UNIVERSITY OF TRENTO - Italy
Information Engineering
and Computer Science Department

Master Degree in Computer Science

Applied Robotics

AA 2015-2016

Identification procedure for Lego Mindstorm motor

Diego Verona, Aliaksandr Siarohin, Mattia Digilio

November 18, 2015

Abstract

Report for the first assignment on Applied robotics: getting and identification parameters of the Lego NXT motors.

In this report we discuss our method of obtaining data from Lego Mindstorm motors as well as estimation of parameters from the motor data.

1 Tools and definitions

1.1 General definition

..

1.2 Used tools

..

2 Collecting motor data

Have been used 2 different methods to collect motor data:

1. Bluetooth connection
2. Usb connection

It is important get the minimum time gap between measures, but it require an extra check to detect if the tachometer sensor is fast and precious as the used transmission. For this report is decider to use USB connection getting 10 different data files with different raw powers.

2.1 Bluetooth data collection

Given by the lab the code to comunicate between our PC and the NXT brick, we learned how it works and we implement the interface to include also the current measure NXT timestamp. The procedure to get data with bluetooth is the following:

1. Send message from PC to brick that set motor power
2. Send message from PC to brick that requests tachometer count from the motor
3. Receive message from brick with tachometer count and relative timestamps
4. Save timestamps and tachometer count to file
5. Repeat all from step 2

Using this methodology there is a very hight latency $\approx 50ms$. It is possible to force the speed connection, but using USB connection it results more reliable and fast. Code is available here: `https://github.com/AliaksandrSiarohin/AppliedRobotics/brofist`

2.2 USB data collection

To establish a connection between PC and NXT brick is possible using a specific Pyton library called ”pyusb”. The procedure to get data with this method is the following:

1. Establish connection with brick using pyusb
2. Set up the motor power on the brick
3. Send (timestamp, tacho count) from PC to brick
4. Receive (timestamp, tacho count)from brick to PC
5. Save collected data to file
6. Go to step 3

Using this methodology is possible to obtain much better performance about $\approx$ $2ms$ latency. Code is available here: `https://github.com/AliaksandrSiarohin/AppliedRobotics/usb_collector`

3 Estimating the parameters from the data

To estimate the parameters we filter the data using butterworth filter and then we estimate the parameters using 2 methods:

- Regular method proposed on the lecture
- Regression method

3.1 Filtering

We use butterwoth filter of order 1 and and cut-off frequency 0.02. For example fig. 1. You can find code for filtering in Filtering. And plot for all the powers in Filtering Plots.

3.2 Parameter Estimation

We need to estimate 3 parameters q, ω_n, ξ. You can find code in Identification

Table 1: Parameters obtained using regular estimation

Power	q	ξ	ω_n
20	0.1573	2.0425	0.5320
30	0.1638	18.0187	0.7611
40	0.1685	0.9028	0.7799
50	0.1720	3.4136	0.8462
60	0.1732	12.7967	0.8444
65	0.1745	14.9685	0.8597
70	0.1733	14.5697	0.8046
75	0.1740	14.7569	0.8493
80	0.1750	15.0400	0.8083
90	0.1746	14.2055	0.8638

3.2.1 Regular method

For the regular method we use the following formulas:

$$q = \text{Last speed value} \tag{1}$$

$$\xi = \sqrt{\frac{\log(\text{overshot})^2}{\pi^2 + \log(\text{overshot})^2}} \tag{2}$$

$$\omega_n = \frac{\log(\frac{alpha}{100}) - \log(\frac{1}{\sqrt{1-\xi^2}})}{-\text{settling time} * \xi} \tag{3}$$

Using this method we obtain 10 sets of parameters which we can see in table 1.

3.2.2 Regression method

Let $\hat{s_t}$ be the speed at time t, and T - finish time, and $s_t(q, \xi, \omega_n)$ will be the speed from our model. The result parameters will be:

$$(q, \xi, \omega_n) = arg \max_{q,\xi,\omega_n} \sum_{t=1}^{T} (s_t(q, \xi, \omega_n) - \hat{s_t})^2 \tag{4}$$

Using this method we obtain 10 sets of parameters which we can see in table 2.

3.2.3 Methods Comparison

For every set of parameters we plot how modeled data and filtered data ax example fig. 2, you can find all the plots in Model comparison plots. For every set of

Table 2: Parameters obtained using regression estimation

Power	q	ξ	ω_n
20	0.1625	28.14	0.9996
30	0.1645	11.73	1.0000
40	0.1691	24.80	0.9990
50	0.1702	24.34	0.9993
60	0.1725	21.38	0.9994
65	0.1740	23.19	0.9995
70	0.1736	21.92	0.9994
75	0.1736	22.47	0.9993
80	0.1752	10.52	1.0000
90	0.1743	10.19	1.0000

Table 3: Mean of mean square error for all data sets.

Power	Regular Method	Regression Method
20	60.71	50.53
30	49.81	49.47
40	73.30	46.96
50	50.96	46.40
60	45.30	45.32
65	44.51	44.55
70	45.07	44.80
75	44.76	44.78
80	44.24	44.07
90	44.52	44.49

parameters we also compute mean of mean square errors for all data sets.

$$\text{result in cell} = \frac{1}{10}\sum_{i=1}^{10}\frac{1}{T}\sum_{t=1}^{T}(s_t^i(q,\xi,\omega_n) - \hat{s_t^i})^2 \tag{5}$$

And we get a result which is in table table 3 (Power is from which data set we obtain a parameters, second column for regular method and third for the regression method). From table 3 we can see that optimal parameters is obtained from (power = 80) data and regression method. We plot filtered data and modeled data with optimal parameters, for example fig. 3. You can find other plots in Result plots.

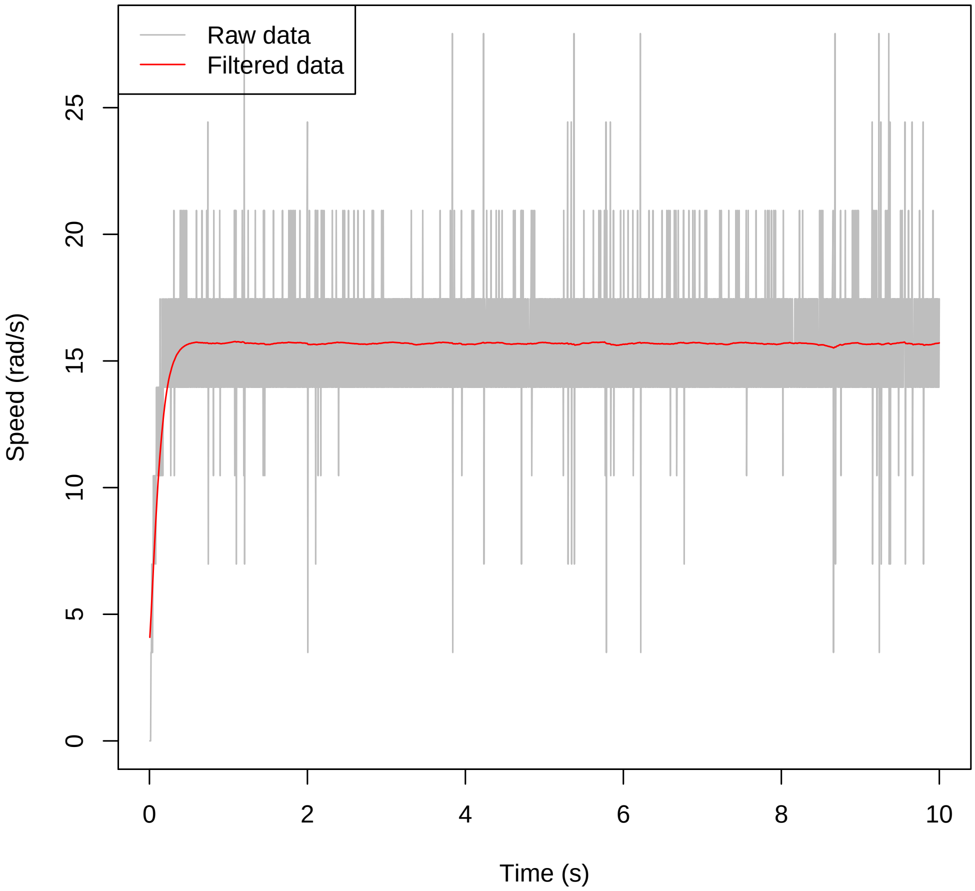

Figure 1: Filtered and raw data for motor power = 90.

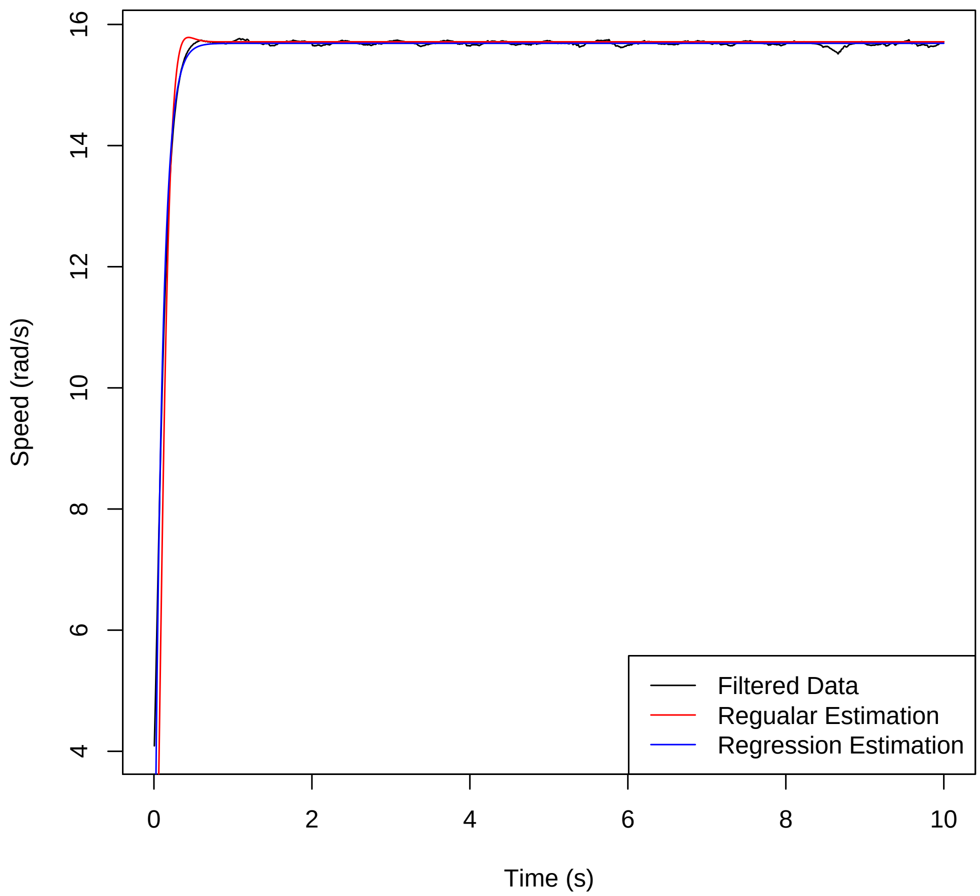

Figure 2: Filtered and modeled data for motor power = 90.

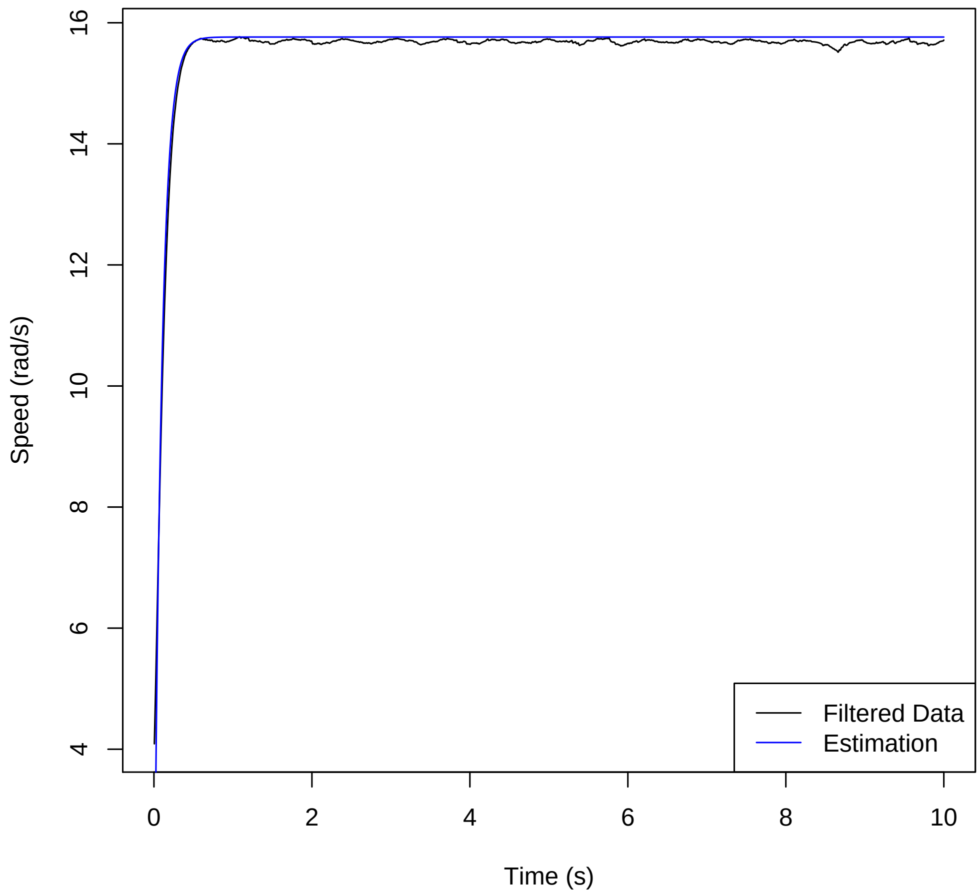

Figure 3: Filtered and modeled data (with result parameters) for motor power = 90.