Intro to ACA Lab project, 2018.

Jens C. Therkildsen, Jonathan Brandtjensen
Aarhus University
IFA

Contents

1. Introduction

For our project, we decided to attempt to implement a automatic parking corrector, specifically for packing between objects. The main feature which has been implemented, is that when an object is detected within a vicinity κ, but larger than a vicinity ℓ_{min}, of either end of a small car, the car will automatically move itself towards the object and stop at a distance ℓ_{min} from it. This is done through a ultrasonic distance sensor on both ends of the car. This creates the basis for expanding the project to include two objects, and making the car find the midpoint, as the routine employed is the same, the difference being delay and system effects introduced by having more inputs going into the system at the same time.

We keep the system to first order, not considering integrators within implemented control schemes, as the only differentiated quantity is the speed, v_0, which is constant most of the time, leaving us only to consider the relative distances, even allowing neglection of the length of the car for now.

2. Overview of Scenarios

The left-right symmetry of the problem eliminates half of our cases instantly. The largest disctinction between all cases is whether or not one or two objects are involved. However only the case involving a single object will be discussed in the report.

2.1. One Object

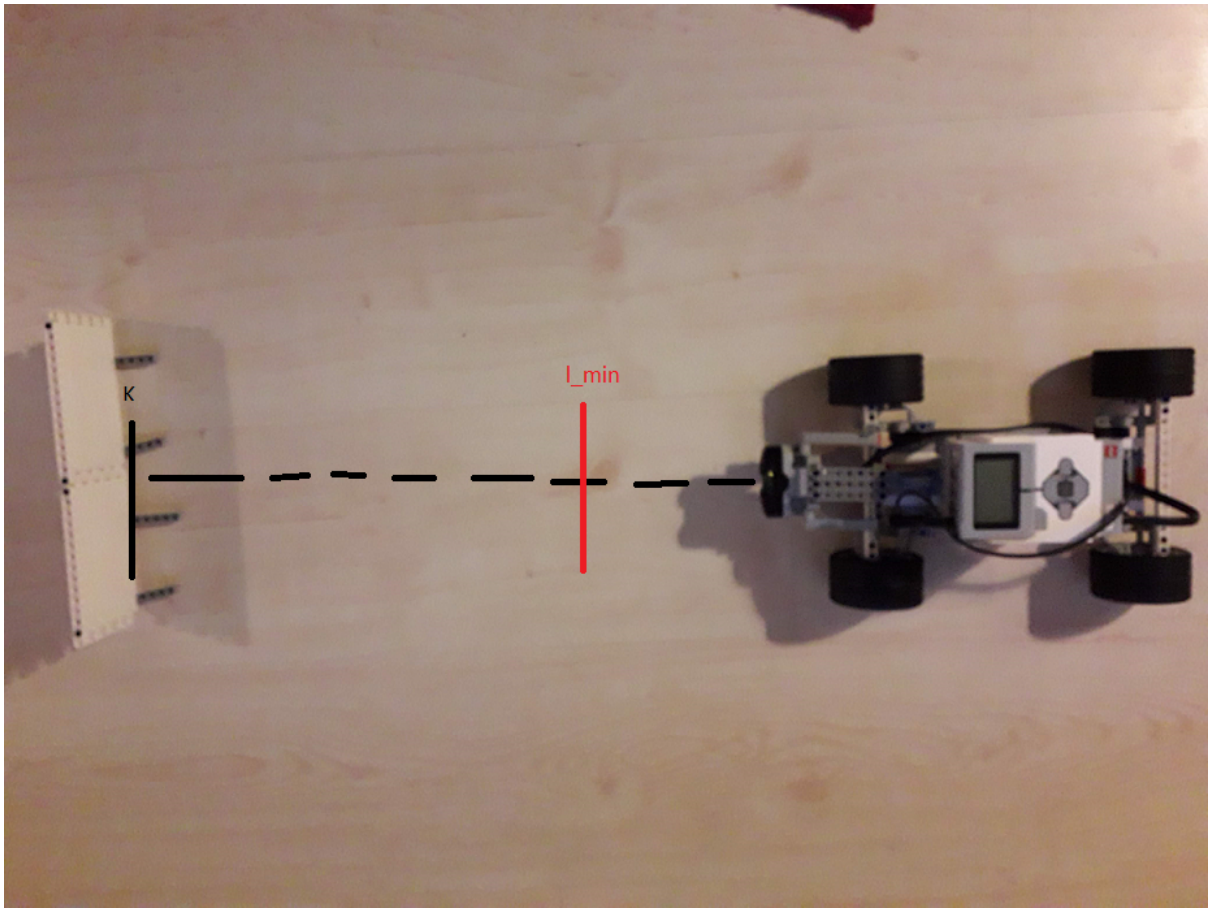

Figure 1: *Picture showing car detecting object K, before needed to move until K is within distance* ℓ_{min}

In the primary use case, a single object is detected with the distance K from the sensor. After comparing whether K is within the ℓ_{min} distance, the car will move either forward or backwards to maintain a distance of ℓ_{min} to the object. In the case of 1 the would move forward.

3. Hardware Considerations and Physical Set Up

3.1. Sensor

For the project, we had two sensors to choose between: The EV3 Infra-red and the EV3 Ultrasound sensor. We ended up selecting the ultrasound sensor following a the measurements and considerations mentioned hereafter.
For our purposes, we had some straightforward of requirements for our sensor:

1. We require the sensor to be able to consistently provide measurements to some accuracy (this accuracy in turn determining ℓ_{min}) and for the accuracy, and the measurements, to be stable.

2. The sensor readings must be unmolested by the physical surroundings, as might be the case with objects on the perimeter of sensors field of view.

3. The time delay between the movements of an object whose proximity is being recorded, and the sensor outputting this new position, must be fairly small. This is important, as it imposes a constraint on the movement speed of our car, v_0.

Having carried out a series of tests on each of the sensors we ended up settling on the EV3 Ultrasound sensor, as this was the one best fulfilling the needs outlined in the points above. Based on these tests

were able to determine values for the parameters mentioned in the above points. The parameters set is provided in the section on modelling.

3.2. DC Motor

In this project a Lego EV3 DC Motor is used. Considerations, such as those described in the subsection on sensors, were not needed here as there were not really any other obvious options on the table.

The EV3 DC motor works as any other DC motor, and is therefore in principal already self-regulating by virtue of the induced e.m.f being proportional to the motor speed. However, this regulation would primarily be in response to a change in load (such as the chassis and wheels of the car).

4. The system and it's transfer function

4.1. Modelling, Parameters and Maths

A primary ingredient in the modelling of this system is to determine a transfer function for the DC engine which provides a suitable balance between described complexity and accuracy.

Below is a simplified schematic diagram depicting the inside of the DC motor:

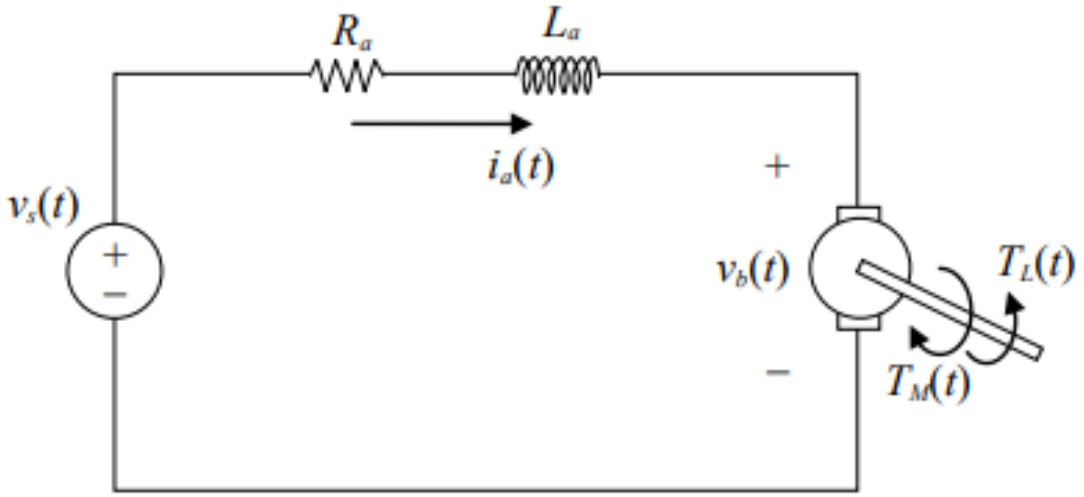

Figure 2: *Schematic drawing of a DC engine with a payload (in our case, the chassis of the LEGO car. The quantity $v_s(t)$ is the input voltage, $i_a(t)$ is armature current, $v_b(t)$ is the induced back e.m.f, R_a is the armature resistance and L_a is the inductance of the motor coil. Finally, $T_M(t)$ is the torque generated by the motor and T_L is the torque of the load counteracting the torque generated by the motor, as well as friction*

The first of the differential equations describing the system is obtained through use of Kirchhoff's second voltage law and the fact that the back e.m.f, $v_b(t)$, is directly proportional to the angular velocity of the rotor (the shaft) in the motor, i.e., $v_b(t) = k_b * \omega(t)$, where the proportionality constant k_b is the "back e.m.f" constant.

The next of the two differential equations describing the system is obtained by considering Newton's laws in terms of moments of inertia and torque (as seen in figure 2). Further simplification of the resulting equations can be done by considering the system in it's steady state.

In steady state we have that $v_b(t) = v_s(t)$, and therefore $\omega(t) = \omega$, i.e., constant. In steady state the same is also true of the current $i_a(t)$, the motor torque T_M and the input voltage v_s. Considering the power input in the system, and that it must balance with the sum of the output and what is dissipated in the resistor, and keeping in mind that the motor torque T_M is directly proportional to the current i_a, then it can be seen that the proportionality constant between torque and current is identical to the previously mentioned proportionality back e.m.f constant k_b.

These considerations then produce the two differential equations describing the system.

The mechanical one:

$$J_M \frac{d\omega(t)}{dt} + B_M \omega(t) = T_M(t) - T_L(t) \tag{1}$$

The electrical one:

$$L_a \frac{di_a(t)}{dt} + R_a i_a + k\omega(t) = v_s(t) \tag{2}$$

Recalling that $T_L(t)$ is the load on the vehicle (e.g., it's mass and the friction between wheels and road), it can be considered approximately constant, so we can for all purposes ignore it and consider it as falling under $T_M(t)$. Using this,and that one can write $T_M(t) = ki_a(t)$, the mechanical equation becomes:

$$J_M \frac{di_a(t)}{dt} + B_M \omega(t) = ki_a(t) \tag{3}$$

Applying the Laplace transform, then renders the following mechanical equation:

$$s\left(sJ_M + B_M\right)\Theta(s) = kI_a(s) \tag{4}$$

With the following electrical equation:

$$\left(L_a s + R_a\right) I_a(s) = V_s(s) - ks\Theta(s) \tag{5}$$

In the case with which we are concerned we are considering distances rather than velocities, both of these being directly related to the angle and angular velocities by the radius. The output of our DC motor is a measurement of angular velocity this means it is necessary to integrate in any case, and therefore define the angle as output, $\Theta(s) = Y(s)$, and the input voltage as input $V(s) = U(s)$.

In what follows subscripts are no longer needed and therefore omitted. Since

$$I(s) = \frac{s\left(sJ + B\right)\Theta(s)}{k} = \frac{V(s) - ks\Theta(s)}{Ls + R}, \tag{6}$$

then some rearrangement gives the following transfer function

$$H(s) = \frac{Y(s)}{U(s)} = \frac{\Theta(s)}{V(s)} = \frac{k}{s(sJ + B)(Ls + R) + k^2 s} = \frac{k}{s\left((Js + B)(Ls + R) + k^2\right)}. \tag{7}$$

This transfer function represents, as it stands above, a second order system. From some internet research, one can see that in DC engines, one will commonly have $\mathcal{O}(R) = 10^{-1}$,$\mathcal{O}(L) = 10^{-3}$, $\mathcal{O}(J) = 10^{-4}$,$\mathcal{O}(k) = 10^{-2}$ and $\mathcal{O}(B) = 10^{-4}$ in S.I units. The large differences in $\mathcal{O}(R)$ and $\mathcal{O}(L)$ then mean that the pole (s) in the term $(Ls + R)$ must be at least an order or two larger in magnitude than the other from the $(Js + B)$ term. Therefore it can be ignored, as s is exponentiated, and the exponential will therefore die out far more quickly for this pole, making it's overall effects on transient behaviour less significant.

In some regards this is not surprising, as the neglected pole represents an electronic term and electronic systems operate on significantly smaller time scales than electronic ones. However, because of this the transfer function of the system then becomes

$$H(s) = \frac{k}{s\left(Js + B + k^2\right)} = \frac{K}{s\left(s\tau + 1\right)}, \tag{8}$$

with the last equation arising by redefining the constants in the first equality to $J = \frac{k\tau}{K}$ and $B = k(\frac{1}{K} - k)$.Although the former is closer to the pole placement that will occur in the next subsection, the latter is more manageable.

4.2. Pole Placement

To make the transient behaviour of the system as we desire, we can place its poles. It is common practice to treat all systems of second order or higher as a second order system, with the desired system properties being related to a second order transfer function as below

$$Q(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n^2 s + \omega_n^2} \quad . \tag{9}$$

Here the damping ration $\zeta = \frac{-ln(\frac{O.S_{\%}}{100})}{\sqrt{\pi^2 + ln^2(\frac{O.S_{\%}}{100})}}$, where $O.S_{\%}$ is the desired overshoot in percent. Also the settling time $T_s = \frac{4}{\zeta\omega_n}$, where T_s is the settling time in seconds.

For our system, we deemed that a settling time of $T_s = 0.5s$ and $O.S_{\%} = 10$ to be decent values. From this the standard form second order transfer function we then determined the position of the desired open loop poles, by solving for zero in the denominator of this form, to be

$$s_{\text{Desireable}} = -8.00007 \pm 10.9151i \tag{10}$$

4.3. State Space Model

Having previously disclosed the differential equations and transfer function of the system at hand, it is fairly straight forward to represent the system in state-space. Using the standard forms

$$\dot{x} = Ax + Bu, \tag{11}$$

and

$$y = Cx + Du \ , \tag{12}$$

one of the system's state space representations can be found quite readily. Since the system's transfer function $H(s)$ is second order, the derivative on the left hand side of the first control equation will cause the resulting system matrices to be 3x3's, however this extra dimension contains no non-zero elements and can therefore be ignored. The state space representation, in phase variable form,then becomes

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{13}$$

and

$$y = \begin{bmatrix} \frac{K}{\tau} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \tag{14}$$

4.4. Observer and Controller Design

In our system we have one entities performing measurements. A ultrasound sensor, providing running measurements of the relative distance between itself and the obstacle.

Based on the idea that the distance measured by the sensor, is technically a position, we have tried to make a observer with controller design to regulate the change in position until we reach the reference ℓ_{min}.

5. Implementations and Results

First we implemented a simulation of the observer and controller

Based on a transfer function made from data from the motor. We found the following transfer function.

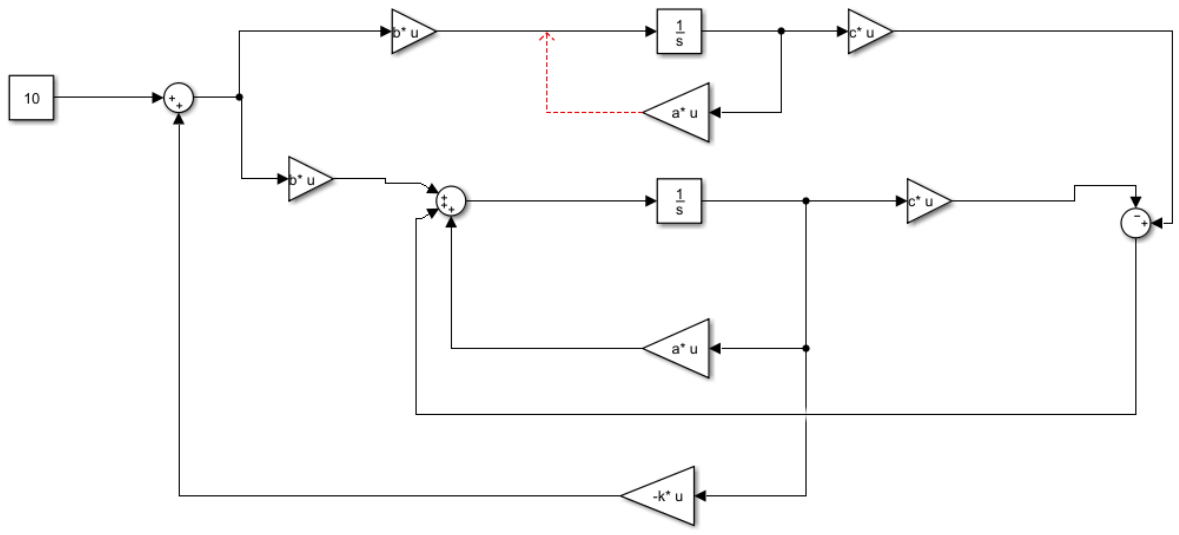

Figure 3: *Picture showing simulation of distance measurement observer and controller*

```
tf =

9667/(100000*(s - 6523/1000))
```

Figure 4: *transfer function of the motor made by matlab function tfest*

Using this transfer function to create a state space model we filled in the variables in the simulation, and tried to create a simulink to send to the lego ev3.

Unfortunately this system was very unstable and oscillated from positive to negative values, until surpassing the limits of the lego ev3 hardware and stopping.

We assume this could be because of a steady state error introduced during the design.

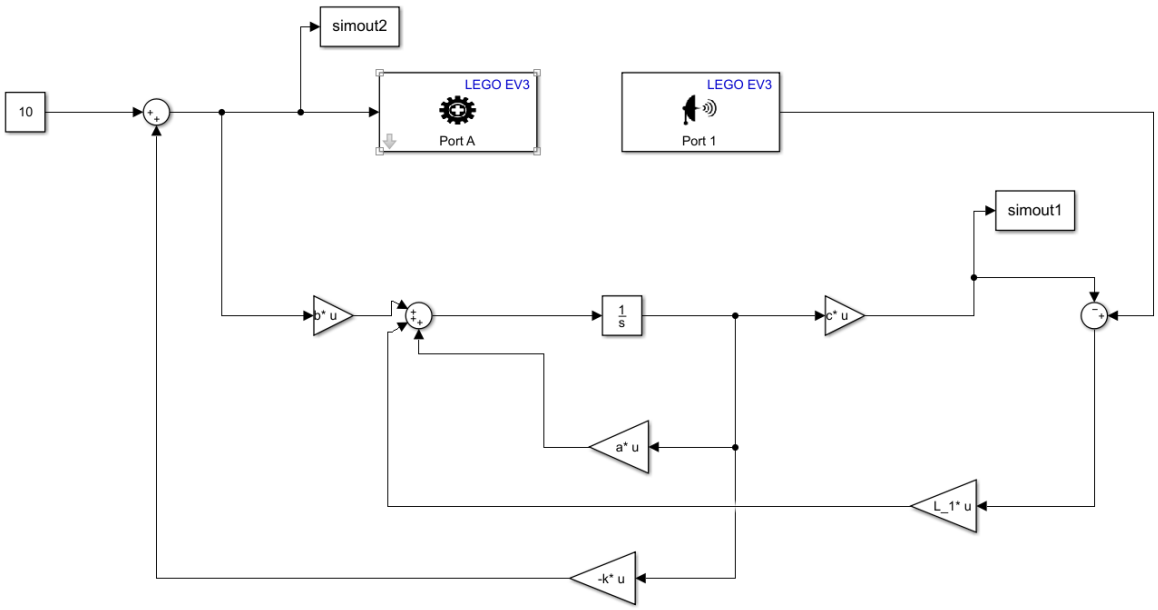

Figure 5: *Picture showing a lego ev3 version of distance measurement observer and controller*