{-4}$ and 1). Plot the evolution of $\hat{P}_{k|k}$ in one plot and the evolution of $L_k$ in another plot. Explain the plots using the analysis of 2(e). Do the values converge towards respectively $\hat{P}_\infty$ and $L_\infty$, computed in 2(d)?

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Plot the evolution of $\hat{P}_{k|k}$ and the corresponding $L_k$ (on a separate figure) in time for different values of $Q$ and $R$ for the same step reference for the position. How do these plots relate to the answers of 2(e) and 2(d)?

(d) Using the data from the previous question 3(c), calculate the evolution of the NIS and the SNIS, and check whether their values are within a certain confidence interval (e.g. 95%) for the distribution you would expect. Is your Kalman filter consistent? If not, can you think of possible causes for this?

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Write down the formulas that are required to calculate the NIS and the SNIS. Fill in all known variables, i.e. the expressions should only depend on variables that you obtain from your experiment(s).
> - Plot the evolution of the NIS ans SNIS in time for the different experiments of 3(c). Make sure it is clear which plot corresponds to which $Q$ and $R$. Use the provided MATLAB function `analyzeconsistency` (see exercise session 5). To import your data from QRoboticsCenter for this specific assignment, you can call the method `KalmanExperiment.createfromQRC3()`.
> - How can you check the consistency of your Kalman filter? Which probability distribution for the NIS and SNIS values do you expect?
> - Is your Kalman filter consistent during the different experiments?
> - In some cases, the Kalman filter may be inconsistent. What are the most important causes for this (in this specific experiment) according to you?

(e) Use a wrong initial estimation for the position. Apply a step as position reference. Plot the evolution of the position estimate together with the measured position by the infrared sensor. Do this for different values of $\frac{Q}{R}$. Explain the different responses.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Apply a step reference in the desired position. Let the estimator start at the same time, but with a wrong initial estimate. Plot the measured distance together with the estimated distance for different values of $Q/R$.
> - For which $Q/R$ ratios do the estimates converge faster to the measurements?

(f) Design a state estimator using pole placement, such that the closed-loop pole of the estimator is 10x *slower* than the closed-loop pole of the state feedback controller you have chosen. Then, start from an initial estimation error and apply a step as desired position signal. Plot the estimated position and measured position in one plot. Explain the observed behavior.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Start from a wrong position estimate and, at the same time, apply a step reference as the desired position. Plot the measured distance together with the estimated distance.
> - Is the control performance satisfactory? If you were to design a state estimator using pole placement yourself, where would you put its closed-loop pole?

## 2.4 Assignment 4: control of a pendulum

This assignment uses a cart with a pendulum. The pendulum is attached to a rotary encoder which allows you to measure the pendulum angle. In this assignment you will design a state feedback controller to control the position of the pendulum mass. A state estimator will be required for providing the controller with state information. In this assignment you can again assume that the velocity control loop is ideal. You should only concentrate on the pendulum in its stable position.

1. Model and identify the system.

   (a) Derive a theoretical continuous model of the velocity-steered cart with a pendulum.

   STRUCTURE YOUR ANSWER AS FOLLOWS:
   - Which physical laws describe the behavior of your system?
   - Derive the model equations. What are its parameters?
   - What is the input of the model and what is its unit?
   - What are the outputs of the model and what are their units?

   (b) Derive a state-space equation for the nonlinear model (i.e. $\dot{\xi} = f(\xi, v)$). Use as states $\xi = [x, \theta, l\dot{\theta} + \dot{x}\cos\theta]^T$, with $x$ the position of the cart and $\theta$ the pendulum angle.

   STRUCTURE YOUR ANSWER AS FOLLOWS:
   - What is the meaning of the different states $x$, $\theta$ and $l\dot{\theta} + \dot{x}\cos\theta$ ?
   - Derive a nonlinear state-space equation, i.e. look for the possibly nonlinear function $f$ such that the evolution of the states is prescribed by $\dot{\xi} = f(\xi, v)$, where $v$ represents the input.

   (c) Linearize the model around the pendulum's stable point. Write down the state-space description and corresponding transfer functions.

   STRUCTURE YOUR ANSWER AS FOLLOWS:
   - What is the equilibrium state $\xi_0$ around which you linearize? Write down its numerical value.
   - Derive the linearized model. Clearly indicate the nonlinear equation(s) you start from, and show which assumptions/linearizations you apply to end up with a linear state-space model.
   - Derive the transfer functions of the linearized model. Show the time domain equation(s) you start from.

   (d) Apart from the gravitational acceleration $g = 9.81$ m/s$^2$, the pendulum length and damping coefficient are the only parameters in the model. Measure the length with a ruler. This length could also be derived from the pendulum's natural frequency. Perform an experiment from which you can deduce the natural frequency and therefore the pendulum length. Can you also derive the damping ratio and therefore the damping coefficient? Illustrate this experiment with a figure illustrating your measurements. Describe how you computed the required system parameters. In the remainder of this assignment, you may neglect the damping in the system.

   STRUCTURE YOUR ANSWER AS FOLLOWS:
   - Explain the experiment to derive an estimate of the pendulum's length and the damping ratio.
   - Write down the formulas that relate these two parameters to the quantities you can read from your experiment data. Then, give their numerical values (don't forget their units).
   - Can you also determine the damping coefficient?
   - Does the calculated length equal the measured length? Explain.

   (e) Discretize the linear and nonlinear system using a forward Euler method. In the remainder we focus on these discretized versions.

   STRUCTURE YOUR ANSWER AS FOLLOWS:
   - Discretize the linear system using a forward Euler method. Clearly show which formulas you use to transform your continuous-time model into a discrete-time one.
   - Discretize the nonlinear system using a forward Euler method. Clearly show which formulas you use to transform your continuous-time model into a discrete-time one.

2. Design and implement a linear Kalman filter. You can use the Arduino code template that is provided on Toledo from here: `CT-EKF-Pendulum`. Since this is a MIMO system, you will need to perform matrix manipulations during your calculations. If the template is not self-explanatory for you, you can find more information in the file `matrices.txt` in the `example` folder.

   (a) Design and tune a linear Kalman filter based on the linearized system equations. Use the encoders as position measurement and the pendulum encoder as angle measurement. The infrared front sensor is no longer used in this assignment. Briefly explain how you have chosen $Q$ and $R$. Implement the steps on the Arduino and apply a trapezoidal velocity profile. This trajectory is built-in in the Kalman filter template. Have a look at the file `trajectory_pendulum.txt` in the `example` folder of the template. Plot the evolution of the estimated states and their uncertainty, the measurements and the velocity input, and explain what you see in the plots based on the working principles of the Kalman filter. Then, repeat the experiment without using the wheel encoder as measurement for the estimation and compare the results with those of the first experiment. Explain your observations.

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   >
   > - Write down the linearized measurement equations.
   > - Write down the numerical values of the matrices $Q$, $R$ and the initial state estimate covariance $\hat{P}_{0|0}$ that you use. Don't forget their units! Motivate your choices.
   > - Implement the Kalman filter on your Arduino and apply a trapezoidal velocity profile as input. Make a separate plot of the evolution of every state in time together with its 95% confidence interval. Use the provided MATLAB function `plotstates` (see exercise session 5). To import your data from QRoboticsCenter for this specific assignment, you can call the method `KalmanExperiment.createfromQRC45()`.
   > - Since the first two states $x$ and $\theta$ are also measurements, plot these measurements together with their 95% confidence interval on top of the corresponding state estimates with their 95% confidence interval that you just plotted (on the same figure). To this end, use the provided MATLAB function `plotmeasurements` (see exercise session 5).
   > - How does the uncertainty of the two measurements compare to the uncertainty of the estimate of their corresponding state? Is this in accordance with the Kalman filter principles? Why (not)?
   > - How do the measurement equations change if you would not use the wheel encoder measurement in the experiment?
   > - Repeat the experiment, but do not use the wheel encoder measurement in your estimator anymore. Make again a separate plot of the evolution of every state in time together with its 95% confidence interval. Use the provided MATLAB function `plotstates` (see exercise session 5).
   > - Does the uncertainty evolve similarly as before? How does the wheel encoder measurement affect the estimation?

3. Design and implement an extended Kalman filter.

   (a) Design an extended Kalman filter based on the nonlinear system equations. Write down the measurement equations.

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   >
   > - Write down the measurement equations, i.e. look for the possibly nonlinear function $g$ so that the outputs are prescribed by $g(\xi, v)$.

   (b) Write down the Jacobian of the state and measurement equation.

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   >
   > - Clearly show how you derive the Jacobian matrix of the state equation.
   > - Clearly show how you derive the Jacobian matrix of the measurement equation.

(c) Implement the extended Kalman filter on the Arduino. Compare its convergence with the linear Kalman filter.

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Do you use the same numerical value for $Q$ as for the linear Kalman filter? Is there any reason to assume that the process noise is different for the extended Kalman filter?
- Do you use the same numerical value for $R$ as for the linear Kalman filter? Is there any reason to assume that the measurement noise is different for the extended Kalman filter?
- Write down the numerical values of the matrices $Q$, $R$ and the initial state estimate covariance $\hat{P}_{0|0}$ that you use. Don't forget their units! Motivate your choices.
- Implement the Kalman filter on your Arduino and apply a trapezoidal velocity profile as input. Make a separate plot of the evolution of every state in time together with its 95% confidence interval. Use the provided MATLAB function `plotstates` (see exercise session 5).
- Compare these plots with the plots you obtained for the linear Kalman filter. Are there differences? If so, where could you see them best? Does this make sense to you?

4. Design and implement a state feedback controller based on the linearized model.

(a) Design a linear quadratic regulator that makes a trade-off between penalizing deviations in the pendulum's mass position and actuation effort. Clearly indicate the objective that the LQR is minimizing.

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Write down the LQR objective, symbolically, for your specific (linearized) model. Clearly indicate the structure of your (LQR) $Q$ and $R$ matrices.

(b) Design a feedforward gain such that zero steady-state errors occur for a step reference in desired pendulum mass position.

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Draw the block diagram of the control configuration you use. Clearly indicate where you add the feedforward signal.
- Show how you symbolically calculate the feedforward gain such that there is no steady-state error for a step reference in the desired pendulum mass position.
- Can you interpret the value you obtained as feedforward gain?

(c) Implement the controller on the Arduino using your Kalman estimator. Apply a step in the reference for the pendulum mass position. Visualize in one plot the responses for different trade-offs between reference tracking and actuation effort. Plot the actuator signal as well.

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Apply the step reference for the pendulum mass position after implementing the Kalman filter + state feedback + feedforward gain. Then, change the feedback gain according to the different trade-offs in the LQR design objective. Plot, on one figure, the step response for the different trade-offs. Make sure it is clear which trade-off corresponds to which line in the figure.
- Make a plot of the requested actuator signal (= command to the velocity control loop) for this reference step for the different designs. Again, make sure it is clear which line corresponds to which design.

## 2.5 Assignment 5: estimation and control of a two-wheel driven cart

This assignment considers the cart mounted with the swivel wheel and with two infrared sensors. Because the cart has two separately driven wheels, it acts as a two-wheel drive (2WD) system that can move in the horizontal plane. The system under consideration is shown in Figure 4. The 2WD cart moves around in the world's $XY$ plane. The coordinates of the cart's (geometric) center point are denoted by $(x_c, y_c)$, and $\theta$, the angle between the $X$-axis and the cart's longitudinal axis, determines the cart's orientation. A local coordinate system $X'Y'$ is attached to the cart at its center. The goal of this assignment is to estimate the cart's position and orientation $(x_c, y_c, \theta)$ by means of a Kalman filter and to use this estimate to make the cart follow a reference trajectory specified in terms of $(x_{c,\text{ref}}, y_{c,\text{ref}}, \theta_{\text{ref}})$.

In this assignment you can again assume that the velocity control loop is ideal. To locate itself with respect to nearby objects, the cart is able to take distance measurements from two infrared sensors: one frontal, and one lateral (one side only). Without infrared measurements available, the cart's position can only be predicted based on the chosen model. This is known as *dead reckoning* and is subject to cumulative errors. When infrared measurements are available, a Kalman filter can correct the state estimation based on the first two statistical moments (mean and covariance) of states and measurements.

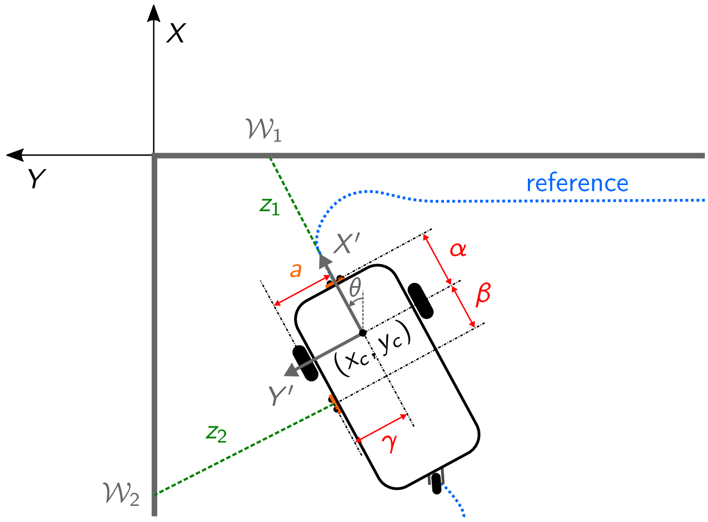



Figure 4: Schematic overview of the robot moving in the world and measuring the distance to the walls. $(x_c, y_c)$ are the coordinates of the robot's center in the world coordinate system $XY$. The robot's orientation is determined by $\theta$, the angle between the $X$-axis and the robot's longitudinal axis $X'$. The dimensions $a$ and $\alpha, \beta, \gamma$ are to be measured for the state equations and output equations respectively.

1. Model the system.

   (a) Write down the state equation. The input of the system can be chosen as $u = \begin{bmatrix} v \\ \omega \end{bmatrix}$, where $\omega$ [rad/s] denotes the rotational velocity of the robot around its center point $(x_c, y_c)$, and $v$ [m/s] is the robot's forward translational velocity (i.e. along its longitudinal axis $X'$). Choose the 2D pose of the cart, expressed in the world frame $XY$, as the state vector $\xi = [x_c, y_c, \theta]^T$ of the system. Use a forward Euler method to discretize the system.

   STRUCTURE YOUR ANSWER AS FOLLOWS:
   - Write down the relation between the forward velocity of the cart $v$, the rotational velocity of the cart $\omega$ and the velocity commands to both motors.
   - Write down the nonlinear continuous-time state-space equations ($\dot{\xi} = f(\xi, [v\ \omega]^T)$).
   - Discretize the nonlinear continuous-time state-space equations using a forward Euler method.

   (b) Write down the measurement equation for both of the infrared sensors ($z_1$, $z_2$). Assume that a (straight) wall is characterized by $\mathcal{W} = \{(x, y)\,|\,px + qy = r\}$. The dimensions $\alpha$, $\beta$, $\gamma$ and $a$ can be measured on your platform if required.

   STRUCTURE YOUR ANSWER AS FOLLOWS:
   - Symbolically express the nonlinear measurement equations ($z = g(\xi, [v\ \omega]^T)$).

2. Design and implement an extended Kalman filter. You can use the Arduino code template that is provided on Toledo from here: `CT-EKF-Swivel`. Since this is a MIMO system, you will need to perform matrix manipulations during your calculations. If the template is not self-explanatory for you, you can find more information in the file `matrices.txt` in the `example` folder.

   (a) Two sources of noise are incorporated in the model. Measurement noise is added to the output equation and is assumed to have a normal distribution with zero mean and a diagonal covariance matrix $R$. Process noise is added to the state equation and is assumed to have a normal distribution with zero mean and a diagonal covariance matrix $Q$. What are potential sources of process noise?

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   > - What are the potential sources of process noise and measurement noise in this system?

   (b) As you should have noticed, the system equations are nonlinear, which is the reason for using an extended Kalman filter to estimate the states. This filter linearizes the state and measurement equations around the current estimated state. Write down the Jacobian of the state equation and measurement equation.

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   > - Clearly show how you derive the Jacobian matrix of the state equation.
   > - Clearly show how you derive the Jacobian matrix of the measurement equation.

   (c) Implement the steps of the extended Kalman filter on the Arduino after tuning $Q$ and $R$. Briefly explain how you have chosen their values. Make sure the cart is located in $(-30, -20)$ cm (in the $XY$ frame, see Fig. 4). Then, apply a trajectory that makes a turn in the corner and stops at $(-15, -35)$ cm. This trajectory is built-in in the Kalman filter template. Have a look at the file `trajectory_swivel.txt` in the `example` folder of the template. The trajectory consists of two phases: one before the turn where both sensors are on and one during and after the turn where no sensors are used to collect measurements. Apply the feedforward inputs of this trajectory and plot the evolution of the estimated states and their uncertainty and the measurements for all phases. Compare these results for the different phases and explain your observations based on the working principles of the Kalman filter.

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   > - Write down the numerical values of the matrices $Q$, $R$ and the initial state estimate covariance $\hat{P}_{0|0}$ that you use. Don't forget their units! Motivate your choices.
   > - Implement the Kalman filter on your Arduino and cover the provided trajectory. Make a separate plot of the evolution of every state in time together with its 95% confidence interval. Use the provided MATLAB function `plotstates` (see exercise session 5). To import your data from QRoboticsCenter for this specific assignment, you can call the method `KalmanExperiment.createfromQRC45()`.
   > - How does the uncertainty evolve for the different states? Is there a difference between the results for the different phases?
   > - How do the measurement equations change if you switch off both sensors? How does the measurement equations change before and during/after the turn?
   > - Does the uncertainty evolve similarly as before? How does it change when the sensor(s) are switched off?

3. Design and implement a state feedback controller for following a position and orientation trajectory. This controller takes as input the tracking error

$$\hat{e} = \begin{bmatrix} x_{c,\text{ref}} - \hat{x}_c \\ y_{c,\text{ref}} - \hat{y}_c \\ \theta_{\text{ref}} - \hat{\theta} \end{bmatrix}$$

   As the system to be controlled is nonlinear and only design approaches for linear systems were addressed during the course, a heuristic control approach is proposed to design the state feedback controller. The control law is computed by first expressing the state error in the coordinate system $X'Y'$ attached to the cart (see Figure 4).

(a) Determine the rotation matrix that is required for the conversion of $(x, y, \theta)$, expressed in the world coordinate system $XY$, to $(x', y', \theta')$, expressed in the local coordinate system $X'Y'$. As this conversion depends on the orientation $\theta$ of the cart, use the estimate $\hat{\theta}_k$ to convert $\hat{e}_k$ to $\hat{e}'_k$.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Find the rotation matrix $R(\hat{\theta}_k)$ such that $\hat{e}'_k = R(\hat{\theta}_k)\hat{e}_k$.

(b) A static feedback matrix $K$ is proposed to compute $u$ from $\hat{e}'$ ($u = K\hat{e}'$) with the following structure:

$$K = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & k_\theta \end{bmatrix}$$

Explain why this structure makes sense.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Why does this structure make sense? What do the different components of this matrix stand for?

(c) Use a systematic approach to tune the values for $k_x$, $k_y$ and $k_\theta$.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Explain how you experimentally tune the three parameters of the feedback gain. What do you check in your experiments and how do you adapt the values accordingly?
> - Apply the reference trajectory to the closed-loop system and make a plot of the tracking errors of $x$, $y$ and $\theta$. Don't use any feedforward signals now (as opposed to the open-loop experiments of 2(c) where you exclusively applied feedforward signals).
> - Make a plot of the control signals $v$ and $\omega$ of the previous experiment.

(d) Finally, improve the tracking performance by adding the explicit feedforward signals again to the system's input: $u = u_{FF} + K\hat{e}'$. Illustrate the performance of the overall system by plotting tracking errors.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Plot the feedforward inputs $v$ and $\omega$ of your trajectory.
> - Compare these feedforward inputs with the control signals of 3(c).
> - Plot the contribution of the feedback control law to the 'complete' control signals (= feedback + feedforward).
> - Is the contribution of the feedback control law to the control signals large? Why (not)?
> - Plot the tracking errors of $x$, $y$ and $\theta$.