Data-driven model learning for LQR control of a dual-rotor helicopter

1 Purpose of the lab

The general methodology for the design of a feedback control is detailed in Figure 1.1[1].

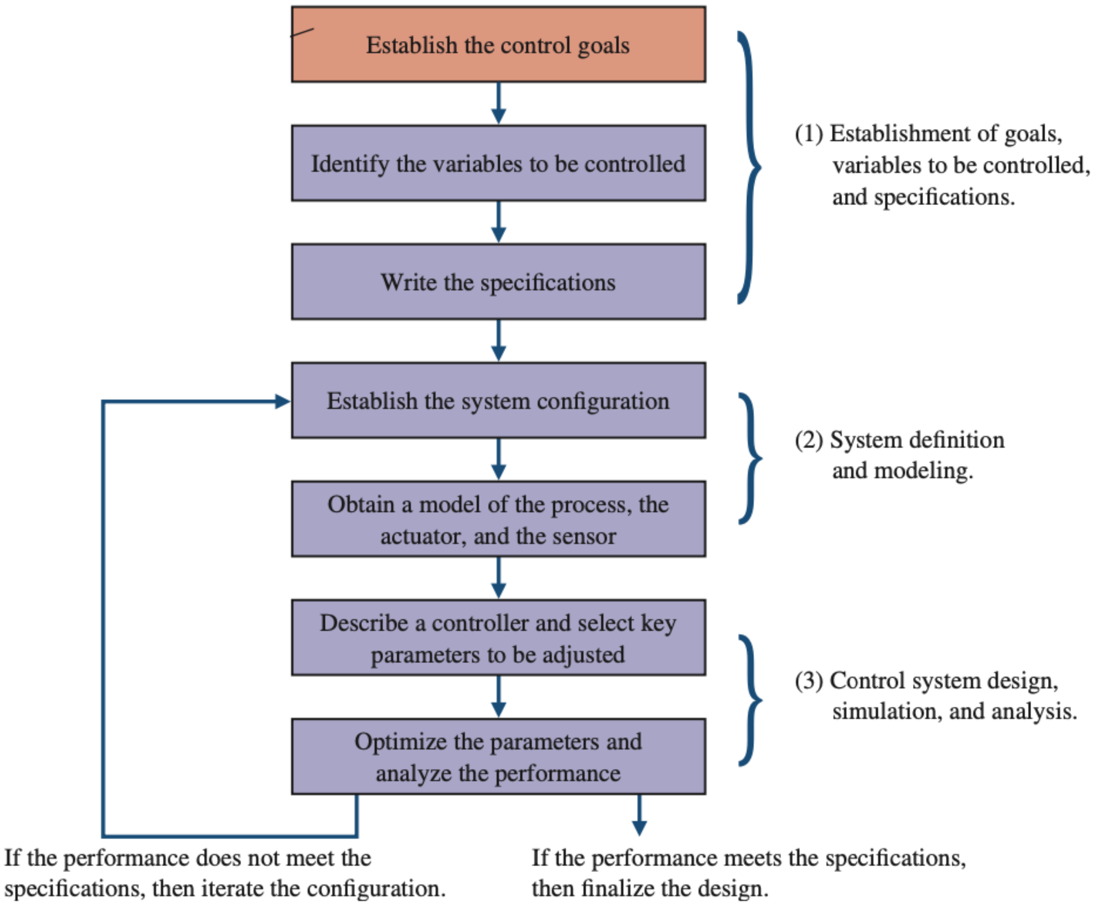

Figure 1.1: The control system design methodology

From Figure 1.1, the controller design problem can summarized as follows:

Given a model of the system to be controlled (including its actuators and sensors) and a set of design goals, find a suitable controller (or determine that none exists!).

[1] From Richard Dorf and Robert Bishop, Modern Control Systems, Pearson, 2022

As with most of engineering design project, the design of a feedback control system is an iterative and non-linear process.
A successful designer must consider the underlying physics of the plant under control, the control design strategy, the controller design architecture (that is, what type of controller will be employed), and effective controller tuning strategies. In addition, once the design is completed, the controller is often implemented in hardware, and hence issues of interfacing with hardware can appear. When taken together, these different phases of control system design make the task of designing and implementing a control system quite challenging.

One key step in the control system design methodology is to obtain a model of the system. This system identification step is explored here where your task is to determine a linear model of a dual-rotor helicopter using data-driven system identification techniques. The identified model will then be used to design a LQR controller.
Even if you have access to powerful parameter estimation/optimization methods for state-space linear model identification such as the SSEST routine for example, there are usually many critical choices that must be made by the user.
The lab is designed to gain experience in the following aspects:

1. How to design inputs to excite a 2 input-2 output system.
2. How to choose a suitable order for the state-space linear model.
3. How to estimate the state-space model parameters from the collected data.
4. How to evaluate the quality of the identified model from the physical perspective.
5. How to use the identified model to design a LQR control.

2 Layout of the lab

In this mini-project, you will use the CONTSID and the System Identification toolboxes and so basic knowledge about how to use these toolboxes is assumed.
Building a parametric model from sampled data involves usually the following steps:

1. Carry out simple step tests in open loop to gain physical insight about the complexity of the system.
2. Experiment design, e.g. choose suitable inputs to get informative data.
3. Carry out the experiment and collect data from the system.
4. Examination and preprocessing data (selection of informative data segments, removal of trends and outliers, etc.).
5. Model structure selection.
6. Estimation of the model parameters.
7. Model validation.
8. Interpretation in a physical manner the identified model.

Normally, the different steps are repeated until you are satisfied with the results of the model validation.
The tasks can be split into 3 main parts:

1. Data collection and analysis.
2. Model identification and validation.
3. Use of the model to design and simulate and implement a LQR controller.

In practice, the data collection part, including the choice of the input excitation signals, is often the most time-consuming and often most crucial part. With a good set of measured informative data, it is relatively simple to quickly develop and evaluate different models.

3 Hardware required

The required hardware for this lab includes:

- a Quanser AERO system configured as a dual-rotor helicopter with the low-efficient blades.

4 Download of the files required for the lab

1. Download the zipped file `Lab_AERO_helicopter.zip` from the course website. Save and unzip it in the c:/temp folder.
2. Start Matlab.
3. By clicking on the browse for folder icon , select the folder that contains the files needed for this lab so that it becomes the current folder in Matlab.

5 The dual-rotor helicopter

5.1 The Quanser AERO system configured as a dual-rotor helicopter

The AERO system from Quanser can be configured as a dual-rotor helicopter as shown in Figure 1.2. The front rotor that is horizontal to the ground predominantly affects the motion about the pitch axis while the back or tail rotor mainly affects the motion about the yaw axis.

Because the rotors on the Quanser Aero are the same size and equidistant from each other, the tail rotor also generates a torque about the pitch axis. As a result, both the front and back/tail rotors generate a torque on each other. This dynamic coupling between the pitch and yaw axes is only seen when using the low-efficiency blades that have been installed on the Quanser AERO for this lab. A set of more efficient blades can be installed.

Figure 1.2: The Quanser AERO system configured as a dual-rotor helicopter

5.2 Inputs/outputs of the dual-rotor helicopter

The dual-rotor helicopter is a two-inputs two-outputs system where the two outputs are:

- the pitch angle $\theta(t)$ in degrees ($^\circ$) ;

- the yaw angle $\psi(t)$ in degrees (°).

while the two inputs are

- the voltage $v_p(t)$ applied to the pitch rotor in V ;
- the voltage $v_y(t)$ applied to the yaw rotor in V.

The objective of this mini-project is to control the angular position of both pitch and yaw axis independently. Thus the measured pitch and yaw angles are the two outputs, and the voltages sent to the rotor DC motors are the two control inputs. Since the interaction between the two rotors causes strong cross coupling, controlling the pitch and yaw axis positions independently requires the design of a MIMO controller. In this project an LQR controller will be designed for this purpose.

The physical modelling through the derivation of equations of motion for the 2 DOF helicopter has been realized by QUANSER. It is presented in the pdf file available on the course website.

From this physics-based modelling, grey-box models written in both transfer function and state-space forms can be obtained. As realized by Quanser engineers, the physical parameters of the grey-box model can be determined from a few simple step-like tests carried out on the AERO system.
Your task will be first to evaluate the obtained grey-box model by comparing its outputs with the measured outputs for steps sent separately to each input.
The controller design will be carried out in continuous-time domain from an identified continuous-time linear state-space model, but the controller will be implemented digitally with a sampling time of T_s =2 ms.

6 Data-driven model learning

6.1 Design of excitation signals

A proper design of excitation signals requires some knowledge about the system.

6.1.1 Basic step-like tests in open loop

To get a feeling of the system and of its dynamics, proceed as follows:

- Connect the AERO to the PC.
- Open the file `Aero_basic_tests.slx`. Set the pitch rotor voltage to 10V and click on `Monitor & Tune` to run the test. Observe the pitch and yaw angle responses.
- Modify the pitch rotor voltage and observe the system responses. Stop the test.
- Set the pitch rotor voltage to 0 and set now the yaw rotor voltage to 10V and run the test by clicking on `Monitor & Tune`. Observe the pitch and yaw angle responses.
- Modify the yaw rotor voltage and observe the system responses. Stop the test.

6.1.2 Step responses in open loop

To get further information about the system dynamic, it is usual to apply single steps separately to each input.

Each axis is allowed to move freely. The response of each axis is examined when applying a voltage to the other axis.

6.1.3 Pitch and yaw responses from individual step on the pitch rotor

1. Open the file `Aero_step_responses_from_pitch.slx` in Simulink.
2. Click on `Monitor & Tune` to run the step test that lasts about 20 secondes where a step of 15V is applied to the pitch rotor. The response in both the pitch and the yaw axis are observed and recorded.

You should get a response similar to the one displayed in Figure 1.3. The data have been stored in the Matlab file `aero_resp_from_pitch_step_HG.mat`.

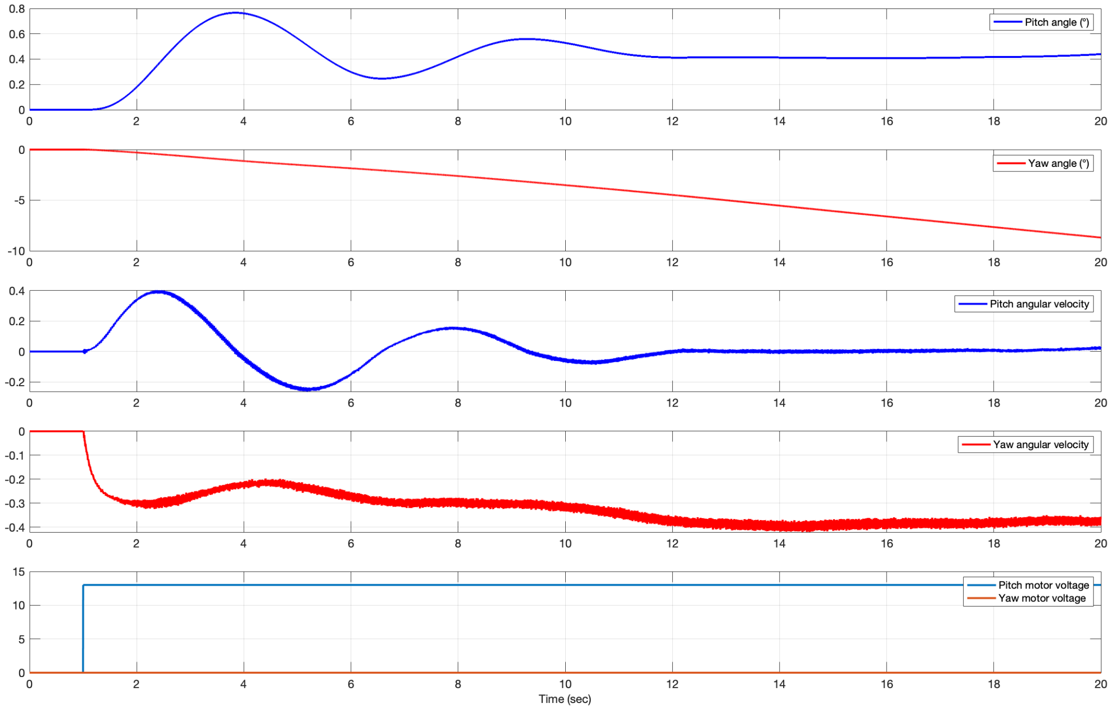

Figure 1.3: Step response of pitch and yaw angles and their derivatives from pitch motor voltage

Figure 1.3 can be reproduced by executing the file `Aero_data_plot.m`. Change the file name at the top of the .m file to get the plot of your data.

6.1.4 Pitch and yaw responses from individual step on the yaw rotor

1. Open the file `Aero_step_responses_from_yaw.slx` in Simulink.
2. Click on `Monitor & Tune` to run the step test that lasts about 20 secondes where a step of 13V is applied to the yaw rotor. The response in both the pitch and the yaw angular position and velocity are observed and recorded.

You should get a response similar to the one displayed in Figure 1.4. The data have been stored in the Matlab file `aero_resp_from_yaw_step_HG.mat`.

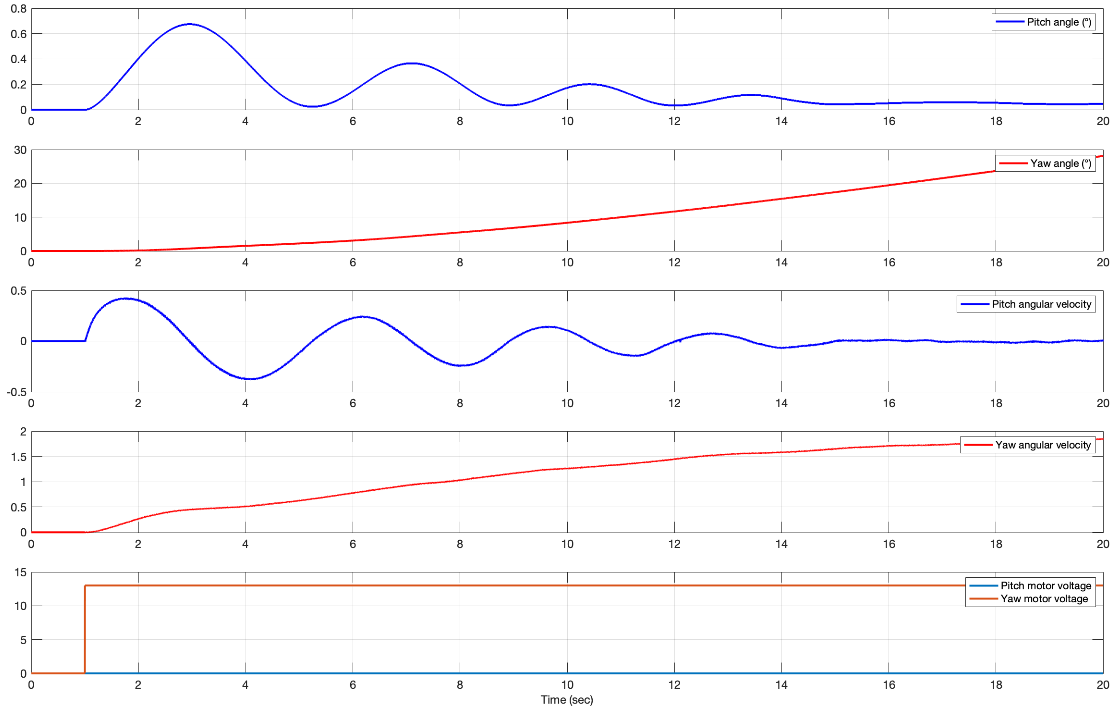

Figure 1.4: Step response of pitch and yaw angles and their derivatives from yaw motor voltage

Figure 1.4 can be reproduced by executing the file `Aero_data_plot.m`. Change the file name at the top of the .m file to get the plot of your data.

6.1.5 Validation of the grey-box model from Quanser

Quanser has run some simple step-like tests to identify the different physical parameters that appear in their linearized grey-box model. They do not however show any validation results where the grey-box model outputs are compared to the measured outputs.
The linear grey-box model determined by Quanser engineers is given in the `quanser_aero_state_space.m` file.
Use the step responses stored in the .mat files to test the grey-box model capacity to reproduce the measured outputs.
What do you conclude about the quality of the given physics-based models ?
Would you use the model to design and implement a LQR control ?

6.1.6 Main characteristics of the system

From the step responses, estimate for each output the main characteristics in terms of steady-state gain and rise time t_r.
These main characteristics should be useful to design new experiments to get a better model.

6.2 New data collection

New inputs should be designed keeping in mind that the experiment should be carried out under conditions that are similar to those under which the model is going to be used.
Note that the effects of system nonlinearities depend heavily on the input profile, the amplitude and frequency content of the excitation signals.

Designing inputs such that the linearity of the system can be ensured and tested from the responses, is necessary.

6.3 Design of square wave excitation inputs

Square wave inputs are first recommended. They present the following advantages:

- the amplitude for both motor voltages can be set different in an easy way;
- the period of one square can be set from a previous simple step response so that the outputs can reach their steady-state values;
- the two square waves can be shifted to make them not too much correlated and so that each step on one motor can be clearly observed on both outputs;
- they are periodic and as such the linearity of the system response can be easily observed and tested. If the system operates in its linear range, the response should also be periodic and similar for each period.

Choose the amplitude and the period of the two square waves. Note that an amplitude larger than 15V may result to a saturation of the pitch angle. Shift your square waves to make them not too much correlated. Once you have designed the two uncorrelated square wave inputs, proceed as follows to carry out the experiment and collect data from the system:

1. Modify the file `Aero_step_responses_from_pitch.slx` in Simulink to send your two uncorrelated square wave inputs. You can use a Signal Generator block. A time-delay block can be used to shift one of the square wave input.
2. Set the amplitude and the period of the two square waves. Note that an amplitude larger than 15V may result to a saturation of the pitch angle.
3. Specify the name of the .mat file where the data will be recorded for further analysis.
4. Click on `Monitor & Tune` to run the test where two different square waves are applied to both pitch and yaw rotor. Observe the pitch and the yaw axis responses that will be saved in a .mat file.

For model validation purpose, repeat the experiment for the same or another square wave excitation signals. From Figures **??**, the following comments should be made:

- the yaw angle has a linear trend that can be physically interpretable since the yaw response has an integrator. We should be able to take this into account in the identification steps;
- the pitch and yaw angle (and their velocities) responses are periodic;
- no strong non-linear effects are visible from the plots.

If the last two comments are not observed, revise the setting of two uncorrelated square wave inputs (amplitude and period) and re-do the experiments.

6.3.1 Data pre-treatment

Once your data for the chosen inputs have been recorded, remove:

- Possible transients needed to reach desired operating point.
- Mean values and possible trends of input and output signals (use `detrend(y,0) or detrend(y,1)` in Matlab).
- Outliers ("obvious erroneous data points"), if any clearly visible from time-domain response observation.

6.3.2 Model order selection

The model to be identified will take the form of a state-space model in canonical form. The only choice to be made is to select the order n for the state-space model. You can also use the physics-based modelling to help you to make your final choice.

6.3.3 Model parameter estimation

The next step is then to choose an appropriate model order and to estimate the parameters of the state-space model.
Note that it is possible to estimate directly canonical state-space models when you use the `ssest` routine.

To specify an continuous-time observability canonical form, use the 'Form' name-value pair input argument, as follows:
`m = ssest(data,n,'Form','canonical')`
Type `doc ssest` for further information. See also `State-Space Realizations` in the Matlab help center.

6.3.4 Model validation

For the purpose of model validation, you can proceed as follows:

1. Compare the model simulation responses with the outputs of the real system with the **e**stimation data.
2. Compare the model simulation responses with the outputs of the real system with the **v**alidation data.

The process of testing the model against "new" data (ie, not the same data as used for the estimation) is called cross-validation and is the most effective way to determine if the model is working correctly or not. Generally, you should expose the model to as many different tests as possible to minimize the risk of accepting an incorrect (bad) model.

Choose your best identified canonical state-space model from this square waves experiments.

6.3.5 Model interpretation from a physical perspective

Your identified state-space model is a black-box model and it is important to **interpret it in a physically sense**.
For this purpose, proceed as follows:

1. Plot the step responses of your identified state-space model. Examine the responses. Do they make sense ?
2. Convert your state-space model into transfer function matrix (see `ss2tf`). Determine the main characteristics of each transfer function.
3. Make sure you to figure out what are the states of your identified state-space model obtained by the `ssest` routine. This will impact the computation of the LQR gain matrix and state feedback controller.

6.4 Design of PRBS excitation inputs

You can skip this section if your are running short on time.

Pseudo-random binary sequence (PRBS) represents a standard option for exciting a system since its spectrum is close to a white noise. See for example the command `prbs` in the CONTSID toolbox which can be used to generate PRBS.

A so-called maximal-length Pseudo-Random Binary Sequence or PRBS is a succession of rectangular pulses of varying length.They are generated by closed-loop shift registers (hardware or software) having n cells. Their main characteristics are the following:

- The duration of the PRBS is $L = (2^n - 1) \times p \times T_s$ where n is the number of cells, p is a coefficient such that the PRBS signal is constant over intervals of minimum length p, and $T_s = 0.002$s is the sampling period. The PRBS is designed from the constraint below:
  - $3 \leq n \leq 18$
  - The duration of the longest pulse (or step) $n \times p \times T_s$ must be larger than the rise time t_r of the system: $n \times p \times T_s > t_r$
- The amplitude of the PRBS must be small to ensure that the system stays in the linear operating range. But, it should large to sufficiently excite the system and be higher than the noise level.

To generate the two uncorrelated PRBS inputs, proceed as follows:

- Open the file `generate_prbs.m` where the `prbs` routine from the CONTSID toolbox is used.
- Specify the design parameters n and p for the two prbs inputs along with the amplitude for the yaw and pitch rotor voltages.
- Run the file to generate the uncorrelated PRBS. Observe their plots.

Once you have designed the two uncorrelated PRBS inputs, proceed as follows to carry out the experiment and collect data from the system:

1. Load in the Matlab workspace the generated PRBS.
2. Open the file `q_aero_PRBS_response.slx` in Simulink.
3. Set the amplitude of the two PRBS. Note that an amplitude larger than 15V may result to a saturation of the pitch angle.
4. Click on `Monitor & Tune` to run the test where two different PRBS are applied to both pitch and yaw rotor. Observe the pitch and the yaw axis responses that will be saved in a mat file.

For model validation purpose, repeat the experiment for the same or another PRBS excitation signals.

6.4.1 PRBS data examination

Once, the data have been collected, plot and examine them carefully to identify possible problems.
Data where the friction effects are non-negligible should not be used in the linear model identification procedure.

From PRBS response examination, answer the following question:

- Are PRBS response data appropriate for determining a linear model of the dual-rotor helicopter ?

7 LQR control design and closed-loop simulation and implementation

From the analysis carried out in the data-driven model learning part, select the best canonical state-space model which will serve as a basis for linear controller design.
The task is now to design an LQR controller that stabilizes the plant and allows setpoint tracking.

Assuming all the states are known we can design an optimal Linear Quadratic Regular (LQR) based full state-feedback control to:

- Control the position of the dual-rotor helicopter to a desired square wave sent to the pitch angle setpoint.
- Control the position of the dual-rotor helicopter to a desired square wave sent to the yaw angle setpoint.

7.1 A video to start with

If you need a refresher about LQR control, watch Brian Douglas' video:
What Is Linear Quadratic Regulator (LQR) Optimal Control? | State Space, Part 4
`www.youtube.com/watch?v=E_RDCFOlJx4&list=PLn8PRpmsu08podBgFw66-IavqU2SqPg_w&index=4`
Listen carefully to the different comments given by Brian Douglas !

7.2 LQR control

Linear Quadratic Regulator (LQR) theory is a technique that relies on a linear state-space model for finding the controller parameters to control the dual-rotor helicopter.
Given that the equations of motion of the dual-rotor helicopter system can be described in linear state-space form. The LQR algorithm computes a control law u such that the performance criterion or cost function

$$J = \int_0^\infty \left(x_{ref} - x(t)\right)^T Q \left(x_{ref} - x(t)\right) + u(t)^T R u(t) dt \tag{1}$$

is minimized. The design weighting matrices Q and R hold the penalties on the deviations of state variables from their setpoint and the control actions, respectively.
When an element of Q is increased, therefore, the cost function increases the penalty associated with any deviations from the desired setpoint of that state variable, and thus the specific control gain will be larger. When the values of the R matrix are increased, a larger penalty is applied to the aggressiveness of the control action, and the control gains are uniformly decreased.
In our case the full state vector x is defined

$$x = \begin{bmatrix} \theta & \dot{\theta} & \psi & \dot{\psi} \end{bmatrix}^T. \tag{2}$$

This control law is a full state-feedback control and is illustrated in Figure 1.5.

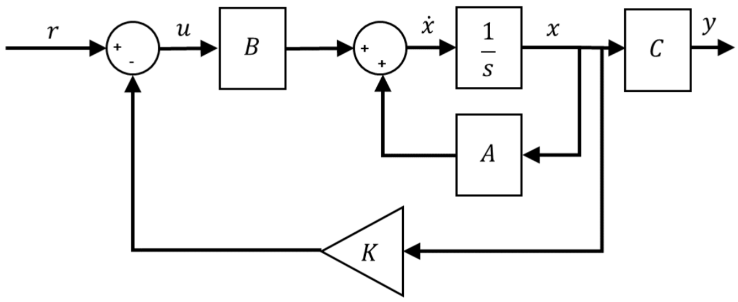

Figure 1.5: Block diagram of standard state-feedback control

7.3 Performance analysis in simulation with Simulink

1. Define your continuous-time state-space model.
2. Compute the gain matrix K that results from your LQR design.
3. Open the file `s_aero_2dof_lqr_control.slx` in Simulink.
4. Click on `Run` and observe the simulated pitch and yaw responses in servo control as well as the motor voltage signals.

7.4 Performance assessment of the LQR control on the dual-rotor helicopter

Test your LQR control on the experimental AERO system:

1. Define your continuous-time state-space model.
2. Compute the gain matrix K that results from your LQR design.
3. Open the file `q_aero_2dof_lqr_control.slx` in Simulink.
4. Click on `Monitor & Tune` and observe the pitch and yaw responses in servo control as well as the motor voltage signals.

Recommendation for your report

This mini-project is evaluated from a written report which should be readable independently. It can be written in French or in English but do not mix both languages.

Your report should be a scientific document. As such, it should be self-contained: that is, someone who has never seen the instructions should be able to understand the problem that you are solving and how you are solving it. This means that all of your choices should be thoroughly explained and motivated. Your final linear state-space model must capture the essential behavior of the dynamic system. However, the most important thing is not that you succeeded in producing the absolute best models, but that you can explain the shortcomings and merits of your model. These shortcomings and merits can come from different aspects and can be of a different nature, such as total fit, fit from input to output, noise model or not, physical interpretation. Therefore, be as clear as possible in your explanations. Also include relevant plots of the phenomena that characterize your choices (and explain what you can see in them). Which final model you recommend and why?
Below is a list of sections that should be included in your report:

1. Short description of the process (input, measured outputs, etc).
2. Description of the experiment design (type of input, possible constraints, etc).
3. Description of the experimental dataset.
4. Data pre-processing (informative data selection, discard of the section with initial transient response or/and nonlinearity effects, etc).
5. Model order selection (method used and results).
6. Parameter estimation (methods used and results).
7. Model validation including physical interpretation.
8. Discuss the strengths/weaknesses of the identified model(s) and specify which model do you recommend and why.
9. LQR setting and results in simulation and implementation.

Your report must be submitted electronically in a .pdf format. The deadline for the submission of your lab report will be specified in class with your instructor.