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 24, 2015

Abstract

Report for the second assignment on Applied robotics: design and implement a 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]

Theorem 2 ***Closed-loop transfer function***. *A closed-loop transfer function in control theory is a mathematical expression (algorithm) describing the net result of the effects of a closed (feedback) loop on the input signal to the circuits enclosed by the loop.* [2]

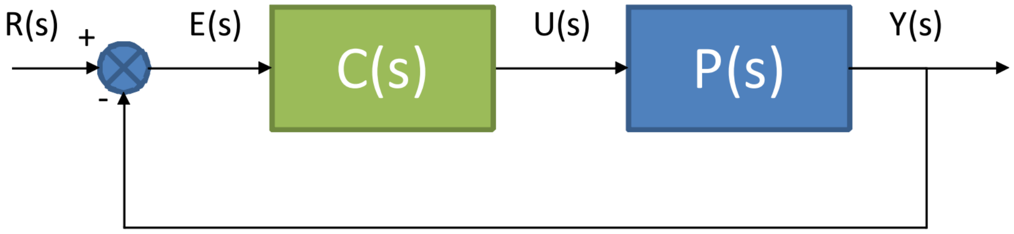

Figure 1: Closed loop with controller

2 Design of continues time controller

2.1 Controller requirements

The contoller should have:

- stady state tracking error equal to 0 we must have a pole in 0
- overshot $< 20\%$
- settling time < 0.4s

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

$$\frac{Re}{Im} = \frac{\xi}{\sqrt{1-\xi^2}} = \pm\frac{ln0.2}{\pi} \tag{3}$$

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

$$Re = \frac{ln(\alpha)}{0.4} \tag{4}$$

[1] `http://disi.unitn.it/~palopoli/courses/ECL/RootLocus.pdf`
[2] `https://en.wikipedia.org/wiki/Closed-loop_transfer_function`

1 pole in 0 to have 0 steady error, zero attract poles,(we want to attract poles to - infinity as much as we can)

2.2 Our design

controller
$$C(s) = \frac{(s+10)^2}{s(s+21)} \tag{5}$$
$$K_c = 10 \tag{6}$$

2 zero(both -10)
2 poles(0,-21)

Root locus can be seen in fig. 2, and the ideal response to 1(t) in fig. 3.
Results of Scicoslab simulation are shown in different figures:

- Speed (ω) in fig. 4
- Power in fig. 5
- Tracking error in fig. 6

Code is available in a shared folder[3].

3 Digital controller

Digital version of controller is obtained using trapezoid rule:

$$\begin{aligned} y_{k+2} = \frac{1}{4+42*T}(&K_c u_{k+2}(4+100T^2+40T) + K_c u_{k+1}(-8+200T^2) \\ &+ K_c u_k(4-40T+100T^2) + 8y_{k+1} - y_k(4-42*T)) \end{aligned} \tag{7}$$

Speed estimated using exponential average:

$$S(t) = 0.075 * S(t) + (1-0.075) * \frac{(Angle(t) - Angle(t-1))}{T} \tag{8}$$

Code is available in a shared folder[4].

4 Conclusion

In this assignment we design motor controller and implement it digital version on Lego NXT motor.

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

[4] https://github.com/AliaksandrSiarohin/AppliedRobotics/blob/81388e0b7589c9aa38e6b5bc4f148191b2f37a03/motor_controller

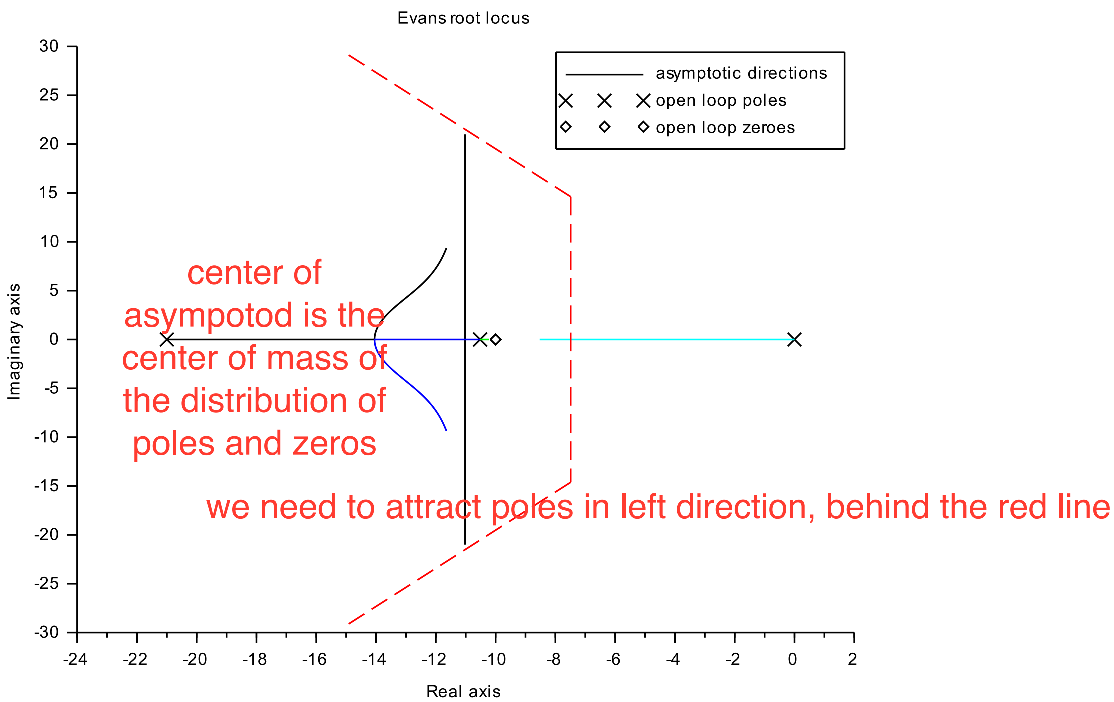

Figure 2: Root locus, red lines show constrains on overshot and settling time

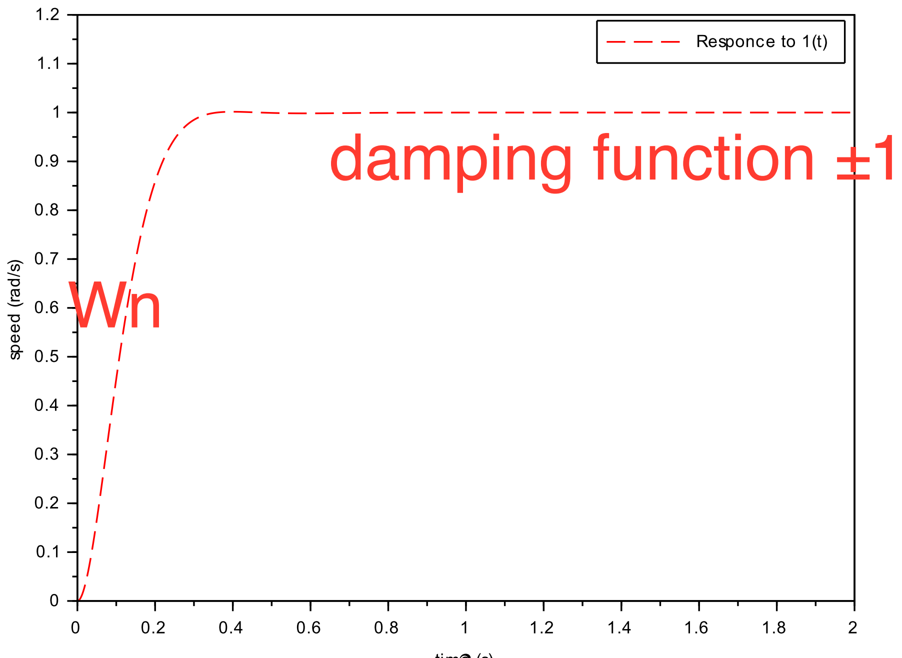

Figure 3: response to 1(t).

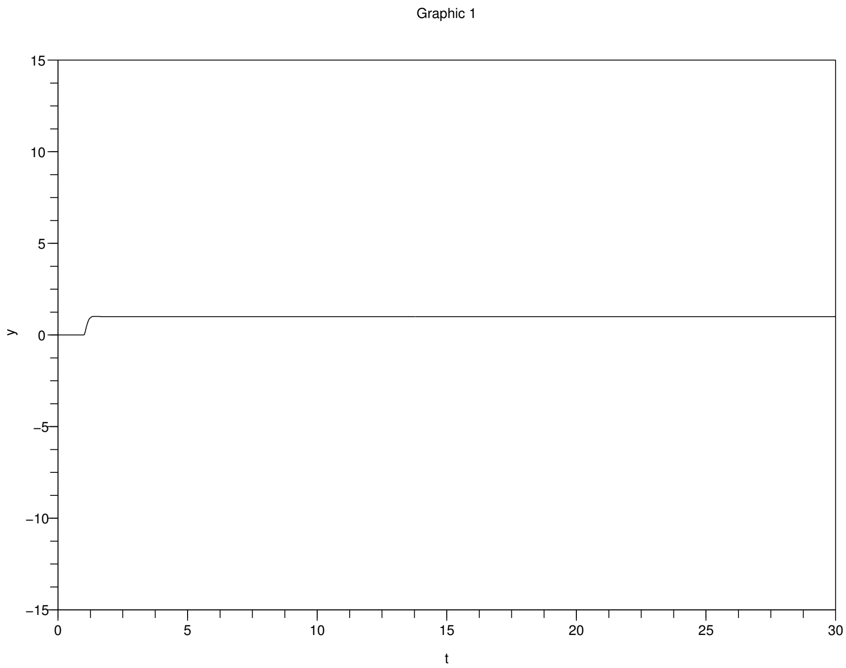

Figure 4: Scicoslab simulation: omega

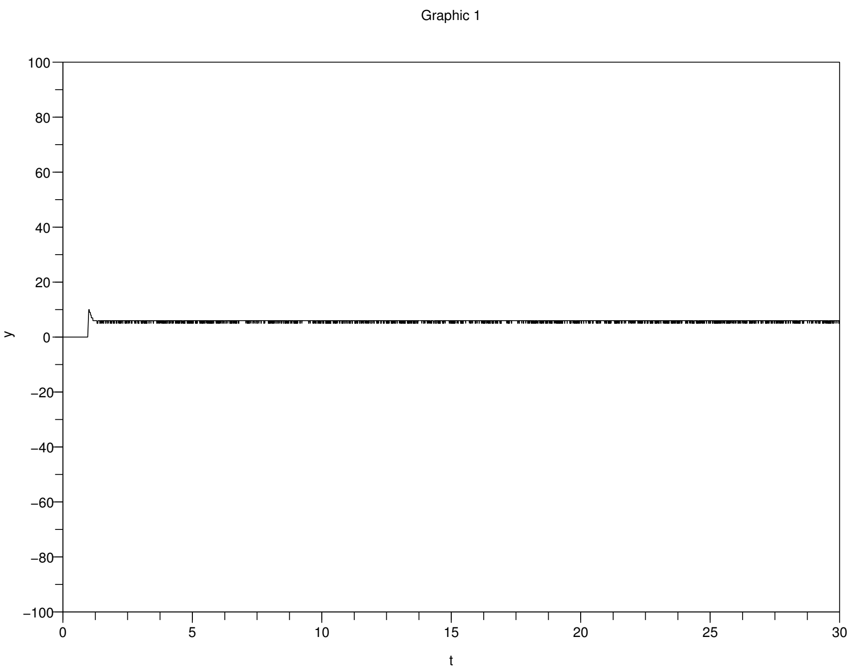

Figure 5: Scicoslab simulation: power

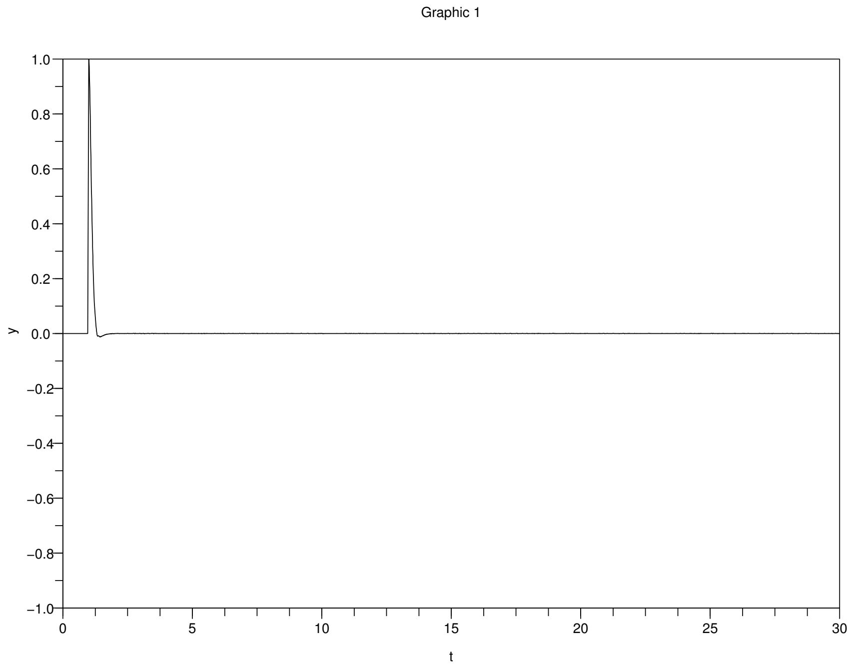

Figure 6: Scicoslab simulation: tracking error