UNIVERSITY OF TRENTO - Italy
Information Engineering
and Computer Science Department

Master Degree in Computer Science

Applied Robotics

AA 2015-2016

Controller design for Lego Mindstorm motor

Diego Verona, Aliaksandr Siarohin, Mattia Digilio

December 15, 2015

Abstract

Report for the second assignment on Applied robotics: design and implement controller for the Lego NXT motor.

In this report we show our controller, describe it properties and describe it digital implementation.

1 General definition

Theorem 1 ***Root locus.*** *The* `root locus`*, or* `Evans locus`*, is a graphical method that depicts the curves of the roots of the denominator of the closed loop transfer function in the complex plane (sometimes called Argand plane or Gauss plane).*
The curves are parameterized by a parameter, typically the gain of the loop [1]

2 Design of continues time controller

2.1 Controller requirments

The contoller should has:

- stady state tracking error = 0
- overshot $< 20\%$
- settling time < 0.4s

Overshot requirement on root locus plot is shown by the following formula:

$$\frac{Re}{Im} = \frac{\xi}{\sqrt{1-\xi^2}} = \pm\frac{ln0.2}{\pi} \tag{2}$$

To show settling time requirement, it is possible to use the dominant pool approximation:

$$Re = \frac{ln(\alpha)}{0.4} \tag{3}$$

2.2 Our design

$$C(s) = \frac{(s+10)^2}{s(s+21)} \tag{4}$$

$$K_c = 10 \tag{5}$$

Root locus can be seen in fig. 1, and the ideal response to 1(t) in fig. 2.
Results of Scicoslab simulation is shown in different figures:

- Ω in fig. 3
- Power in fig. 4

[1] `http://disi.unitn.it/~palopoli/courses/ECL/RootLocus.pdf`

- Tracking error in fig. 5

Code is available in a shared folder[2].

3 Practical case: vehicle

Using 2 different motors applied on the NXT brick, it is possible to do a simple modelling of a vehicle. The scope of simple experiment is to let our model to go straight without any visual or position control, but only using the the following implementation of a digital controller.
Digital version of controller is (obtained using trapezoid rule):

$$y_{k+2} = \frac{1}{4+42*T}(K_c u_{k+2}(4+100T^2+40T) + K_c u_{k+1}(-8+200T^2) \\ + K_c u_k(4-40T+100T^2) + 8y_{k+1} - y_k(4-42*T)) \quad (6)$$

Speed estimated using exponential average:

$$S(t) = 0.075 * S(t) + (1-0.075) * \frac{(Angle(t) - Angle(t-1))}{T} \quad (7)$$

Code is available in a shared folder[3].

4 Conclusion

[2] `https://github.com/AliaksandrSiarohin/AppliedRobotics/tree/master/controler`

[3] `https://github.com/AliaksandrSiarohin/AppliedRobotics/tree/master/motor_controller`

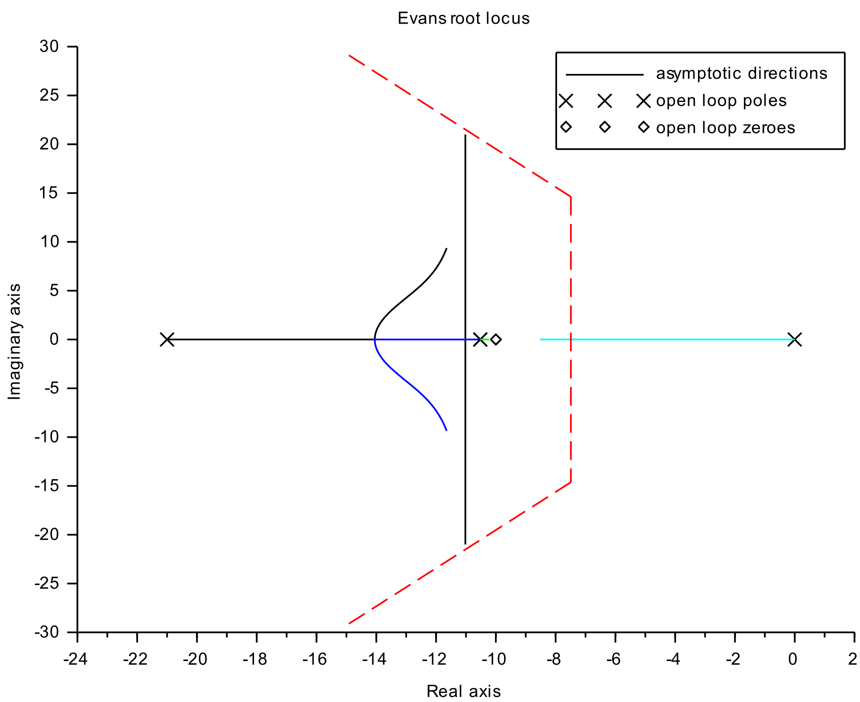

Figure 1: Root locus, red lines show constains on overshot and settling time

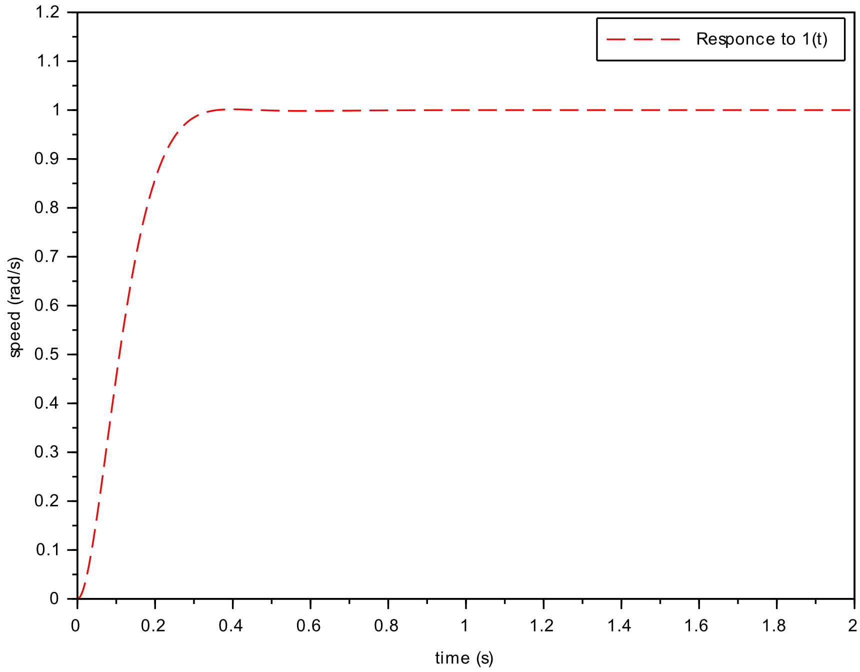

Figure 2: response to 1(t).

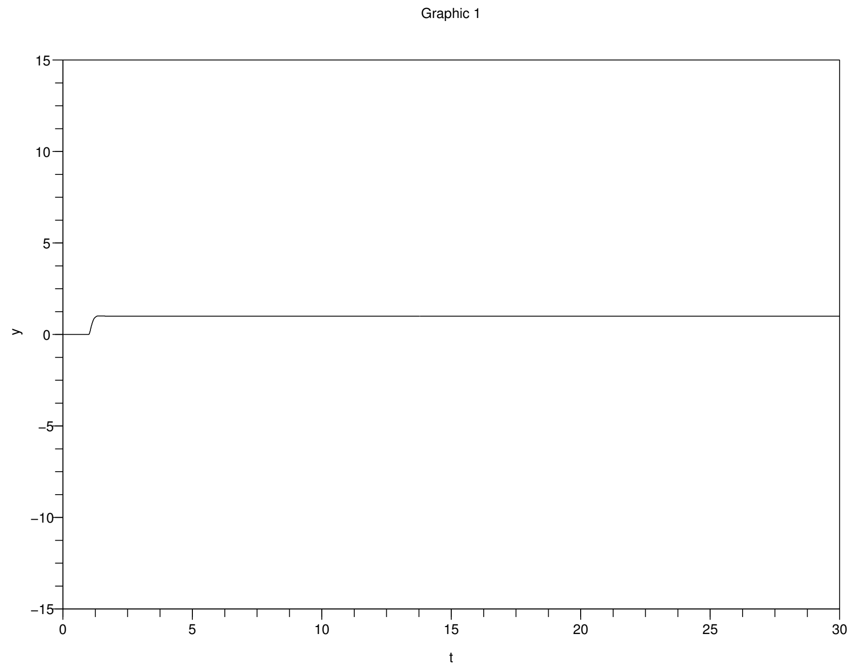

Figure 3: Scicoslab simulation.

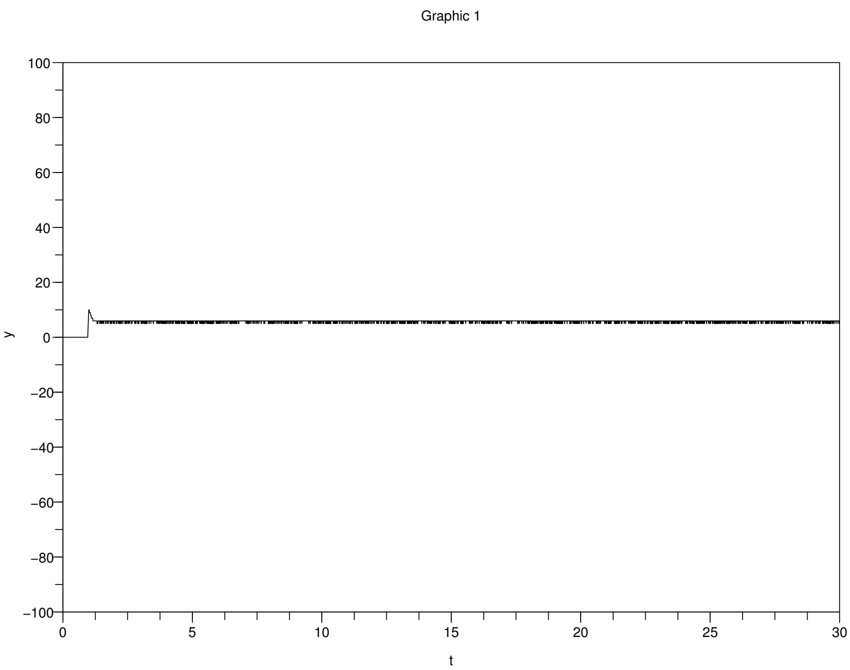

Figure 4: Scicoslab simulation.

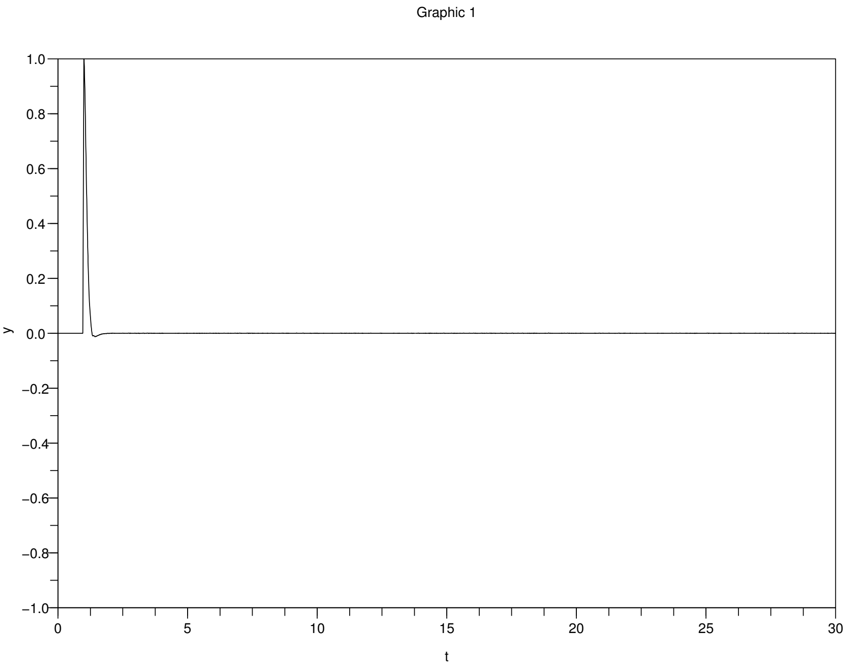

Figure 5: Scicoslab simulation.