



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 19, 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 its properties and describe its 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

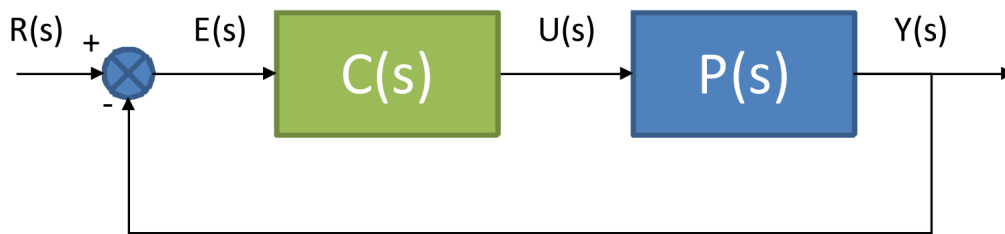


Figure 1: Closed loop with controller

2 Design of continues time controller

2.1 Controller requirments

The contoller should have:

- stady state tracking error equal to 0
- overshoot $< 20\%$
- settling time $< 0.4s$

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

$$\frac{Re}{Im} = \frac{\xi}{\sqrt{1-\xi^2}} = \pm \frac{\ln 0.2}{\pi} \quad (3)$$

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

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

¹<http://disi.unitn.it/~palopoli/courses/ECL/RootLocus.pdf>

²https://en.wikipedia.org/wiki/Closed-loop_transfer_function

2.2 Our design

$$C(s) = \frac{(s + 10)^2}{s(s + 21)} \quad (5)$$

$$K_c = 10 \quad (6)$$

Root locus can be seen in fig. 3, and the ideal response to 1(t) in fig. 4. Results of Scicoslab simulation are shown in different figures:

- Ω in fig. 5
- Power in fig. 6
- Tracking error in fig. 7

Code is available in a shared folder³.

3 Practical case: vehicle

Using two different motors applied on the NXT brick, it is possible to do a simple modelling of a vehicle. The aim of this experiment is to let our model go straight ahead autonomously without any external sensors, but only using the the following implementation of a digital controller to manage the motors.

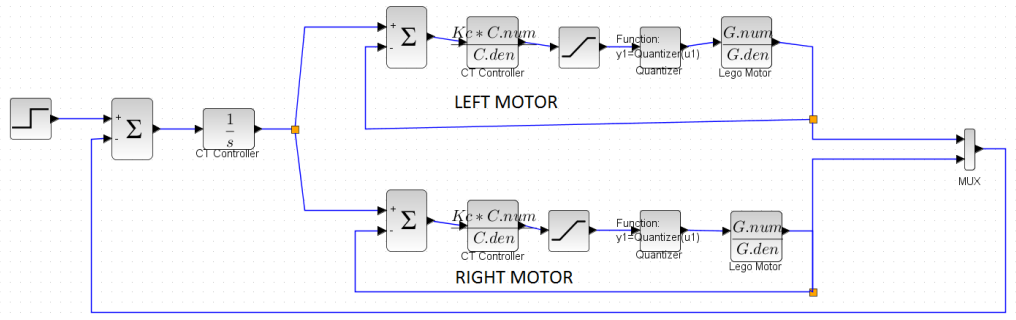


Figure 2: Closed loop with controller

Digital version of controller is obtained using trapezoid rule:

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

³<https://github.com/AliaksandrSiarohin/AppliedRobotics/tree/master/controller>

Speed estimated using exponential average:

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

Code is available in a shared folder⁴.

4 Conclusion

⁴https://github.com/AliaksandrSiarohin/AppliedRobotics/tree/master/motor_controller

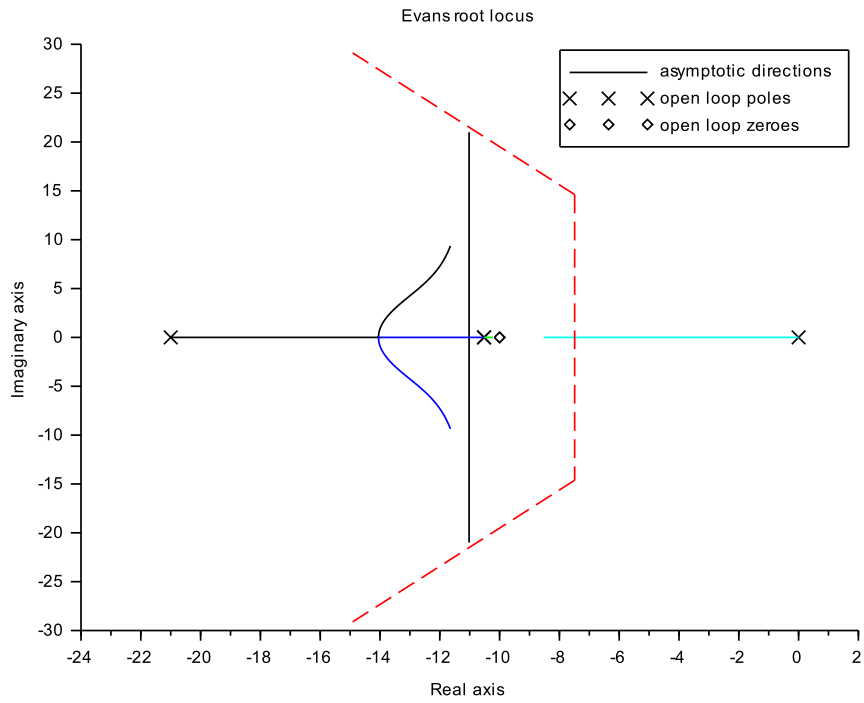


Figure 3: Root locus, red lines show constraints on overshoot and settling time

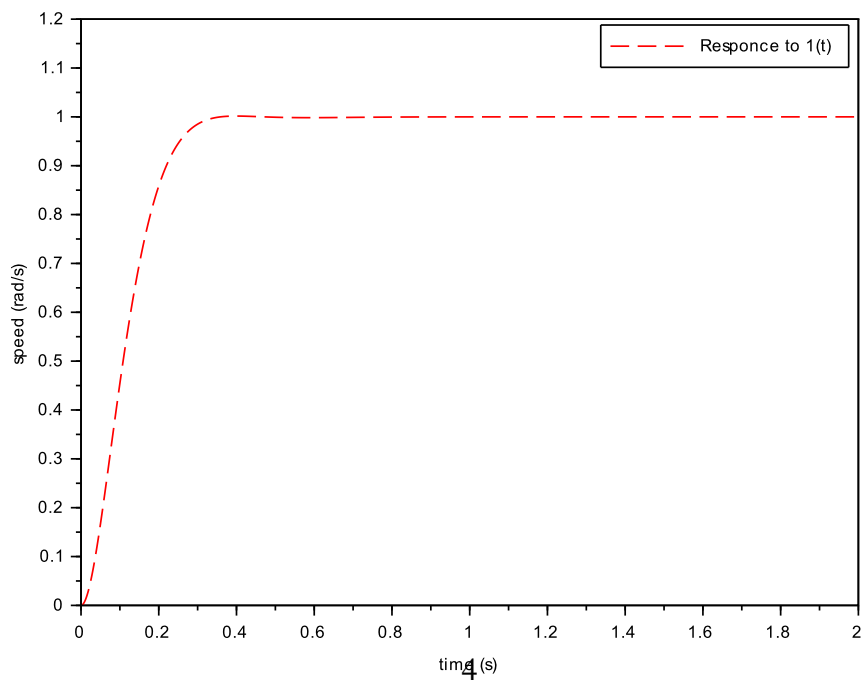


Figure 4: response to $1(t)$.

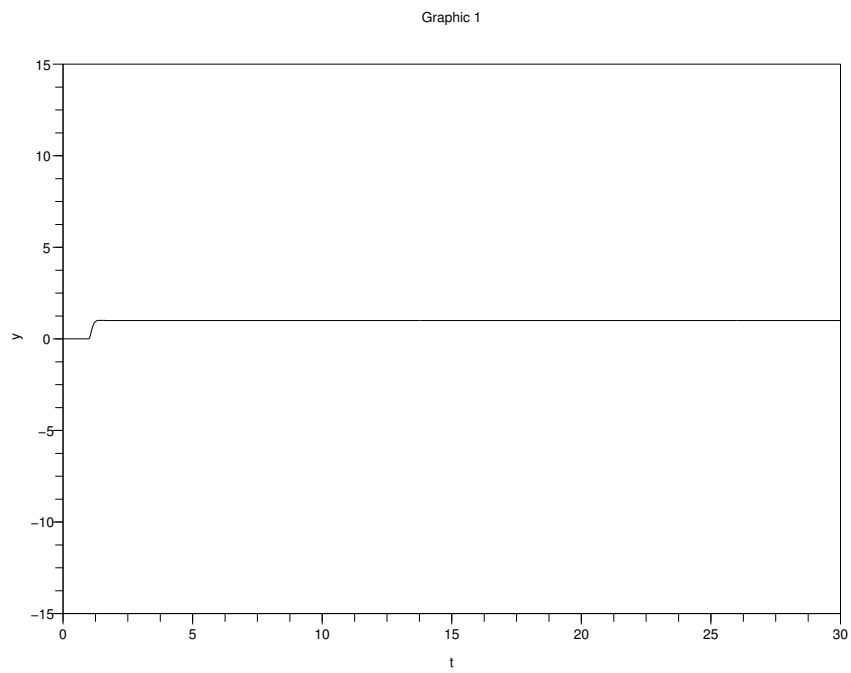
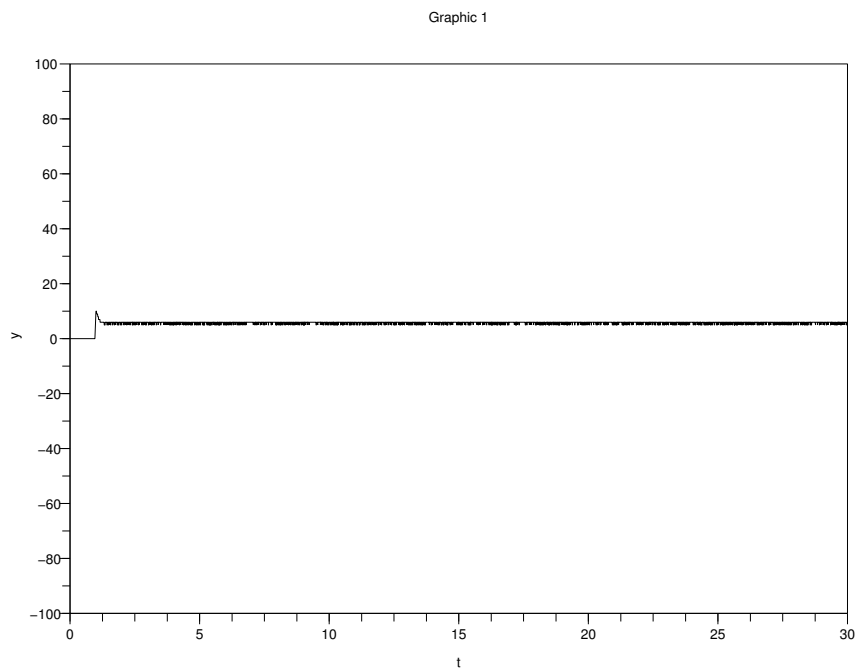


Figure 5: Scicoslab simulation: omega



5
Figure 6: Scicoslab simulation: power

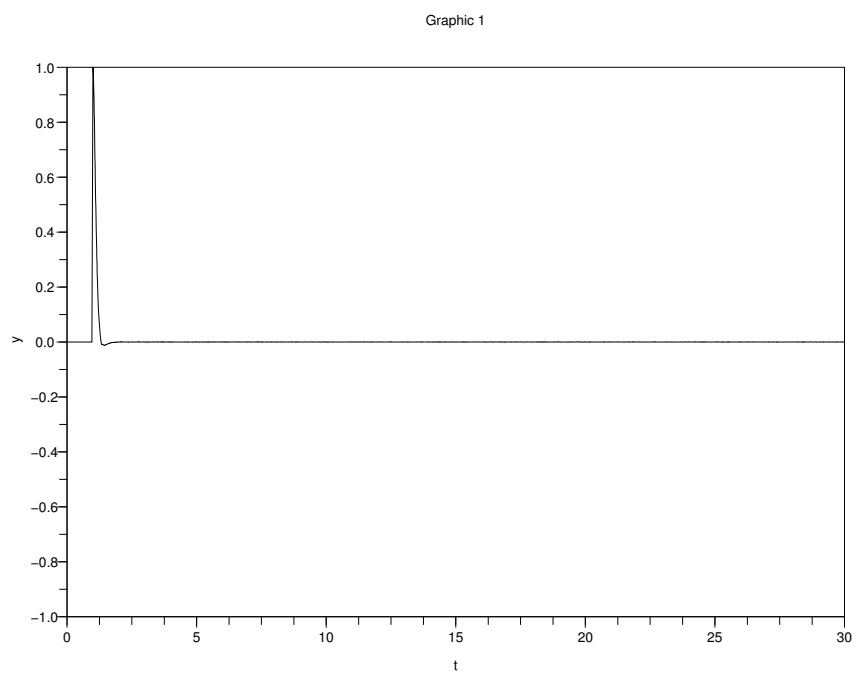


Figure 7: Scicoslab simulation: tracking error