ELEN90055 Control Systems
Workshop 3

Drive wheel modelling and control for a simulated robot

1 Introduction

The aim in Workshop 3 is to design the drive wheel motor control system for a LEGO MINDSTORMS EV3 robot.[1] The robot has two DC motors as actuators for its two drive wheels; see Figure 1. The workshop involves the application of analysis and design tools introduced in previous workshops. It also provides a bridge to Workshop 4, where a more challenging control problem is considered; that is, balancing the actuated inverted pendulum mounted on the front of the robot, which is shortened and fixed into position for Workshop 3. As in Workshop 2, a model-based approach to open-loop and feedback control system design is explored, including the following:

1. system modelling and parameter identification from measured data;
2. controller design based on a system model and the application of methods for system analysis to certify compliance with specifications;
3. control system simulation based on models and performance evaluation with respect to these models;
4. deployment of the controller in hardware to observe and validate operational performance;
5. iteration from Step 2 (and perhaps 1) if observed performance is poor at step 3 or 4.

Workshop 3 is to be completed over a three week period. A structured report (one per group) is required. The assessment for Workshop 3 contributes 10% to your overall mark for the subject.

Note: Since this workshop is effectively a design project, you are welcome to do any sensible thing beyond what is described below to improve your control design (e.g. devising better models or considering different controllers).

*Prepared by A. Saberi, A. Lang, G. Nair, based on a document by M. Cantoni.

[1]Note that you will not work with the actual robot. Your designed controllers will be tested on a detailed nonlinear model and is purely simulation based.

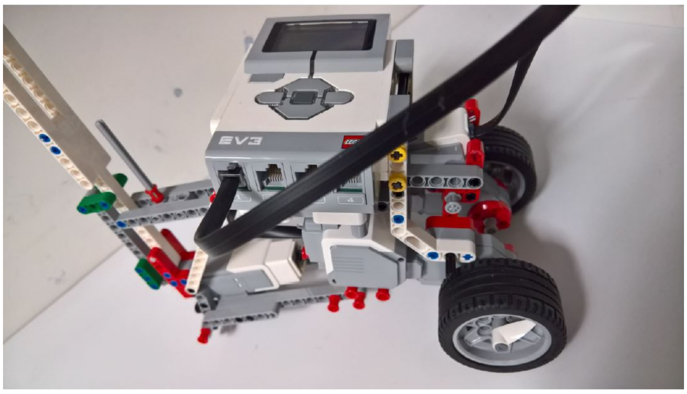

Figure 1: LEGO EV3 two-wheeled robot

2 System modelling

The aim of this workshop is to design control systems for the motor used to actuate the drive wheels of a LEGO EV3 robot. The motor model developed below is a simplistic one. Nonetheless it is useful for feedback control system design. Importantly, the modelling error and corresponding model uncertainty can be accounted for in the design of a feedback controller to robustly meet achievable performance specifications. Such robustness is not achievable via open-loop control based on model inversion.

The robot includes three DC motors. One is for actuating an inverted pendulum mounted on the front. This motor plays no role in Workshop 3, as the pendulum is fixed into place. Each of the other two motors actuates a drive wheel located at the back of the robot. Translation of the robot along a straight line can be achieved by making the wheels rotate in unison, without slip; a difference in the rotation of the wheels re-orients the robot's heading (i.e., differential steering left or right). A model of the system is now developed from physical principles and experimental data, to inform subsequent design of control systems for commanding angular position of the drive wheels under load, and thus, translation and heading re-orientation across an operating surface.

2.1 Physical modelling

With reference to Figure 2, a linear DC motor model involves the relationships

$$e(t) = K_1 \frac{d\theta}{dt}(t) \text{ and } \tau(t) = K_2 i(t).$$

The signal $e(V)$ is the back-EMF voltage induced by the rate of change in magnetic flux linkage of the motor winding as the shaft rotates, and the motor parameters $K_1(V.s/deg)$ and $K_2(N.m/A)$ are called the back-EMF and armature constants, respectively. Given sufficient friction between the drive wheel attached to the motor and the operating surface, so that there is no slip, the mass of the robot presents as an inertial load with moment of inertia $J(kg.m2)$. In addition, a load torque $Bd\theta/dt$ with damping coefficient $B(N.m.s/deg)$ can be used to model the effect of gear train and other viscous friction losses arising

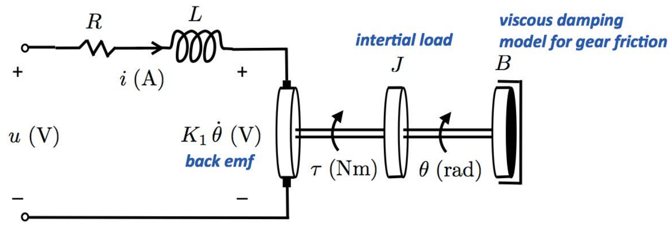

Figure 2: DC motor modelling

from robot motion. Applying Newton's second law on the mechanical side, and Kirchoff's voltage law on the electrical side of the motor, yields the following ODE model:

$$J\frac{d^2\theta}{dt^2}(t) = K_2 i(t) - B\frac{d\theta}{dt}(t) \tag{1}$$

$$u(t) = Ri(t) + L\frac{di}{dt}(t) + K_1\frac{d\theta}{dt}(t). \tag{2}$$

In this model, R (Ohms) is the armature resistance and L (H) is the armature inductance.

Applying the Laplace transform and assuming zero initial conditions (i.e., system initially at rest) gives

$$\frac{\Theta(s)}{U(s)} = \frac{K_2}{s}\frac{1}{(Js+B)(Ls+R)+K_1K_2}$$

In most situations, the contribution of the armature inductance to the dynamics is negligible (i.e., it only gives rise to an additional very fast transient). Setting $L = 0$ yields

$$G(s) := \frac{K_m}{s(T_m s + 1)}, \tag{3}$$

where K_m, and T_m are constant parameters. Rather than using (perhaps measured) estimates of the robot parameters K_1, K_2, B, J, and R, the transfer function parameters K_m, and T_m can be determined directly from experiments as described below.

2.2 Using data to determine model parameters

The Simulink model shown in Figure 3 can be used to perform "Identification" on the Robot system which is a detailed nonlinear model (blue coloured box). Data produced by the outputs include encoders on each motor and movement on $XY-$plane. The model (3) of a DC motor is included in the Simulink model; see the green coloured blocks. The transfer function parameters K_m and T_m can be set from the MATLAB command line.

These parameters can be adjusted by trial-and-error so that "Identification" yields measured and modelled motor angle signals that are well matched in the Simulink scope labelled 'motor angles'; ensure that the magenta coloured switches are configured such that SW1 and SW2 drive the motors with the

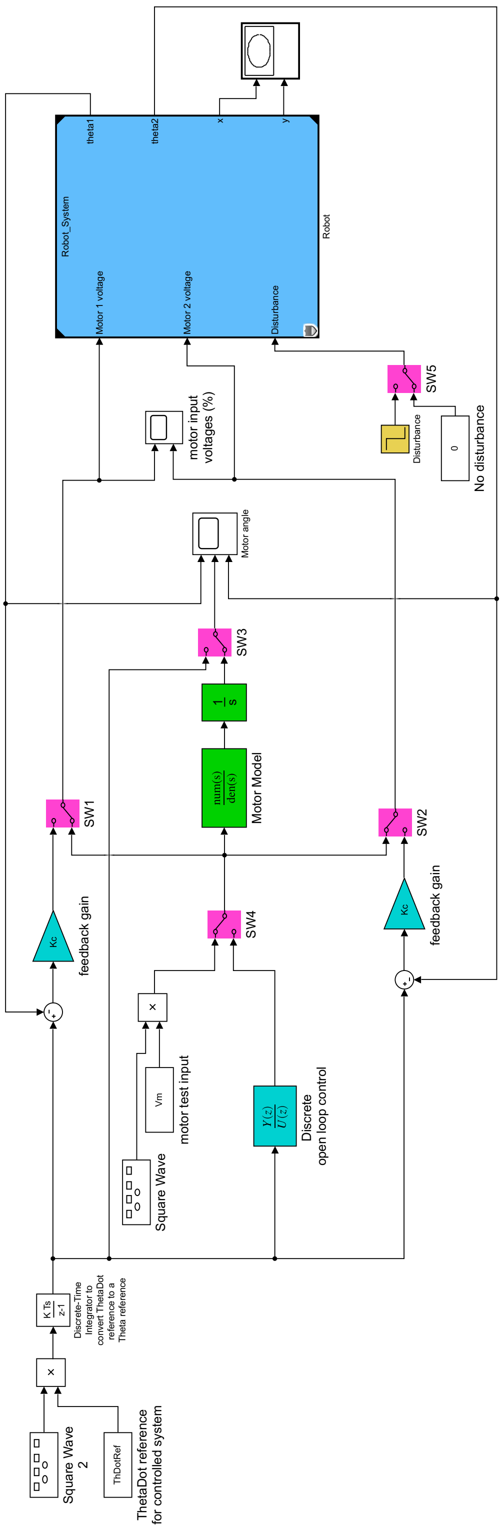

Figure 3: Simulink model of the robot configured for direct command of drive motor (SW1=down, SW2=up, SW3=down, SW4=up, and SW5=down) as required during model development.

motor test input voltage signal (i.e., SW1=down, SW2=up), SW3=down, and SW4=up. Default settings for the model parameters Km and Tm are specified in the MATLAB script Robot_model_param.m, along with other Simulink model parameters; e.g., Vm the motor test input voltage magnitude as a percentage of full supply (%). Take the following steps to run an "Identification" for determining motor model parameters and ranges for these across different operating conditions (some steps don't need repeating):

1. Open the simulink model Robot_model.slx;
2. Run the script Robot_model_param.m at the MATLAB command line to set default parameter values;
3. Check that the magenta coloured switches are configured with SW1=down, SW2=up, SW3=down and SW4=up (ensure switch SW5=down);
4. Adjust values for parameters Vm, Km, and Tm;
5. Run the simulation for 10s.

Exercise. *For different runs, determine motor model parameters Km and Tm that yield a "good match" to both measured motor responses. Note that the two drive motors have slightly different responses. The model parameters should be set to match the two motor measurements equally well. The default square-wave frequency is SqWaveFreq=1/6 (Hz). Increasing the square-wave frequency will yield a smaller range of motion (as the motors will be on in each direction for less time). The point is not to identify very precise values for each scenario. This understanding can be used later to inform control system design. Document the outcomes of your experiments for use when preparing the final report. It is possible to be more systematic than the trial-and-error approach suggested above for identifying parameter values given data. However, such methods involve tools from signal processing and optimization, which would be a substantial digression from the core material of this subject.*

3 Controller design

Command over the angular position of each drive wheel provides a means for translating and re-orienting the heading of the robot across an operating surface with su cient friction to ensure no slip. As explored above,

$$G(s) := \frac{\Theta(s)}{U(s)} = \frac{K_m}{s(T_m s + 1)}$$

is a reasonable model of the transfer function from the motor voltage input to the motor angle under load. Parameter values K_m and T_m can be identified ed as described in the previous section, which must be completed before proceeding. The specifications for control system design are as follows:

1. the controlled system is stable;

2. the angle error (i.e., the difference between the angle reference and the wheel angle) is less than 5% of the constant angular speed (x 1s) in steady-state for a ramp angle reference;

3. the controlled motor voltage for a triangle wave (integrated square wave) wheel angle reference, with speed ThDotRef=210 (deg/s) and frequency SqWaveFreq=1/6 (Hz), does not saturate (i.e., exceed 100%) at any time.

The challenge is to achieve these specifications for a suitable range of motor model parameter values, as informed by the modelling exercise above. Both open-loop and feedback (i.e., closed-loop) control schemes are investigated below. The investigation illuminates the limitations of open-loop control and the robustness properties that closed-loop control can furnish.

3.1 Open-loop control

An open-loop approach to control system design involves the use of a controller that approximates the inverse of the system to be controlled. The controlled system is simply the series interconnection of this approximate inverse compensator with the system dynamics. As explored below, such an approach is limited in terms of robustly achieving performance specifications.

Exercise. *Build a new simulink model to facilitate exploration of open-loop motor control. For suitable nominal motor model parameter values K_{m0} and T_{m0}, consider an open-loop controller with transfer function*

$$C_{OL}(s) = \frac{s(T_{m0}s+1)}{K_{m0}(T_0 s+1)^2},$$

where T_o is a sufficiently small time-constant. Note that

$$G(s)C_{OL}(s) = \frac{1}{(T_o s+1)^2}$$

provided $K_m = K_{m0}$ and $T_m = T_{m0}$. In this case, T_o sets the time-constant of the controlled system, and thus, the frequency bandwidth of angle reference commands that the system output tracks well, even for the nominal model. Obtain analytical step response and investigate the impact of perturbing the plant model parameters K_m and T_m, without adjusting the parameters of the open-loop controller, apply final value theorem and find the steady state output value. Justify your analysis by simulation. Let T_{OL} be the open-loop transfer function, derive the relative-error transfer functions: $\frac{1}{T_{OL}}\frac{\partial T_{OL}}{\partial K_m}\Big|_{(K_m,T_m)=(K_{m0},T_{m0})}$ and $\frac{1}{T_{OL}}\frac{\partial T_{OL}}{\partial T_m}\Big|_{(K_m,T_m)=(K_{m0},T_{m0})}$. Next document the outcome of your analysis and simulation studies for use when preparing the final report.

The Simulink model shown in Figure 3 can be used to perform open-loop control of the kind considered above. Specifically, the light-blue coloured block is the transfer function of a time discretized approximation of the controller $C_{OL}(s)$ for sampled-data realization on the digital EV3 hardware. The code

```
COL = tf([Tm0 1 0],Km0*[To*To 2*To 1]);
COLdisc = c2d(COL,Ts,'tustin');
[COLnum,COLden] = tfdata(COLdisc,'v');
```

in the MATLAB script Robot_model_param.m generates

- the continuous-time open-loop controller transfer function COL, for nominal values Km0 and Tm0 of the motor parameters and the aforementioned nominal controlled system time-constant To;

- the numerator and denominator polynomials of the discrete-time transfer function obtained by applying the Tustin transform $s = \frac{2}{T_s}\frac{z-1}{z+1}$, (i.e. $z = \frac{2/T_s+s}{2/T_s-s}$) to the continuous-time transfer function $C_{OL}(s)$, for given sampling period T_s (set to default value Ts=0.005 (s) in the MATLAB script. Note that the Tustin transform yields the transfer function of a discrete-time system that corresponds to use of trapezoidal approximation in fixed-step numerical integration of a differential equation that has the given continuous-time transfer function.

Re-run these three lines of code (in the MATLAB command line window) to account for adjustments made to the values Km0, Tm0, and To.

Exercise. *Reconfigure the magenta coloured switches to SW1=down, SW2=up, SW3=up, SW4=down (i.e., only change SW3 and SW4 relative to Figure 3). Run the simulation to observe performance of the open-loop controller. Consider the position of the robot (XY graph) and check it moves in a straight line or not; give your reasons. Also, consider disturbing the motion of the wheels (by configuring SW5=up). Compare and contrast the observed behaviour of the robot with the simulations performed during design. Document the outcome of your investigation for use in preparation of the final report.*

3.2 Feedback control

Feedback control can be employed to achieve more robust command over motor shaft angle than open-loop control. As shown in Figure 3, the motor angle sensor can be used to compute an error signal, relative to the commanded position ThetaRef. A constant gain feedback controller (see cyan coloured proportional gain blocks) can be used to set the voltage input to the motor on the basis of the measured error.

Exercise. *Informed by the modelling exercise above, design a constant gain (proportional) feedback controller with transfer function $C_{CL}(s) = K_c$ that complies with the specification given at the start of the section for a loaded motor with model parameters in suitable value ranges. Consider the use of Bode plots of the compensated open-loop and closed-loop sensitivity transfer functions to investigate the effect of varying K_c, and the use of a root-locus plot. Also use an appropriate augmentation of the Simulink model from the open-loop control design exercise, to simulate the closed-loop response to appropriate reference signals for the angular position. Document the outcome of your design process, including supporting analysis, for use in preparation of the final report.*

Exercise. *Using the simulink model shown in Figure 3, run the simulation with feedback control of the drive motors. Ensure that the switches are configured as follows: SW1=up; SW2=down; SW3=up; SW4=down; SW5=down. Consider the position of the robot (XY graph) and compare it with the open-loop control. Consider disturbing the motion of the wheels (by configuring SW5=up). Compare and contrast the observed behaviour of the robot with the* *simulations performed during design. Let T_{CL} be the closed-loop transfer function, derive the relative-error transfer functions: $\frac{1}{T_{CL}}\frac{\partial T_{CL}}{\partial K_m}\Big|_{(K_m,T_m)=(K_{m0},T_{m0})}$ and $\frac{1}{T_{CL}}\frac{\partial T_{CL}}{\partial T_m}\Big|_{(K_m,T_m)=(K_{m0},T_{m0})}$ and compare them with results you obtained in the open-loop control section. Document the outcome of your investigation for use in preparation of the final report.*

4 Report template

The structure of the typewritten report you are required to submit for assessment is outlined below. The strict page limit is 6 pages (11 pt, 1.5 spacing) inclusive of everything (plots, figures, tables, etc.). Each (sub)section below includes a description of the content expected in the corresponding section of the report.

I. Introduction

Briefly describe the objectives of the project description and of Matlab and Simulink, as the tools, and the role of these in executing the project.

II. System modelling

Briefly present the model structure and summarize simplifying assumptions. Include explanation of the method used to obtain suitable model parameter ranges (and nominal values) from measured data, with plots to justify your conclusions. Explain observed behaviour that is not captured by the simple model (i.e., the of source difference between modelled and observed data).

III. Control design

Present the outcome of open-loop and closed-loop controller design exercises. Include justification for design choices (e.g., mathematical analysis, plots, etc), and simulations of performance against specifications.

IV. Implementation and testing

Present performance of open- and closed-loop control. Compare and contrast real behaviour on the protected Simulink model with behaviour on the simplified G(s). Validate that the performance specifications are met, and if not, explain why the behaviour is different from the simplified model. Use plots to illustrate.

V. Conclusions

Any relevant discussions including things that you have learned and succeeded in accomplishing. Briefly consider possible directions for improving your control design.

5 Summary of main points

By the end of Workshop 3, you should have gained an improved understanding of model-based control design methods, within the context of solving a practical problem. Model-based design provides scope for rigorous analysis as part of the design flow. Of equal importance is the hands-on experience you should have gained, as well as an appreciation of the relative merits of open-loop and closed-loop control.