State-space control of a mobile robot

1 Modeling of the robot

We consider a mobile robot schematized in Figure 1. We assume that it moves on a horizontal surface due to a driving force (also horizontal), denoted $\overrightarrow{F}$. The movement is studied along a unique axis (along a straight line). The set of forces slowing down the robot are seen as the force denoted $\overrightarrow{F_c}$ whose intensity is equal to α (a positive constant transcribing the frictions) multiplied by the speed of the robot.

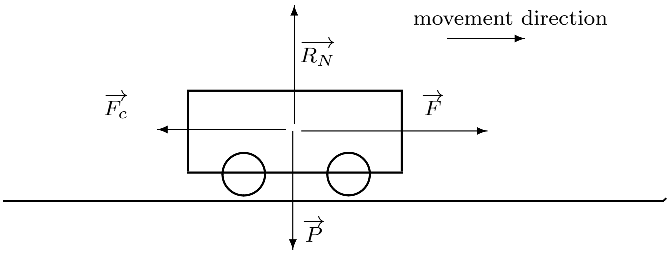

Figure 1 – Representation of the mobile robot

We act on the voltage applied to the electric motor (rotating the wheels) to make the driving force varying. We consider that the position and the speed of the robot are measured. We then consider a dynamic model of the robot with :

— the intensity of the driving force $\overrightarrow{F}$ as input, denoted v ;

— the position of the robot as output, denoted y.

We choose as state variables : $y(t)$ the position of the robot and $\dot{y}(t)$ its speed, that is

$$x(t) = \begin{pmatrix} y(t) \\ \dot{y}(t) \end{pmatrix}.$$

1. While neglecting the moment of inertia of the wheels and denoting m the robot mass, find a state-space representation for the robot.

In the following, we aim at controlling the position of the robot. It is then natural to consider as input a target position u. The control applied to the system is then still the driving force v having for value $u - y$ (see Figure 2).

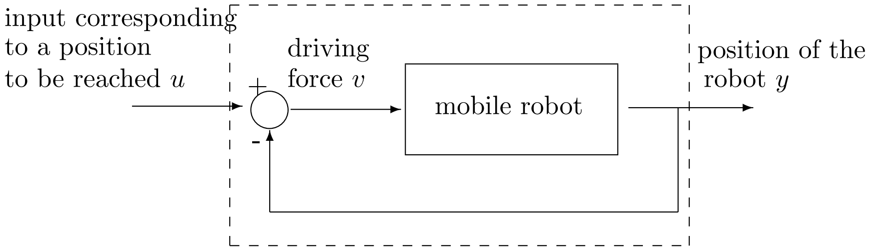

FIGURE 2 – System with input u and output y

2. Find the state-space representation with input u (instead of v) and still y as output.

3. Prove that the system is stable ?

4. Consider a second-order system described by the following transfer function :

$$H(p) = \frac{Y(p)}{U(p)} = \frac{k}{1 + \frac{2\xi}{\omega_n}p + \frac{1}{\omega_n^2}p^2}.$$

Verify that if we choose as state-vector $x(t) = \begin{pmatrix} y(t) \\ \dot{y}(t) \end{pmatrix}$, then we get for state-space-representation matrices

$$A = \begin{pmatrix} 0 & 1 \\ -\omega_n^2 & -2\xi\omega_n \end{pmatrix}, B = \begin{pmatrix} 0 \\ k\omega_n^2 \end{pmatrix} \text{ and } C = \begin{pmatrix} 1 & 0 \end{pmatrix}.$$

5. Comparing the state-space representation obtained at Question 2 and the state-space representation of a second-order system detailed at Question 4, it is obvious that the robot can be studied as a second-order system. Deduce the expressions for the damping ratio ξ, the static gain k and the natural frequency ω_n.

6. A unit step response of the robot is depicted in Figure 3. By means of this curve, identify the static gain, the damping ratio and the natural frequency of the system. Deduce the value of m and α.

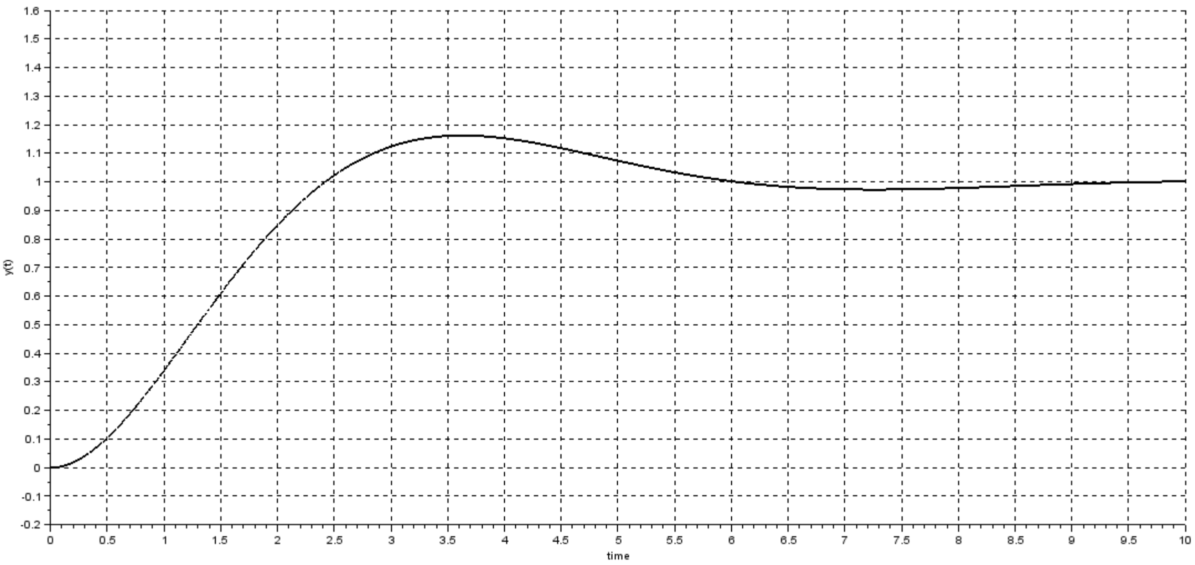

FIGURE 3 – Step response of the robot.

7. Deduce the settling time to within 5% ($t_{r5\%}$) for the robot.

8. What will happen if we apply a step whose range is 2 (to move the robot 2 units of distance farther) and if there is an obstacle 2.2 units of distance farther ?

9. In the following, we use Xcos application with Scilab (equivalent to Simulink with Matlab) to do the experiments. To get one's hand in, construct the Xcos model depicted in Figure 4 to simulate the step response of the robot. Plug-in the following information :
   - Enter the corresponding A, B, C, D matrices in the State-Space box (the column separator is ',' and the row separator is ';').
   - Set the Clock box with the following parameters : Period=0.01 ; Initialization time=0.
   - Configure the simulation time (Simulation>Configure) to 10.

   Run the simulation and check that you obtain a similar step response to Figure 3.

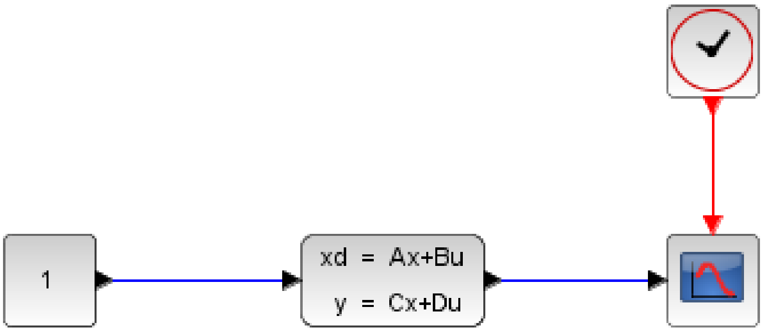

FIGURE 4 – Xcos model to simulate the step response of the robot.

2 State feedback

As depicted in Figure 5, we want to find a state-feedback matrix L and a gain K_{re} in order to impose some desired poles and a chosen static gain for the robot.

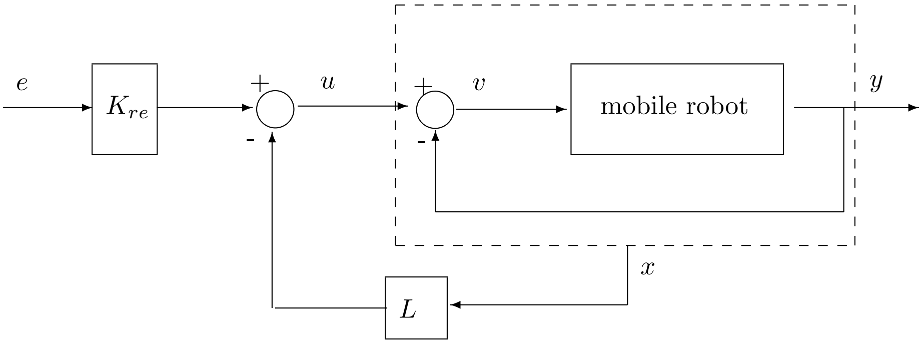

FIGURE 5 – Principle of the state-feedback

Let us denote ξ_d and ω_{nd} the damping ratio and the natural frequency desired for the robot once the state-feedback is applied.

10. Check that the system is controllable. Find the state-feedback matrix L as a function of ξ, ω_n, k (system parameters) and ξ_d, ω_{nd} (desired parameters for the system once the feedback is applied).

11. Express the gain K_{re} as a function of ξ, ω_n, k, ξ_d and ω_{nd} to be added upstream the feedback to obtain a unit static gain for the controlled system.

We want to find a control law such that the robot reaches the target position (specified by reference input e) as fast as possible without overshoot (to avoid collisions with the possible obstacles behind the target position).

More precisely, we first want to find regulation matrix L (see Figure 5) such that the system behaves as a second-order system with 1 as damping ratio and 2 as natural frequency. As an indication, with $\xi = 1$ the settling time to within 5% ($t_{r5\%}$) can be evaluated by :

$$t_{r5\%} \approx \frac{4.8}{\omega_n}.$$

12. Explain why these choices (for the damping ratio and the natural frequency) make it possible to fullfill the control objectives.

13. Give the desired characteristic polynomial.

14. Find the regulation matrix L of the state-feedback.

15. Find the gain K_{re} to be added upstream the feedback to obtain a unit static gain for the controlled system.

16. Use Xcos to experiment your state-feedback control. Construct the Xcos model depicted in Figure 6 to simulate the step response of the robot with the state-feedback control. Plug-in the following information :

- Enter the corresponding A, B, C, D matrices in the State-Space box. Note that we are interested in all the states at the output of this, so matrix C is different from the one in the model
- Variables L and K_{re} declared in the gain boxes have been defined in the Scilab console.
- Set the Clock box with the following parameters : Period=0.01 ; Initialization time=0.
- Configure the simulation time (Simulation>Configure) to 10.

Run the simulation and check that you obtain a step response fullfilling the control objectives.

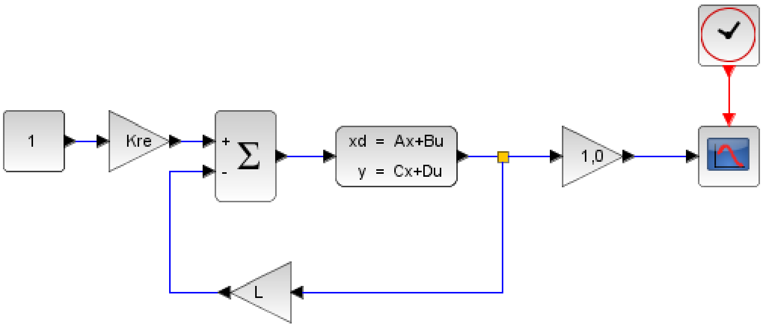

FIGURE 6 – Xcos model to simulate the step response of the robot with the state-feedback.

17. Use again the Xcos models to simulate a step response of the robot without and with the state feedback control. Modify these Xcos models to visualize control input v applied to the robot. Give an interpretation of why the ranges of signals v are different.

18. Consider now that signal v must take its values between -5 and +5 (corresponding to the range of voltages which can be applied to the electric motor used in the robot). Using the Xcos model for the robot with the state-feedback control, and by a "generate and test" approach, tune matrix L and gain Kre such that $t_{r5\%}$ is the smallest possible (still with $\xi_d = 1$) for a unit step as reference input e while avoiding control input v to exceed the defined bounds.

3 Discrete-time representation

Up to here, the control of the robot has been studied in the continuous-time domain. The application of state-feedback L and gain K_{re} could then be done by means of electronic circuits amplifying the measures of the position and the speed, as well as the reference input.

In the following, we will consider that the control should be done by means of a computer (typically embedded in the robot). This leads us to study the same system but in the discrete-time domain.

19. Considering that we will design a control law such that the robot will behave as a second-order system with the natural frequency found at question 18, choose a proper sampling period Δ.

20. As a reminder, the discrete-time state-space represenation $(A_{ech}, B_{ech}, C_{ech})$ of a continuous-time system represented by (A, B, C) and sampled at period Δ is :

$$A_{ech} = e^{A\Delta} \quad , \quad B_{ech} = \int_0^\Delta e^{A(\Delta-\nu)} d\nu B, C_{ech} = C.$$

Find matrices $(A_{ech}, B_{ech}, C_{ech})$ using Scilab function `dscr()`.

21. Verify the controllability of the system. The scilab function `rank` can be used for this purpose.

22. We want to find a regulation matrix L such that the system behaves as a second-order system with natural frequency $\omega_{nd} = 2$ and damping ratio $\xi_d = 0.9$. In discrete-time domain, this leads to consider the following desired characteristic polynomial :

$$P_{des}(z) = (z - z_1)(z - z_2)$$

with

$$\begin{aligned} z_1 &= e^{\left(-\xi_d \omega_{nd} + j\omega_{nd}\sqrt{1-\xi_d^2}\right)\Delta} \\ z_2 &= e^{\left(-\xi_d \omega_{nd} - j\omega_{nd}\sqrt{1-\xi_d^2}\right)\Delta}. \end{aligned}$$

Use `ppol` Scilab function to find the regulation matrix L. Note that function `complex` can be used to declare complex numbers.

23. Find the gain K_{re} to be added upstream the feedback to obtain a unit static gain for the controlled system. Function `inv` can be used to invert matrices.

3.1 Observer design

Let us now assume that only the position of the robot can be measured. To be able to apply the state-space feedback, we have to estimate the speed of the robot. For this purpose, we use an observer and the final control structure is depicted in Figure 7.

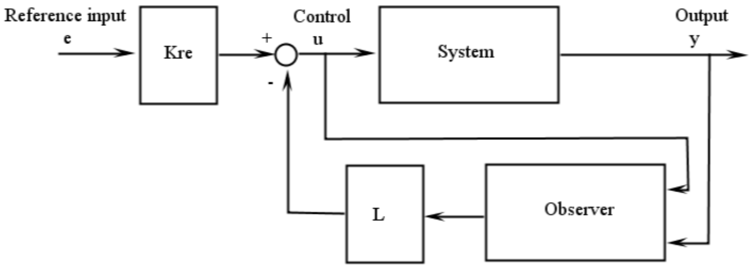

FIGURE 7 – Control structure with an observer

24. Verify that the system is observable using function `rank()`.

25. We want to impose the poles of the observer to be equal to zero (at the origin of the complex plane). Find matrix L_{obs} accordingly using function `ppol()`.

26. Use Xcos to experiment your state-feedback control with an observer.

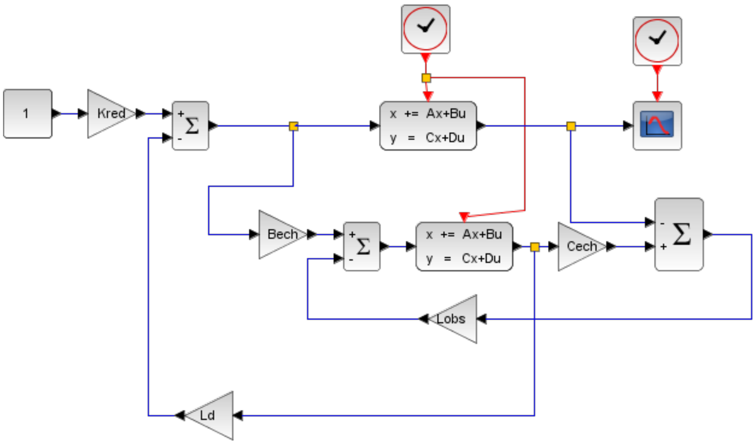

FIGURE 8 – Xcos model to simulate the step response of the robot with the state-feedback and the observer.