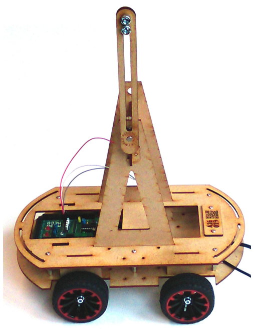
Control Theory – Assignments

1 Introduction

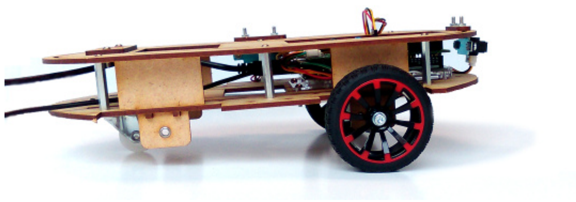
The evaluation of the course Control Theory is based on practical assignments in which you will apply several techniques taught during the course in order to control the cart presented in Figure 1. All carts have two separately driven wheels in the front. At the back, one type of cart is equipped with 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 Sunday October 12 2025** (23:59) in the provided Ultra Groups. After the registration deadline, 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 you encounter serious issues with your teammate (e.g. your team partner is not contributing) and you are not able to solve them, please report it immediately to Prof. Swevers or a TA. We will then take action to try to solve the issues. The ombudsman/woman can be involved to negotiate if this is desired. If the issues cannot be solved, the group will be split and each student has to continue individually. Deadline for reporting this type of issues is **November 10, 23.59**. This type of issues are not accepted anymore after this deadline. If your partner stopped following the course, please report this immediately as well. If for reasons mentioned above you are alone in a group, you still perform all the (remaining) assignments.



(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 according to the instructions below. In addition, we ask for the source codes of your implementations as well as a movie of one of your experiments.

The **deadline** for handing in your report plus source files and movie is **Friday December 19, 2025** (23:59). 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 on the first page of each PDF file. Save as `teamXX_2025_report_assignment_YY.pdf`, with XX your team number and YY the assignment number.

- Answer all questions of the assignments. All questions are assessed. General introductions and bridging texts in between the answers are not appreciated. The report should consist solely of your answers to the questions asked.
- 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*;
  - most of the variables and parameters have physical units, so indicate them;

  Ensure that the length of each answer adheres to the guidelines provided with each question. 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 oral discussion on your results, we would like to emphasize once more that your report should be **complete and self-explanatory**.
- Provide clear figures: do NOT use colored lines (we evaluate your reports based on a black and white printout) but use different line types to distinguish between signals within one figure. Use the line types in a consistent way for all figures within one report. All lines and differences between line types must be clearly 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.
- If your report is too long and/or figures are not clear, i.e. you consistently do not adhere to the specified rules, we will deduct up to 2 points out of 10 for the corresponding assignment.

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_2025_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_2025_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.

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): Monday October 27, 4.05 - 6.00 PM, Aud. A
- **assignment 2** (FB2): Friday November 07, 2.00 - 3.55 PM, Aud. C
- **assignment 3** (FB3): Friday November 21, 2.00 - 3.55 PM, Aud. C
- **assignment 4, 5** (FB4 & 5): Friday December 05, 10.35 - 12.30 AM, Aud. C
- **all assignments** (FB*): Monday December 15, 4.05 - 6.00 PM, Aud. C

In the first four feedback sessions, only questions concerning the specified assignment(s) that 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 TAs to solve them on your behalf. During the tutorial session, it will be explained and illustrated which type of questions are accepted and will be answered.

In case your cart is broken, please do not hesitate to pass by the electronic equipment service point as soon as possible such that they can fix whatever is broken. Do not return broken platforms without further notice!

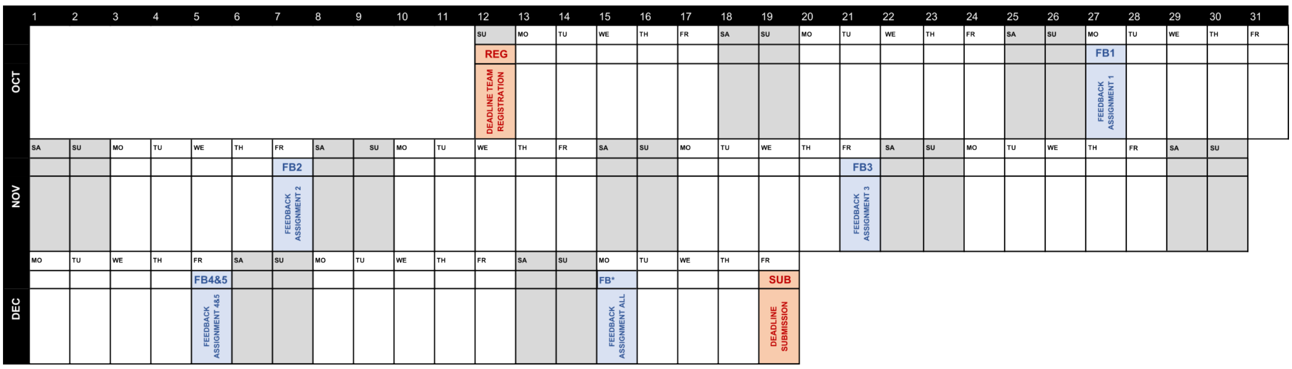


Figure 2: Timeline overview of deadlines and feedback session of the exam assignments.

2 Assignments

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

2.1 Assignment 1: identification of the cart

1. Select an appropriate model structure for the cart, for the dynamic relation between voltage applied to the DC-motors and the velocity of the cart measured by the wheel encoders. Each cart is equiped with two voltage controllerd DC-motors. The relation between the voltage and the velocity can be approximated as a linear model of a DC-motor with load and friction, assuming:
   - the same voltage is applied to both motors,
   - no wheel slip,
   - only forward and backward motions are considered (no turning)
   - cart mass and ground friction are lumped into the wheel inertia and friction respectively.

   (a) Select a discrete-time model structure that you will use for identification of the relation between motor voltage to rotational wheel velocity (**max. 1 page**).

   > Structure your answer as follows:
   >
   > 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 the continous-time transfer function relating voltage to velocity and denote all variable and parameters with their physical units. Then you present the discrete-time transfer function, clearly indicating the orders of the numerator and denominator and the number of delays.
   >
   > Briefly (max. 5 lines) explain how the discrete-time model was derived and briefly (max. 5 lines) motivate the choice(s) you made during this model structure selection. 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 two.

   (b) Write down the recursion expression (difference equation) and criterion that you use to estimate the model parameters (**max. 1 page**).

   > Structure your answer as follows:
   >
   > Write down the recursion expression(s) (difference equation(s)) specifically for your model(s) and write down the error criterion that you minimize to find the parameters.
   >
   > Write down the matrix formulation of the parameter estimation problem specifically for your/one of your model(s).

2. Identify the cart by exciting the motors while the cart is on the ground.

   (a) Which excitation signal do you apply? Why? (**max. half a page**)

   > Structure your answer as follows:
   >
   > Motivate the choice of the excitation signal you selected/designed.
   >
   > Plot your excitation signal. Do you apply it just once or do you repeat it? Why?

   (b) Estimate the model parameters and validate the accuracy/characteristics of the model(s) for both motors (**max. 1 page**)

Structure your answer as follows:

- Plot the measured response and response simulated with the identified model(s). Make sure that differences are visible (do not show the whole measured sequence, zoom in), also plot difference between measurement and simulation.
- Discuss the differences between simulation and measurement (transient and steady-state). Discuss transient and steady-state characteristics of the model(s) and compare them with those of the measurements/simulation. If you have identified multiple models, compare them and decide which model **structure/order** you prefer and will continue to use. Discuss the differences between the models of both motor and discuss wether these difference significant are significant or not (and motivate).

(c) Verify if data filtering can improve the identification (**max. 1 page**)

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, ...) and motivate your choices.
- Redo the identification after filtering your data.
- Plot the measured response and response simulated with the model identified with filtered data and the model of 2(b).
- Perform the same model validation as in 2(b) and conclude which model you prefer and will continue to use.

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 the frequency response method. Use the identified models (one for each motor) selected in part 2.

(a) What type of controller do you choose? Why? (**max. 1/4 page**)

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, cross-over frequency, integration time, ...). (**max. 1 page**)

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 that you enter in these formulas, and the numerical values you obtain when evaluating them.
- 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.
- Verify that the specifications are met in time and frequency domain by making appropriate plots.

(c) Is there a theoretical limitation on the closed-loop bandwidth that can be achieved? Is there a practical limitation? (**max. 1/4 page**)

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. Validate the controller. (**max. 1 page**)

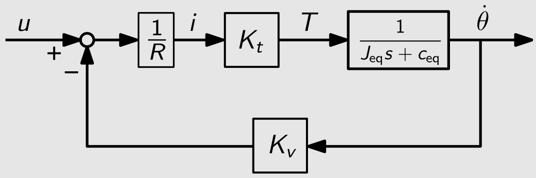
Structure your answer as follows:

- 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.
- Discuss the abovementioned comparisons in detail. Also compare the performance characteristics (transient and steady-state) with designed characteristics.

(b) Apply a constant force disturbance to the cart and validate your controller with respect to steady state performance. (**max. 1 page**)

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Draw the block diagram of your control configuration, including the force disturbance signal, and clearly indicate where the disturbance is entering the loop. Hint: the dynamics of your cart behave like a DC motor with a load, which are schematically represented in the figure below. u is the input voltage, i the motor current, T the motor torque, $\dot{\theta}$ the motor speed, R the armature resistance, J_{eq} the equivalent inertia projected on the motor shaft, the equivalent damping is c_{eq}, and $K_t = K_v$ is the motor constant.



- Apply the same velocity setpoint as in 2(a). Make the same plots as in 2(a).
- Is the controller still tracking the reference despite the disturbance? Explain why (not).

(c) Repeat the controller design and select a cross-over frequency of about $0.5Hz$. Validate this controller for the cart on a ramp and compare with the results of 2(b). (**max. 1 page**)

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Make the same plots as in 2(b) including the results of 2(b).
- Analyse and discuss the differences between the responses of both controllers. What is the contributions of the different parts of the controller in both cases?

2.3 Assignment 3: state feedback and state estimation

In this assignment you will control the position of the cart which is 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 state estimator. This estimator uses distance measurements from the frontal infrared sensor to correct its estimate. The 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.

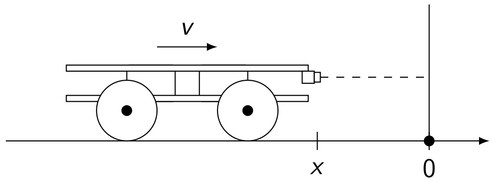


Figure 3: Cart measuring its distance to a wall.

For this assignment you can use the Arduino code provided on Toledo: `CT-StateEstimation_StateFeedback`. This code implements a working state estimator 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, but you still need to enter your designed values for K and L.

1. Design a state estimator and a state feedback controller using pole placement.

   (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) 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?

   (c) Design the state feedback controller gain K using pole placement.

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   >
   > Derive an expression for the pole of the closed-loop system (assuming full state feedback, without estimator) as a function of the sample time T_s and the state feedback gain K. How does it change as a function of K?
   >
   > Make a pole-zero map and draw how the pole location changes with varying K.
   >
   > Simulate a closed-loop step response for the same varying K. Plot all responses on the same figure, to compare the performance.
   >
   > Explain how the properties of the time response for a step reference relate to the pole locations and to the value of K. Can we choose K such that the closed-loop system becomes unstable?
   >
   > Which value do you take for K, and why? Don't forget the units!

   (d) Design the state estimator gain L using pole placement.

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Derive an expression for the pole of the closed-loop estimator as a function of the sample time T_s and the state estimator gain L. How does it change as a function of L? Can we choose L such that the estimator becomes unstable?
- How do you design appropriate values for L when using pole placement? Which trade-offs do you take into account?
- Give the value you obtained for L. Don't forget the units!

2. Implement and test a state estimator and a state feedback controller.

   (a) *For this question, include the state estimator, but do not include the controller.* Use a wrong initial estimation for the position in the `resetStateEstimator`-routine in `robot.cpp`. Plot the evolution of the position estimate together with the measured position by the infrared sensor. Do this for different values of L. Explain the different responses and link the experimental results with the theoretical expected results from 1(d).

   STRUCTURE YOUR ANSWER AS FOLLOWS:

   - Let the estimator start with a wrong initial estimate. Plot the measured distance together with the estimated distance for different values of L.
   - For which values of L do the estimates converge faster to the measurements? Explain using the analysis and trade-offs treated in 1(d).
   - How does the value of L influence the closed-loop estimator pole and the time response of the estimator?

   (b) *For this question, do not use the state estimator, but do include the controller. Directly use the position measurement to perform feedback on.* 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(c).

   STRUCTURE YOUR ANSWER AS FOLLOWS:

   - Plot, on one figure, (i) the (position) step reference and (ii) its measured response for different values of K.
   - Do the properties of these time responses correspond with the theoretical expectation for each of the values of K? Verify and explain this, by comparing with the pole locations and corresponding step responses analyzed in 1(c).
   - 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) *For this question, include both the estimator and the controller, such that the controller acts on the state estimates from the estimator.* 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. Perform an experiment in which the estimator starts while at the same time a step is applied as desired position signal. Plot the estimated position and measured position in one plot. Explain the observed behavior.

Structure your answer as follows:

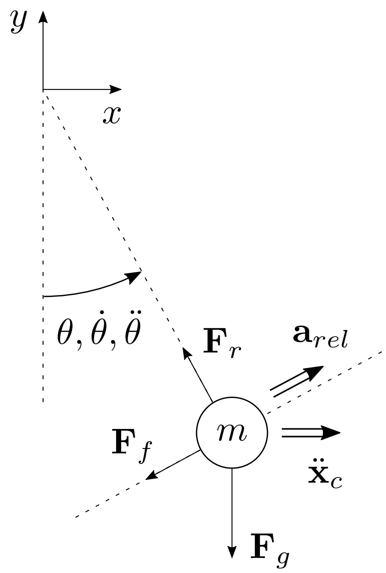
- Start from a good position estimate and, at the same time, apply a step reference as the desired position. Plot the measured distance together with the estimated distance.
- Start from a wrong position estimate and, at the same time, apply a step reference as the desired position. Plot the measured distance together with the estimated distance.
- Is the control performance satisfactory in both cases? Why (not)? If you were to design a state estimator using pole placement yourself, where would you put its closed-loop pole?
- Write down the poles of the full closed-loop system, including estimator and controller.
- What do you notice when examining the poles of the full closed-loop system, and is this what you would expect? Does the estimator performance influence the control performance? Could you deduce that from the closed-loop transfer function and the closed-loop poles of the complete control system including estimator and controller? How / why not?

2.4 Assignment 4: control of a pendulum

This assignment considers a moving 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 absolute position of the pendulum mass with respect to a starting position. 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 around its stable position.

1. Modelling the system formed by the pendulum on the cart:
   Below, we derive a theoretical continuous model of the velocity-steered cart with a pendulum. We assume that the pendulum has a length l and a mass m.

   Free body diagram of the pendulum mass:

   The following forces act on the pendulum:

   - $\mathbf{F}_f$: damping force (coming from the friction torque in the hinge). Its magnitude is proportional to $\dot{\theta}$.
   - $\mathbf{F}_r$: the force along the rod
   - $\mathbf{F}_g$: the gravitional force
   - $\ddot{\mathbf{x}}_c$: the acceleration of the cart, which is, from a relative motion point of view, the acceleration of the moving reference frame.
   - $\mathbf{a}_{rel}$: the relative acceleration (purely rotational) of the mass with respect to the moving reference frame. It equals $l\ddot{\theta}$.

   Writing down the force equilibrium along the line perpendicular to the pendulum rod results in the following equation:

   $$l\ddot{\theta} + \frac{c}{m}\dot{\theta} + g\sin\theta + \ddot{x}\cos\theta = 0$$

   Using the equation above, the cart with pendulum can be modeled as a system with:

   - Control input : v, unit m/s (equals $\dot{x}$)
   - Outputs : Pendulum angle $\theta\,[\text{rad}]$, cart position $x\,[m]$.

   Note that the damping of the system can be neglected as it is quite small, simplifying the dynamics.

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

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - What is the meaning of the different states x, θ 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.
> - Write down the measurement equations, i.e. look for the possibly nonlinear function g so that the outputs are prescribed by $y = g(\xi, v)$.

2. Derive a discrete linear model from the continuous-time nonlinear model. In this assignment two different approaches will be compared: (1) first discretizing and then linearizing the nonlinear model, and (2) first linearizing and then discretizing the linearized model.

(a) Normaly, the Zero-order hold method would be used to discretize the system, because this is also what physically happens in the system. However, because we choose to discretize first and linearize afterwards, it is difficult to use the ZOH-method due to the nonlinear dynamics. Therefore we resort to using a substitution method, selecting the forward-Euler method. Discretize the continuous-time nonlinear model using the forward-Euler method.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Discretize the nonlinear model. Clearly show which formulas you use to transform your continuous-time model into a discrete-time one.
> - Discretize the measurement equations. Clearly show which formulas you use to transform your continuous-time equations into a discrete-time one.

(b) Linearize the discrete-time model around a state ξ.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Explain your choice of the state ξ around which you linearize for the extended Kalman filter.
> - Derive the Jacobian of the discrete-time state and measurement equations. Write down 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.

(c) Now, obtain a discrete linear model by first linearizing the continuous-time nonlinear dynamics and then discretize using a zero-order hold which is now possible with the obtained linear dynamics.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - Linearize the continuous-time nonlinear dynamics around the previously selected equilibrium point.
> - Discretize the linearized continuous-time model using a zero-order hold approach. (hint: use the `c2d` MATLAB command)

(d) Analyse and compare the stability of both the obtained linear discrete time systems.

> STRUCTURE YOUR ANSWER AS FOLLOWS:
>
> - What do you notice, and explain possible differences you see. Explain why the zero-order hold approach is preferred for the rest of the assignment?

3. Design and implement a linear Kalman filter to estimate the states of the system. 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) Discuss the use of a linear or an extended Kalman filter.

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Explain the differences between a linear and extended Kalman filter. What model do they use in their predictions and corrections? Is there any reason to assume that there is a difference between the process noise or measurement noise for both Kalman filters?
- How does the linearization point differ for the linear and extended Kalman filter? Are they equivalent? What is the expected effect on the performance of these estimators?
- Discuss where each type of Kalman filter is more appropriate to use. Are they both equally performant or necessary for each type of model?

(b) 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.

STRUCTURE YOUR ANSWER AS FOLLOWS:

- What are the potential sources of process noise and measurement noise specifically for the setup and model in this assignment? Which approximations have an influence on which type of noise?

(c) Design and tune a linear Kalman filter based on the linearized system equations using the zero-order hold method. Use the encoders as position measurement and the pendulum encoder as angle measurement. The infrared front sensor is no longer used in this assignment. Choose a reasonable value for the initial state estimate covariance $\hat{P}_{0|0}$. 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. Explain how you have chosen Q and R. Analyze and discuss the effect of varying Q and R on the behavior of the linear Kalman filter.

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Explain the meaning of Q and R. How do you choose reasonable values for them? Which tradeoffs do you take into account?
- Choose sensible values for the initial state estimate covariance $\hat{P}_{0|0}$ and explain your choice. Do not forget the units!
- Implement the Kalman filter on your Arduino and apply a trapezoidal velocity profile as input. Plot, on one figure, the evolution per state in time for varying Q and R values (max. 4 combinations). The number of figures is thus equal to the amount of states. To import your data from QRoboticsCenter for this specific assignment, you can call the method `KalmanExperiment.createfromQRC45()`. (*Hint:* Play around with the ratio of Q and R.)
- What is the influence of Q and R on the behaviour of the linear Kalman filter? Does this correspond to the tradeoffs you expected? For which $\frac{Q}{R}$ ratios do the estimates converge faster to the measurements? Choose numerical values for Q and R and motivate your choice based on the plots. Do not forget their units!

(d) Discuss the uncertainty on the estimated states with and without the wheel encoder measurements. Plot the evolution of the estimated states and their uncertainty, and the measurements 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:

- Plot the measurements of x and θ 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?

4. Design and implement a state feedback controller (LQR) based on the linearized model around ξ^*.

   (a) Design a feedforward gain such that zero steady-state errors occur for a step reference in desired absolute 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?

   (b) Implement the controller on the Arduino using the estimates of your Kalman filter. Apply a step in the reference for the pendulum mass position. Visualize in one plot the responses for different trade-offs between Q and R. Plot the actuator signal as well. Analyze and discuss the choice of Q and R based on the observed behavior of the controller.

   Structure your answer as follows:

   - What is the meaning of Q and R in the LQR design objective?
   - Give a symbolic description of the structure of Q and R, give their dimensions and discuss why this structure makes sense.
   - 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 varying trade-offs between Q and R (again max. 4 combinations). Plot, on one figure, the step response for the different trade-offs. Make sure it is clear which values 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.
   - Explain the effect of the different trade-offs on the observed controller behavior based on these plots. How does it influence the convergence speed of the controller? Are there limitations to be taken into account?
   - Report the chosen numerical value of Q and R and motivate your final choice. Do not forget their units!

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 θ, 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, θ) by means of a Kalman filter and to use this estimate to make the cart follow a reference trajectory specified in terms of $(x_{c,\text{ref}}, y_{c,\text{ref}}, \theta_{\text{ref}})$. The trajectory involves three phases: firstly, a straight line has to be travelled at a constant forward velocity ($v_{ref} = 2$ cm/s) after which a turn has to be made at that same forward velocity with a constant rotational velocity ($\omega_{ref} = 0.44$ rad/s). Finally, a second straight line at that same forward velocity has to be tracked.

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.

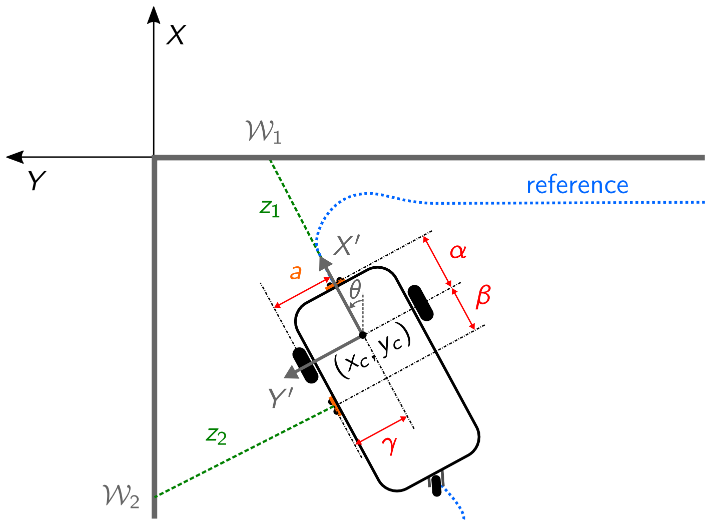


Figure 4: Schematic overview of the robot moving in the world and measuring the distance to the walls. (x_c, y_c) are the coordinates of the robot's center in the world coordinate system XY. The robot's orientation is determined by θ, the angle between the X-axis and the robot's longitudinal axis X'. The dimensions a and α, β, γ 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 ω [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.

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

   (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 α, β, γ and a can be measured on your platform if required.

   STRUCTURE YOUR ANSWER AS FOLLOWS:
   - Symbolically express the nonlinear measurement equations ($z = h(\xi, [v\ \omega]^T)$) in terms of α, β, γ and a, generally and specifically for the situation depicted in Figure 4.

2. Derive and discuss the use of an extended Kalman filter. Discretize your nonlinear continuous-time model and measurement equations using the forward Euler discretization scheme. Choose a state around which to linearize and linearize the discrete-time model. Discuss when and why you would choose a linear or extended Kalman filter and their respective differences.

   (a) Discretize the continuous-time nonlinear model using the forward Euler method.

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   > - Discretize the nonlinear model. Clearly show which formulas you use to transform your continuous-time model into a discrete-time one.
   > - Discretize the measurement equation. Clearly show which formulas you use to transform your continuous-time model into a discrete-time one.

   (b) Linearize the discrete-time model around a state ξ^*.

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   > - Explain the meaning and choice of the state ξ^* around which you linearize for the extended Kalman filter.
   > - Discuss the differences between the choice of this state ξ^* for the linear and extended Kalman filter. Are they equivalent? What is the effect on the performance of these estimators?
   > - Derive the Jacobian of the discrete-time state and measurement equation for the situation depicted in Figure 4. Write down 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.

   (c) Discuss the use of a linear or an extended Kalman filter.

   > STRUCTURE YOUR ANSWER AS FOLLOWS:
   > - Explain the differences between a linear and extended Kalman filter. What model do they use in their predictions and corrections? Is there any reason to assume that there is a difference between the process noise or measurement noise for both Kalman filters?
   > - Discuss where each type of Kalman filter is more appropriate to use. Are they both equally performant or necessary for each type of model?

3. 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 specifically for the setup and model in this assignment? Which approximations have an influence on which type of noise?

   (b) Design and tune an extended Kalman filter on the Arduino. Choose a reasonable value for the initial state estimate covariance $\hat{P}_{0|0}$. Implement the steps of the extended Kalman filter on the Arduino. 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. Explain how you have chosen Q and R. Analyze and discuss the effect of varying Q and R on the behavior of the extended Kalman filter.

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Explain the meaning of Q and R. How do you choose reasonable values for them? Which tradeoffs do you take into account?
- Choose sensible values for the initial state estimate covariance $\hat{P}_{0|0}$ and motivate your choice. Do not forget the units!
- Implement the Kalman filter on your Arduino and cover the provided trajectory. Plot, on one figure, the evolution per state in time for varying Q and R values (max. 4 combinations). The number of figures is thus equal to the amount of states. 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()`. (*Hint:* Play around with the ratio of Q and R.)
- What is the influence of Q and R on the behaviour of the extended Kalman filter? Does this correspond to the tradeoffs you expected? For which $\frac{Q}{R}$ ratios do the estimates converge faster to the measurements? Choose numerical values for Q and R and motivate your choice based on the plots. Do not forget their units!

(c) Discuss the uncertainty on the estimated states before and after the turn. Plot the evolution of the estimated states and their uncertainty, and the measurements and explain what you see in the plots based on the working principles of the Kalman filter.

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Plot the evolution of the state estimates in time with their 95% confidence interval. To this end, use the provided MATLAB function `plotmeasurements` (see exercise session 5).
- How does the uncertainty evolve for the different states? Is there a difference between the results for the different phases of the trajectory? Explain your answer based on the principles of the Kalman filter.
- How do the measurement equations change if you switch off both sensors?
- How do the measurement equations change before and during/after the turn? Consider hypothetically that the sensors would not be turned off. Describe in words what would change, a detailed derivation is not required.

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

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

In this part of the assignment, you will design, implement and tune an LQR controller based on a linearized model.

To design an LQR tracking controller, the error dynamics are derived in the local cart coordinate system X'Y': $\dot{e}' = R(\theta_c)(\dot{\xi}_{ref} - \dot{\xi}_c) + \dot{R}(\theta_c)(\xi_{ref} - \xi_c)$. In this assignment, we neglect the rotation over time of the cart frame $R(\theta_c)$ and assume that this frame can be considered static. Expanding the error dynamics using the previously derived cart dynamics $\dot{\xi}_c = f(\xi_c, u)$, the following Jacobians of the error dynamics can be derived:

$$\frac{\partial \dot{e}'}{\partial \xi_c} = \begin{bmatrix} 0 & 0 & v_{ref}\sin(\theta_{ref} - \theta_c) \\ 0 & 0 & -v_{ref}\cos(\theta_{ref} - \theta_c) \\ 0 & 0 & 0 \end{bmatrix} \quad \frac{\partial \dot{e}'}{\partial u} = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}$$

As linearization point, we choose a point on the trajectory where all states and inputs of the cart equal the desired states and inputs of the reference trajectory ($\xi_{ref} = \xi_c$ and $\dot{\xi}_{ref} = \dot{\xi}_c$). This leads to the following continuous-time state equation matrices:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -v_{ref} \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}$$

After discretization with the forward Euler method, the discrete-time state equation matrices are:

$$A_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -T_s v_{ref} \\ 0 & 0 & 1 \end{bmatrix} \quad B_d = \begin{bmatrix} -T_s & 0 \\ 0 & 0 \\ 0 & -T_s \end{bmatrix}$$

Since we do not have access to the actual states of the system, all above equations in practice will contain the state estimate $\hat{\xi}$ instead of ξ. Except for A_d and B_d, all equations above are continuous-time equations.

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

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Find the rotation matrix $R(\hat{\theta}_{c,k})$ such that $\hat{e}'_k = R(\hat{\theta}_{c,k})\hat{e}_k$.

(b) Derive the structure of the feedback matrix K using Matlab.

STRUCTURE YOUR ANSWER AS FOLLOWS:

- Symbolically write down the structure of Q and R you use and explain your choice.
- What are the sizes of Q and R and what is the meaning of their components?
- Write the structure of the feedback matrix K you obtain from Matlab. (*Hint:* use the *dlqr* command.)
- Does this structure make sense? Why and what do the different components of this matrix stand for?

(c) Use a systematic approach to tune Q and R.

STRUCTURE YOUR ANSWER AS FOLLOWS:

- What is the meaning of Q and R in the LQR design objective?
- Apply the reference trajectory to the closed-loop system and make a plot of the tracking errors of x, y and θ for varying Q and R (again max. 4 combinations). Do not 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 ω of the previous experiments.
- Explain the effect of the different choices of Q and R on the observed controller behavior based on these plots. How does it influence the convergence speed of the controller? Are there limitations to be taken into account?
- Report the chosen numerical value of Q and R and motivate your choice. Do not forget their units!