MCEN90017: Advanced Motion Control
Lab 2: Modelling and Control of an Active Suspension Experiment (Weeks 6-9)

Report Due Date: 11:59PM on Sunday May 7, 2017

1 Introduction

In this assignment you will work with the Active Suspension Experiment (ASE) equipment and design a controller for it. You will be working in groups of 2. Each group will be required to submit a report via the LMS site of not more than 10 pages on the findings related to the tasks given in the assignment.

The University of Melbourne takes plagiarism seriously. Each group are required to submit a report that complies with the university policies on plagiarism. It is your responsibility to read and understand the policies as detailed in https://academichonesty.unimelb.edu.au/.

2 Learning objectives

- Experience of mathematical modelling of a control plant
- Design and implementation of model-based controllers

3 Before you start

ASE is a technology that allows achieving greater degree of ride comfort and car handling. A demonstration of such a system in action can be seen here:

https://www.youtube.com/watch?v=eSi6J-QK1lw

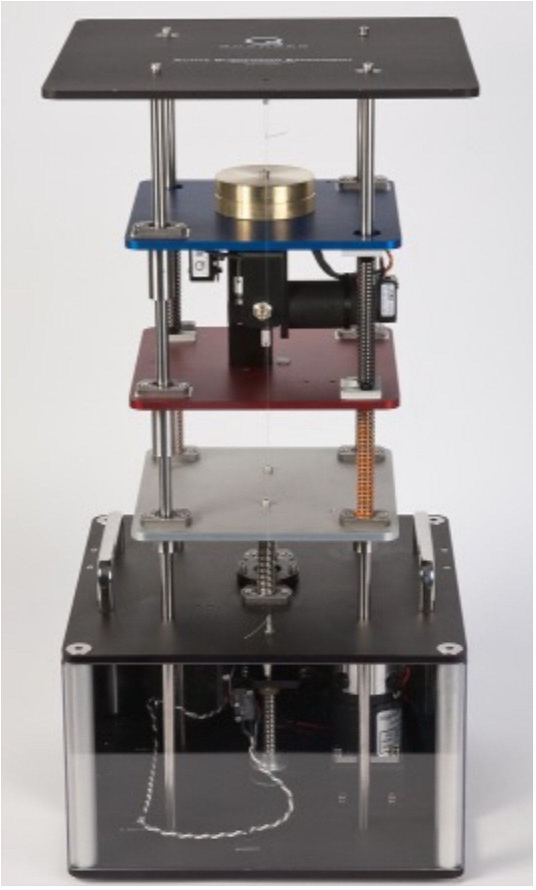
In this lab you are tasked to design controllers for a quarter car active suspension system.

3.1 The ASE equipment

The setup is a Quanser Active Suspension Experiment, which is composed of 3 main components; [1]The ASE equipment (see Figure 1 to Figure 4 and Table 1), [2]amplifier, and [3]the I/O device compactRIO (cRIO) system from National Instruments.

This ASE system consists of two masses, each supported by a spring and a damper. The sprung mass, represents the mass of the vehicle body while the unsprung mass, represents the tire in the quarter-car model. This is a fourth order system because there are four independent storage elements, the two masses and the two springs.

In this lab, the controller will be designed and implemented in LabVIEW and then will be compiled and transferred to the cRIO which controls the equipment in real-time.



Top Plate (Blue, Vehicle Body Mass)

Middle Plate (Red, Vehicle Tire Mass)

Bottom Plate (White, Road)

Figure 1: Quanser Active Suspension Experiment

ID	Component	ID	Component
1	Top Plate (Blue, Vehicle Body Mass)	2	Accelerometer Gain Potentiometer
3	Middle Plate (Red, Vehicle Tire Mass)	4	Suspension Encoder
5	Bottom Plate (White, Road)	6	Suspension Motor Capstan Cable
7	2 Adjustable Springs (Vehicle Suspension Springs)	8	Spring Holder Set Screw
9	2 Adjustable Springs (Vehicle Tire Springs)	10	Linear Bearing Blocks
11	Bottom Plate Encoder Connector	12	Top Plate Encoder Connector
13	Bottom Plate Counter Weight Springs	14	Payload Mass (Brass)
15	Active Suspension DC Motor	16	Bottom Plate Encoder
17	Bottom Plate Servo Motor	18	Lead Screw
19	Encoder Thread	20	Stainless Steel Shafts
21	Accelerometer	22	Accelerometer RCA Connector
23	Plant Top Cover	24	Limit Switch Safety Lights
25	Plant Handles	26	Bottom Plate Motor Connector
27	Limit Switch Push Key	28	Safety Rod
29	Movable Spring Holders	30	Safety Limit Switch
31	Suspension Motor Connector	32	Suspension Encoder Connector
33	Encoder Thread Anchor	34	Top Plate Encoder

Table 1: Active Suspension Experiment Components

Caution: ***High-frequency signal applied to a motor will eventually damage the gearbox motor and the motor brushes. To protect the motor, you should always band limit your signal to a value within 10Hz.***

Note: The limit switch (Component #30) gets triggered when the bottom plate (Component #5) moves too close to its vertical range limits. In such a case this solenoid sensor directly deactivates the power of the servo motor (Component #17) and the red light will be on (Component #24). In this situation the simulation should be stopped by the operator in the software and the bottom plate (Component #5) should be moved inside its working range. At this point the two red and green lights will be off, meaning that the plate is inside its working range but the motor is still not active. Then the limit switch push-key button (Component #27) should be pressed in order to activate the servo motor (Component #17) again.

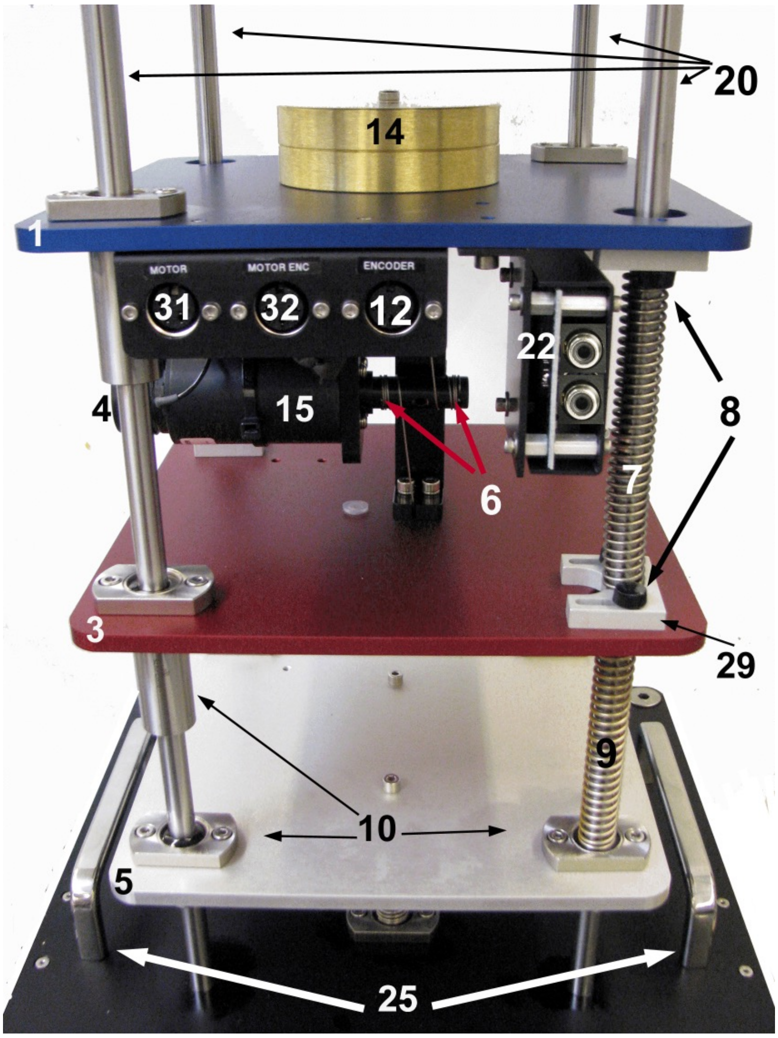


Figure 2: Active Suspension Experiment Top Front View

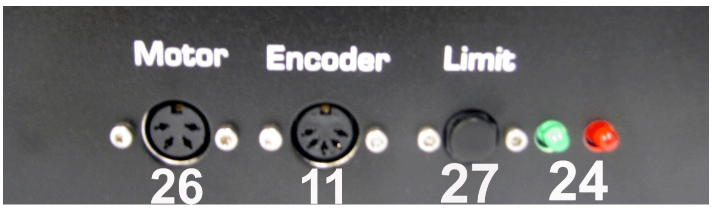
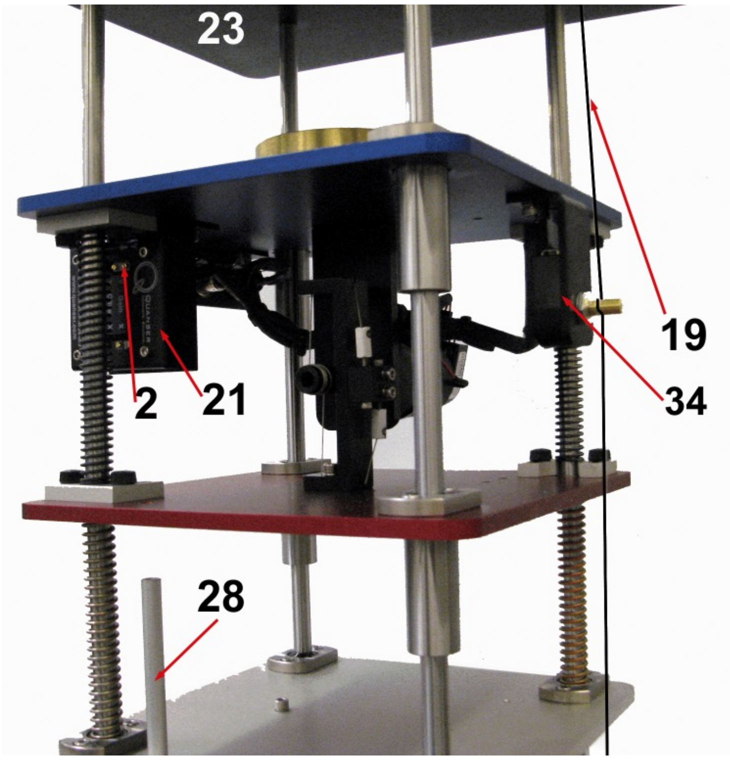
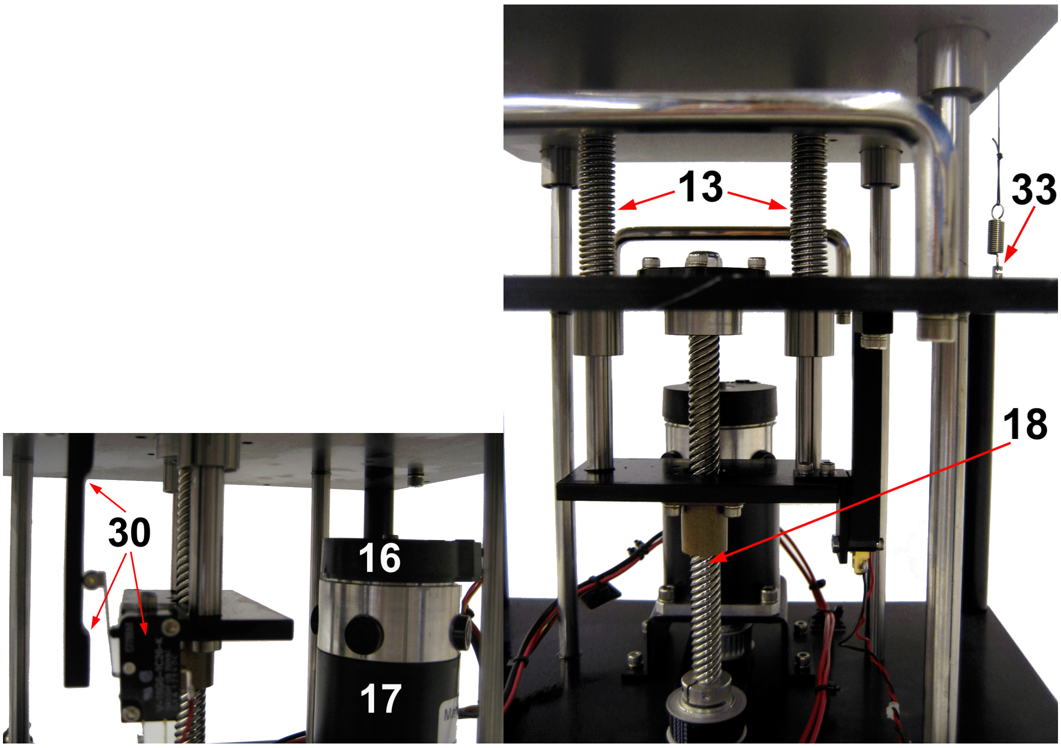


Figure 3: Active Suspension Experiment Bottom Front Panel



(a) Side View



(b) Base Inside View

(c) Base Inside View

Figure 4: Active Suspension Experiment Side View

3.2 Cable connections

Refer to the appendix for the wiring of the setup.

Note: Due to variability in dynamics for different ASE, please use the same ASE throughout the lab in order to have consistency in your work. Hence, please record the equipment number of the ASE.

3.3 List of VIs

You are given the following VIs to do the tasks;

1. *ASE Open Loop.vi* – is useful to do the mathematical modelling of the equipment
2. *ASE Experiment Control.vi* – is used to implement a real-time controller for the equipment. In this file, take note of the switch labelled “Control Mode” – which is the on/off switch for the controller.

4 Control design

You are to design controllers that actively drive the suspension so that they maximise the performance of the system. The performance of the system is measured by the following three metrics. Different vehicle types place different importance on these metrics. For example, a sports car will typically sacrifice Ride Comfort to achieve better Road Handling, whilst the opposite is true in a luxury vehicle.

- *Ride Comfort* is related to vehicle body motion sensed by the passengers. A measure for the Ride Comfort is the acceleration of the sprung mass in the quarter car model.

- *Suspension Travel* refers to relative displacement between the vehicle body and the tire. It is constrained within an allowable workspace and excessive suspension travel can lead to premature ageing of the system. In the equipment, relative displacement between the sprung mass and the unsprung mass represents Suspension Travel.

- *Road Handling* is associated with the contact forces between the road surface and the vehicle tires. These forces provide the friction between the road and the tires in a real car. They depend on the tire deflection. In a quarter car model, the displacement between the unsprung mass and the road represents the tire deflection.

Tip: Placing any variable that need to be modified (for e.g. controller parameters) on the front panel instead of the block diagram will speed up the running/compiling of VIs.

TASK 1 Mathematical modelling

The mass-spring damper model is as illustrated in Figure 5. With this model, the two inputs to the system are considered to be active suspension control command F_c and the road surface position z_r. Furthermore, it is reminded that the reference frames in the figure are used to choose the generalized coordinates, i.e. x_1 and x_2.

The generalized coordinate x_1 represents the tire displacement (unsprung mass in quarter car model) and x_2 represents the vehicle body displacement (sprung mass in the quarter car model) all with respect to the points where the springs are relaxed. The positive directions are upwards.

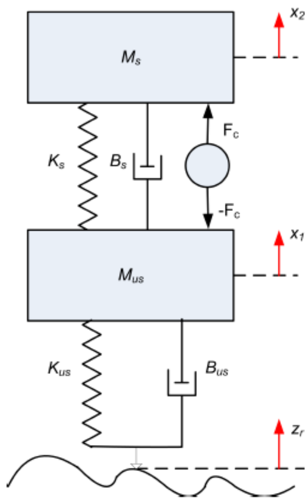


Figure 5: Double mass-spring-damper model used to model the ASE.

Parameter	Name	Unit
M_s	Sprung mass	(Kg)
M_{us}	Unsprung mass	(Kg)
K_s	Suspension stiffness	(N/m)
K_{us}	Tire stiffness	(N/m)
B_s	Suspension inherent damping coefficient	(Ns/m)
B_{us}	Tire inherent damping coefficient	(Ns/m)

Table 2: ASE equipment properties

Derive the governing differential equation of motion of ASE for given M_s, M_{us}, K_s, K_{us}, B_s, and B_{us}, where M are the masses, K the spring constants, and B the damping constants. To find out equations of motion (EOM) for this system, the free body diagram for each mass should be determined. There are two masses in the system and the forces applied to each mass should be drawn on the diagrams. There will be two equations of motion.

In the second step, you need to prove that the gravity force only changes the equilibrium points in the ASE EOM and it does not affect the dynamics of the system.

Hints: After you obtain the dynamic equations of the mass-spring-damper system, find the *equilibrium* point of x_1 and x_2. Also, you might find the following change of variable from x_1 and x_2 to z_{us} and z_s useful;

$$x_1 = z_{us} - \frac{M_s + M_{us}}{K_{us}} g \tag{1}$$

$$x_2 = z_s - g\left(\frac{M_s}{K_s} + \frac{M_s + M_{us}}{K_{us}}\right) \tag{2}$$

Where z_{us} and z_s are the deflections of the unsprung and sprung mass, respectively, from the equilibrium state. With the change of variable, the equilibrium will be at $z_{us} = 0$ and $z_s = 0$.

Finally you should derive the following state-space representation of the model in the form

$$\begin{aligned} \dot{x} &= Ax + bu \\ y &= Cx + Du \end{aligned} \tag{3}$$

Where

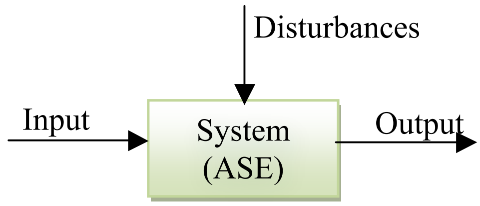
$$x = \begin{bmatrix} z_s - z_{us} \\ \dot{z}_s \\ z_{us} - z_r \\ \dot{z}_{us} \end{bmatrix}, \quad u = \begin{bmatrix} \dot{z}_r \\ F_c \end{bmatrix}, \quad y = \begin{bmatrix} z_s - z_{us} \\ \ddot{z}_s \end{bmatrix} \tag{4}$$

$$\begin{gathered} A = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -a_1 & -a_2 & 0 & a_2 \\ 0 & 0 & 0 & 1 \\ a_3 & a_4 & -a_5 & -a_6 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & b_1 \\ -1 & 0 \\ b_2 & -b_3 \end{bmatrix} \\ C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -a_1 & -a_2 & 0 & a_2 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & b_1 \end{bmatrix} \end{gathered} \tag{5}$$

The four states are the suspension deflection/travel, the vehicle body velocity, the tire deflection (which is a measure of road handling), and the tire velocity. The first input to the system is the road surface velocity, and the second input is the control action that will be later designed. The first measured output of the system is the suspension travel. Assuming that the vehicle body is equipped with an accelerometer, the second measured output of the system will be the body acceleration.

The values of a_i and b_i are terms involving the properties of the mass/spring/damper (Table **2**). They should be found by performing system identification.

System Identification



System identification is a way of building mathematical models of dynamical systems subject to unknown disturbances based on measured data from the systems. The intention is to obtain an appropriate model that captures the dynamical behaviour of the system (i.e. ASE) relevant for a particular purpose, in this case for control design.

1) In order to carry out system identification, input-output data are required. Hence, the first step is to collect data by running ASE in open loop, i.e. run *ASE Open Loop.vi* with appropriate input signals and sampling time to obtain persistently excited data to identify model that capture the relevant dynamics useful for control design.

Tips:

Data extraction can be done easily using a Waveform Chart in LabVIEW. Right-clicking a waveform chart and selecting Export → Export Data to Excel will do the job.

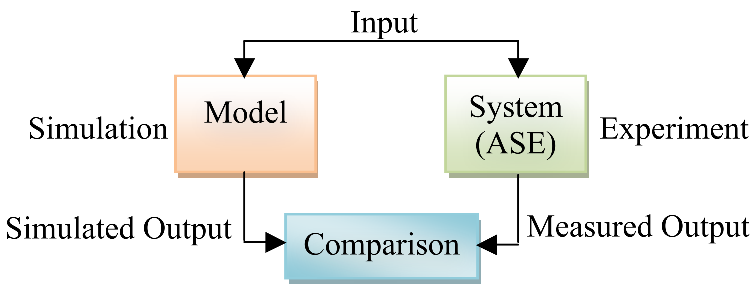
Due to limited lab sessions, hence limited access to ASE, it is recommended that as many sets of data (with different appropriate input signals) are collected as possible during lab sessions. Then, analysis and system identification can be performed outside of lab time. In addition, extra data set is also required for model validation

2) As the model derived is in continuous time, in order to perform system identification the model needs to be discretised, for e.g. using zero order hold method.

3) Then, find the values of a_i and b_i using system identification methods (for e.g. least square) based on the data collected from the ASE.

Tip: note that $\boldsymbol{a_i}$ and $\boldsymbol{b_i}$ only appeared in the second and fourth rows of matrix A and B in (5).

4) Validate the model obtained by comparing the responses simulated by the model against the real data that are not used to identify the parameters.



5) Analyse and discuss your results.

TASK 2 LQR design and implementation

LQR is an unconstrained model-based controller. It finds the optimal input $u = F_c$ from the infinite-horizon optimisation problem

$$\begin{aligned} &\min J := \int_0^\infty x^{\mathrm{T}} Q x + R u^2 \;\; dt \\ &\text{subject to } \dot{x} = Ax + Bu \end{aligned} \tag{6}$$

The performance index J penalizes the states of the system, i.e., suspension travel and tire deflection as the two performance measures, as well as the body and tire velocities through the weighting matrix Q. The weighting matrix Q is a symmetric positive semidefinite matrix and has to be full rank. It is typically a diagonal matrix. Each entry of the diagonal represents how heavy ones want to penalise deviations in the corresponding state from zero.

The performance index also reflects the control limitations by penalizing the control input through the weighting gain R. The weighting matrices affect how LQR minimizes the function and are, essentially, tuning variables.

The solution to the optimisation problem is

$$F_c = -Kx \tag{7}$$

Where the gain K is given by

$$K = R^{-1} B^{\mathrm{T}} P \tag{8}$$

and P is the solution of the continuous time Riccati differential equation. Wikipedia has a good entry on LQR design you can refer to for how to calculate P [1]. Note that you only need to consider the second input F_c when designing the gain, as the first input z_r acts as a disturbance.

Design an LQR controller in MATLAB by simulating the ASE using the model you obtained in TASK 1. Design this controller to handle a simulated pulse road bump of amplitude 0.01m at a frequency of 0.3Hz.

To get the gain K, you can use MATLAB's built-in function `lqr`. Think about the performance objectives you need to achieve. After you get the LQR state-feedback gain K, implement the controller in LabVIEW to control the equipment in real-time. Analyse and discuss your results.

TASK 3 MPC design

Not only that you need to maximise the performance of the suspension system as measured by the three metrics given previously, there are some performance specifications that you need to satisfy. These specifications come in the form of constraints.

Model predictive control (MPC) is a model-based controller, which explicitly incorporates constraints into the feedback controller. To maintain computational tractability it is applied to a discrete time model of the plant, leading to a formulation of an online optimisation problem such as:

$$
[u_k, u_{k+1}, \ldots, u_{k+N}] = \underset{u_k, u_{k+1}, \ldots, u_{k+N}}{\arg \min} J := \sum_{k=1}^{N} x_k^{\mathrm{T}} Q x_k + R u_k^2
$$

$$
\begin{aligned}
& x_{k+1} = A_d x_k + B_d u_k \\
\text{subject to:} \quad & x_{lb} < x_k < x_{ub} \qquad \text{for all } k = 1 \ldots N \\
& u_{lb} < u_k < u_{ub}
\end{aligned} \tag{9}
$$

which has the solution u_k^*, x_k^* for all $k = 1 \ldots N$. The MPC control that is implemented on the system is given by the first element of the control trajectory:

$$
u = u_1^* \tag{10}
$$

Notice the difference between LQR in (6) and the MPC problem above. First, you should see that the performance index is now summed over a finite horizon. Also, notice that there are additional (performance) constraints bounding the states and input of the system.

The performance specifications that need to be satisfied are:

Specification	Constraint
Actuator limit (input constraint)	$\lvert F_c \rvert \le 4\text{N}$
Maximum suspension travel	$\lvert z_s - z_{us} \rvert \le 0.01\text{m}$
Maximum vehicle body velocity	$\lvert \dot{z}_s \rvert \le 0.2\text{ms}^{-1}$

Table 3: Performance specifications

To guarantee feasibility of the closed loop system, it is typically necessary to add a terminal constraint to the problem formulation in (9). This involves an additional (non-performance) constraint requiring the state at the end of the horizon to be within a terminal set, X_f:

$$
x_{k+N} \in X_f \tag{11}
$$

Cast the MPC optimisation problem (9) - (11) as a quadratic program (QP), which you can solve using LabVIEW's native QP solver or using Matlab.

Now, design an MPC controller to handle a simulated pulse road bump of amplitude 0.01m at a frequency of 0.3Hz.

Investigate what happens if you impose some or all constraints for the given road condition.

<u>Note</u>: You only need to implement the MPC in simulation. You do not need to implement MPC in real-time to control the equipment due to computational restrictions on the platform.

Tune the MPC by adjusting the prediction horizon (represented by N) and the cost weighting matrices Q and R to get a desirable performance.

Discuss the effect of N and the weighting matrices. In addition, discuss the trade-off between performance and computation time.

TASK 4 Comparison of MPC and LQR

To satisfy the performance specifications given in 3, try tuning the LQR by changing Q and R. Discuss and compare the performance of LQR with that of the tuned MPC, in terms of the performance measures and performance specifications.

Note: As you are not implementing MPC in real-time, do this task in LabVIEW or Matlab simulation only.

How could MPC be implemented in real-time? Also, when are the LQR and MPC are basically the same?

5 Assessment

Report (20 marks)

Prepare a report to discuss your findings. The report should:

1. Address and discuss the tasks given above including all data used,
2. Display sound understanding of the controllers that have been implemented, and
3. Discuss any other relevant thoughts.

6 Report submission

Prepare a report as per the task given as a Portable Document (.PDF) or MS Word Document (DOC, DOCX). Use an appropriate engineering report format. A small portion of the assignment mark will be assigned to the formatting of the report

Attach the relevant VIs and Matlab files with your report. Please remember to include the names of the group members in your report. Include a scanned copy of the plagiarism coversheet signed by your group members.

Compress the report as well as the relevant VI and Matlab files as a .ZIP file and send it to LMS before the due date.

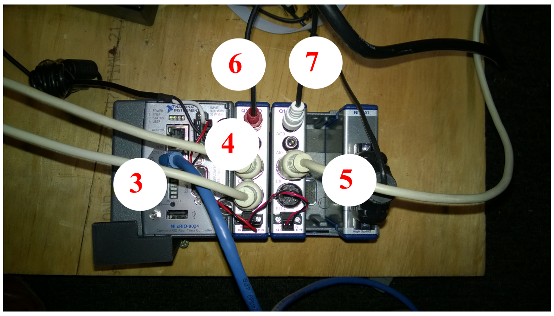
7 References

[1] http://en.wikipedia.org/wiki/Linear-quadratic_regulator

Revision history:
Created by Jalil Sharafi &Vincent Bachtiar (2014-5)
Revised Su Ki Ooi (Mar 2016)
Revised Chris Manzie (March 2017)

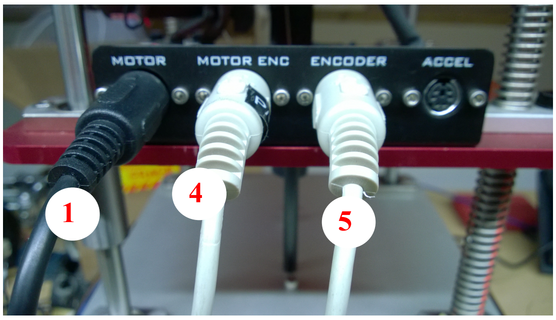
8 Appendix

8.1 Equipment setup wiring

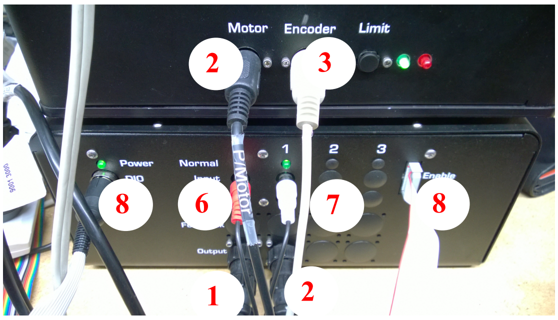


cRIO

ASE base



ASE middle plate



Amplifier