THE UNIVERSITY OF REGINA

Faculty of Engineering and Applied Science

ENIN 833– COMPUTER AIDED PROCESS ENGINEERING

SUMMER 2018

Project Report on:

GAIN SCHEDULED IMPLICIT AND EXPLICIT MPC CONTROL OF MASS-SPRING SYSTEM

SUBMITTED BY:

SHUBHAM SHARMA (Student Id: 200398667)

PIJOE JACK PETER (Student Id: 200391997)

TABLE OF CONTENTS

1 LIST OF FIGURES:

2 LIST OF NOMENCLATURES:

1. MPC ---- Model Predictive Control
2. M -------- Mass
3. g --------- Gravitational Acceleration
4. k --------- Spring Constant
5. F --------- Pulling Force
6. b------- --Spring constant

3 ABSTRACT:

This project mainly focuses on ‘Gain Scheduled Implicit and Explicit MPC Control of Mass-Spring System. Here we used the method of gain scheduling, i.e. we made more than one linear controller and activated those one at a time with the help of a scheduling signal to control a non-linear plant. Two different MPC control techniques are used for this process named as Implicit MPC Control and Explicit MPC control. Implicit MPC control is the traditional method for the applications and is mainly used for the systems having large sampling times and for the systems having larger computational times. Explicit MPC controllers are used for the applications having lower computational times and for the systems having smaller computational times. In this project, two Implicit MPC controllers and two explicit MPC controllers have been generated for the two fold dynamics of the current Mass Spring system problem. Pull Force is the system input and the position of mass M1 is the output. Hence with the help of Simulink we are tracking position of mass M1 with a reference value R while using two different techniques. Finally, the output of both the techniques is measured and compared to know which is better technique in this project.

4 INTRODUCTION AND LITERATURE REVIEW:

The motivation behind this project is to implement a Gain Scheduled MPC controller on a mass-spring system and see its performance. Mass-spring systems are most commonly used in automobiles and with the help of this project; the performance of every vehicle using this system will improve. In this project, we use a simple system consisting of two masses connected to two springs. A mass-spring system is a system of masses, which are connected by springs with several degrees of freedom. For example, a system consisting of two masses and three springs will have three degrees of freedom. This explains that its configuration illustrates two generalized coordinates that are chosen to be the displacements of the first and second mass from the equilibrium position. It is essential to understand the amplitude, period and frequency of mass-spring systems for many reasons. In a car, the springs must be made in such a way that it osculates at the right frequency and with the right amplitude.

Model predictive control (MPC) is a new method of process control developed in the 1980's that is used for controlling a process while satisfying a set of constraints. Model predictive control (MPC) deals with a class of computer control algorithms, which utilize an explicit process model to predict the future responses of a plant. At each of the control intervals, an MPC algorithm attempts to optimize future plant behaviour by calculating a sequence of future manipulated variable adjustments. The first input in this optimal sequence is then sent into the plant, and the entire calculation is done over and over at subsequent control intervals. Originally developed to fulfil the specialized control needs of power plants and petroleum refineries, MPC technology are now being used in a wide variety of application areas including chemicals, food processing, automotive, and aerospace applications.

The models that are used in MPC are intended to represent the behaviour of complex dynamical systems. MPC models are able to predict the change in the dependent variables of a modelled system that will be caused due to the changes in the independent variables. While doing a chemical process, independent variables, which can be adjusted by the controller, are mostly either the set points of a regulatory PID controller (i.e., pressure, flow, temperature, etc.) or the final control elements (valves, dampers, etc.). Independent variables that cannot be adjusted by the controller are considered as disturbances. In these processes, the dependent variables are other measurements that represent either process constraints or control objectives.

A lot of recent publications give us a good introduction to theoretical and practical issues that are associated with MPC technology. Rawlings J. B. (2000) gives an excellent introductory tutorial aimed at control practitioners. Allgower, Badgwell, Qin, Rawlings, and Wright (1999) also presents a more comprehensive overview of nonlinear MPC and moving horizon estimation, which also includes a summary of recent theoretical developments and a lot of numerical solution techniques. Mayne, Rawlings, Rao, and Scokaert (2000) also provide a comprehensive review of theoretical results that are obtained on the closed-loop behaviour of MPC algorithms. A lot of books on MPC have been published recently (Allgower & Zheng, 2000; Kouvaritakis & Cannon, 2001; Maciejowski, 2002).

Several papers were presented reviewing the industrial MPC applications using non-linear models at the 1998 Non-linear Model Predictive Control workshop, which was held in Ascona, Switzerland (by Qin and Badgwell, 2000). Froisy (1994) and Kulhavy, Lu, and Samad (2001) described about the industrial MPC practice and future developments. In the recent years, the MPC landscape has seen a drastic change, with a large amount of increase in the number of reported applications, and notable improvements in technical capability.

5 PROBLEM DEFINITION:

```
                         /-----\     k1                ||
                  F <--- | M1   |----/\/\/\------------[|| wall
    ||                   | |---/                       ||
    ||     k2            \-/      /----\               ||
wall||]--/\/\/\-------------------| M2 |               ||
    ||                            \----/               ||
    ||                                                 ||
----yeq2----------------- y1 ------ y2 ---------------yeq1----> y axis
```

Fig. 6.1

Two masses M1 (1kg) and M2 (5kg) are connected with spring constants k1 (1N/m) and k2 (0.1N/m). The friction coefficient of masses M1 and M2 are 0.3 and 0.8 respectively. The initial wall mount positions of masses M1 and M2 are 10 and -10 units. Mass M1 is pulled by a force F that is a manipulated variable. Position of mass M1 and a contact sensor are available for feedback.

```
M1 = 1;          % mass
M2 = 5;          % mass
k1 = 1;          % spring constant
k2 = 0.1;        % spring constant
b1 = 0.3;        % friction coefficient
b2 = 0.8;        % friction coefficient
yeq1 = 10;       % wall mount position
yeq2 = -10;      % wall mount position
```

Fig. 6.2

This system follows a two-fold dynamics:

1a.) When force F is applied, masses are detached and mass M1 moves freely

1b.) When force F is removed, M1 and M2 move together

2.) This collision is completely inelastic

Gain Scheduling is an approach to control non-linear systems and it uses a family of linear controllers. Each controller provides a satisfactory control for different operating points in a system. In this method, the system uses scheduling variables to determine the system's current operating region and to enable an appropriate linear controller. An example of this system is the aircraft flight control system in which the Mach number and the altitude are the scheduling variables and they have different linear controllers.

To improve the efficiency of a system, inactive controllers do not determine optimal control moves. In order to provide a bump less transfer between controllers, the inactive controllers continue to perform state estimation. The main purpose of bump less transfer is to prevent sudden changes in the operating variables when the controller switching occurs.

Gain Scheduled MPC design switches between a pre-defined set of MPC controllers in a coordinated fashion. It is able to control a non-linear plant over a wide range of operating conditions. This approach is mainly used when a single prediction model cannot provide adequate controller performance.

The first case in this system is where a pull force F is applied to the model and mass M1 moves freely. When this force F is removed, masses M1 and M2 moves together. In this system the states are the position and velocity of mass M1 and the manipulated variable is the Pull force F. The measured disturbance is a constant value of 1, which provides calibrated spring force to the right value, and the measured output is the position of mass M1.

The output is found using two techniques:

1.) Implicit MPC Control
2.) Explicit MPC Control

Implicit MPC Control is the traditional MPC for the applications. In this design technique, we use two MPC linear controllers 1.) MPC controller for the case when mass M1 detaches from M2 (MPC1). 2.) MPC controller for the case when mass M1 and M2 are together (MPC2) and then a scheduling signal is designed for switching the controllers at run time.

Explicit MPC Control uses offline computations to determine all operating regions in which the optimal control moves are determined by evaluating a linear function. They require fewer run-time computations than traditional MPC and are useful for applications that require small sample times.

6 SOLUTION METHODOLOGY:

The common parameters in this system are as follows:

```
Ts = 0.2;    % sampling time
p = 20;      % prediction horizon
m = 1;       % control horizon
```

GAIN SCHEDULED IMPLICIT MPC DESIGN:

The first step is to design a MPC controller for the case where M1 detaches from M2. The code given below explains the design.

```
MPC1 = mpc(sys1,Ts,p,m);
MPC1.Weights.OV = 1;
```

The constraints for MPC1 are specified as follows:

```
MPC1.MV = struct('Min',0,'Max',30,'RateMin',-10,'RateMax',10);
```

The next step is to design a MPC controller for the case where M1 and M2 move together. The code given below explains the design.

```
MPC2 = mpc(sys2,Ts,p,m);
MPC2.Weights.OV = 1;
```

The constraints for MPC2 are specified as follows:

```
MPC2.MV = struct('Min',0,'Max',30,'RateMin',-10,'RateMax',10);
```

The simulation process is done using three different steps:

1.) Gain Scheduled MPC control is simulated using Multiple MPC Controllers Block. The below given code explains this process.

```
disp('Start simulation by switching control between MPC1 and MPC2 ...');
disp('Control performance is satisfactory.');
open_system([mdl '/signals']);
```

```
sim(mdl);
MPC1saved = MPC1;
MPC2saved = MPC2;
```

This step explains that use of two controllers provides good performance under all conditions.

2.) Gain Scheduled MPC control is simulated using MPC1 only (when masses are not in contact). The below given code explains this process.

```
disp('Now repeat simulation by using only MPC1 ...');
disp('When two masses stick together, control performance deteriorates.');
MPC1 = MPC1saved;
MPC2 = MPC1saved;
sim(mdl);
```

Simulation using MPC1 only denotes that the performance degrades whenever two masses joins

3.) The last step is where Gain Scheduled MPC control is simulated using MPC2 only (when masses M1 and M2 are in contact). The below given code explains this process.

```
disp('Now repeat simulation by using only MPC2 ...');
disp('When two masses are detached, control performance deteriorates.');
MPC1 = MPC2saved;
MPC2 = MPC2saved;
sim(mdl);
```

Simulation using MPC2 only denotes that the performance degrades whenever two masses are separated causing the controller to apply excessive force.

GAIN SCHEDULED EXPLICIT MPC DESIGN:

An explicit MPC design is created by first defining the operating ranges for the input signals, controller states, and reference signals. An explicit MPC range object is created using the corresponding traditional controller, MPC1.

```
range = generateExplicitRange(MPC1saved);
```

After the operating ranges are defined two Explicit MPC controllers that correspond to MPC1 and MPC2 are designed.

For MPC1:

```
expMPC1 = generateExplicitMPC(MPC1saved,range);
```

For MPC2:

```
expMPC2 = generateExplicitMPC(MPC2saved,range);
```

In order to implement gain-scheduled explicit MPC control, the Multiple MPC Controllers block is replaced with the Multiple Explicit MPC Controllers block and the simulation is done.

```
expModel = 'mpc_switching_explicit';
open_system(expModel);
sim(expModel);
```

7 RESULTS:

Results coming by implementing both implicit and explicit models into Simulink have been shown and explained in this section.

1. IMPLICIT MPC CONTROL TECHNIQUE:

   a) When only Implicit MPC1 is activated:

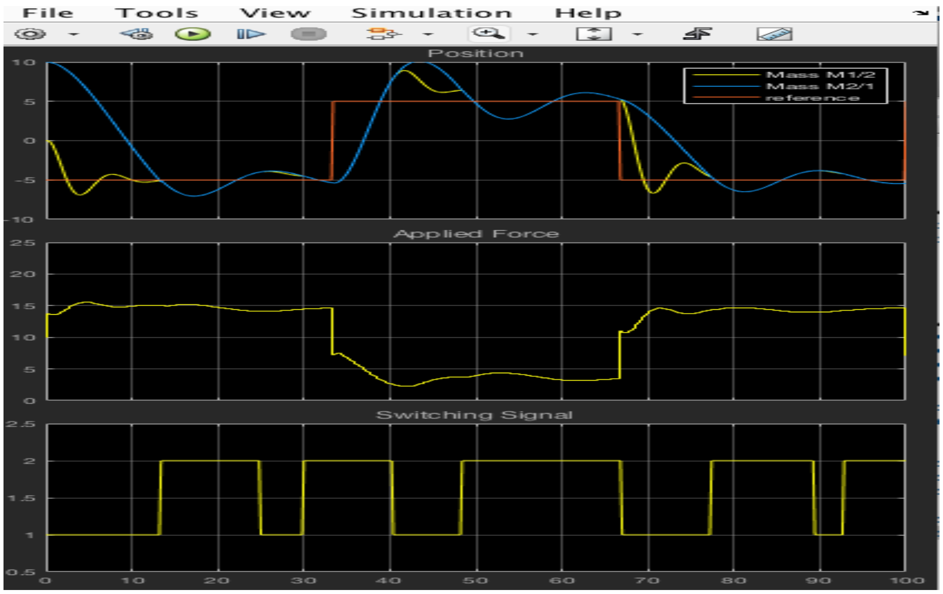

Fig. 7.1

We can see that performance of system degrades whenever two masses joins.

   b) When only Implicit MPC2 is activated:

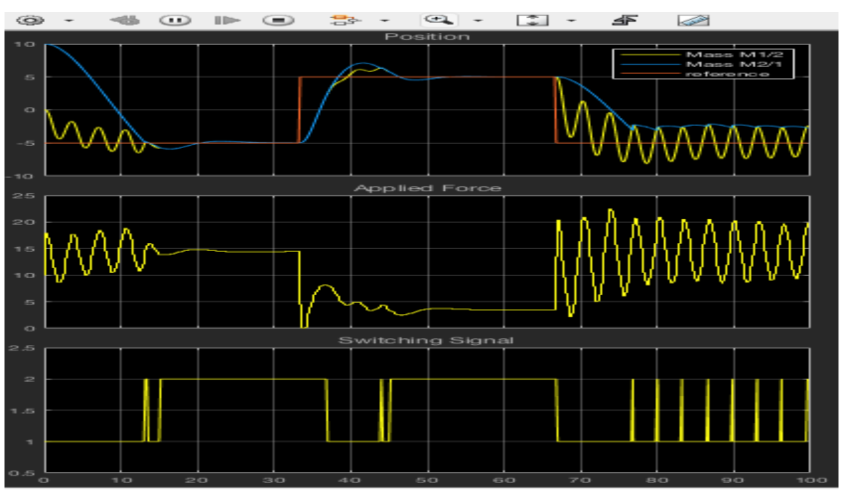

Fig. 7.2

We can see that performance of system degrades whenever two masses separates.

c) Simulation Results for Multiple MPC Controllers Block (When both controllers operating):

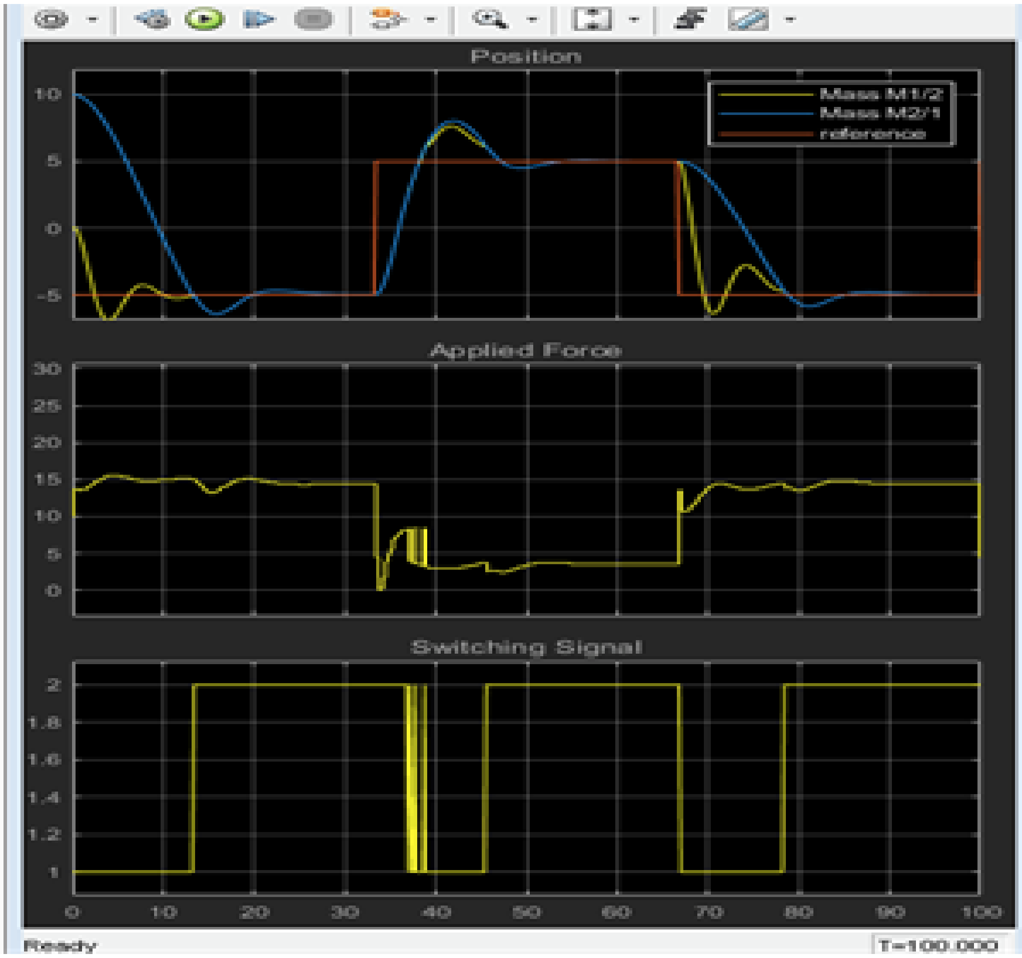

Fig. 7.3

So we can see that in this case when we are using gain scheduling and using both the controllers along with the switching signal, we are getting best output out of all the three cases. Hence gain scheduling helps in improving the response of the output for the cases where the system cannot be controlled by only just one controller.

2. <u>EXPLICIT MPC CONTROL TECHNIQUE:</u>
   In this control technique, explicit MPC controllers are created and implemented on Simulink with using two controllers together along with switching signal and outputs are generated as follows:

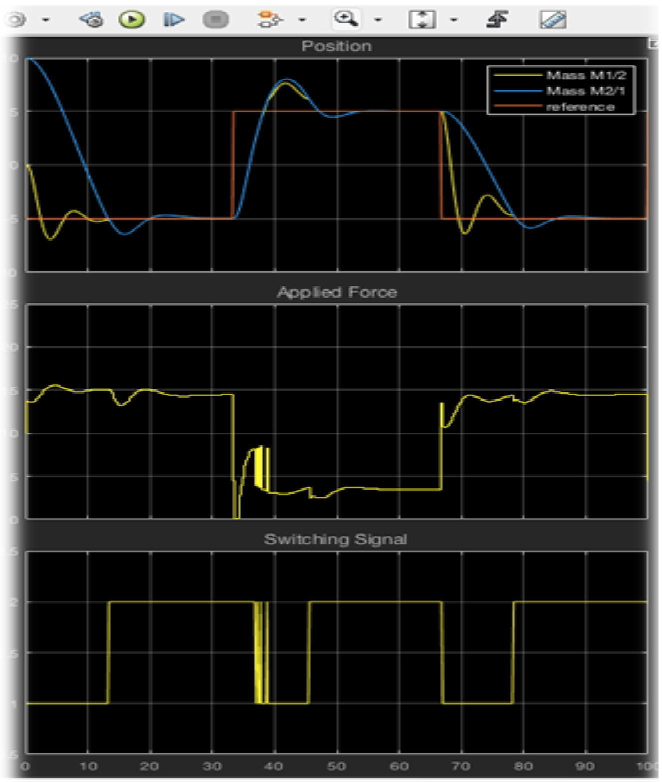

Fig. 7.4

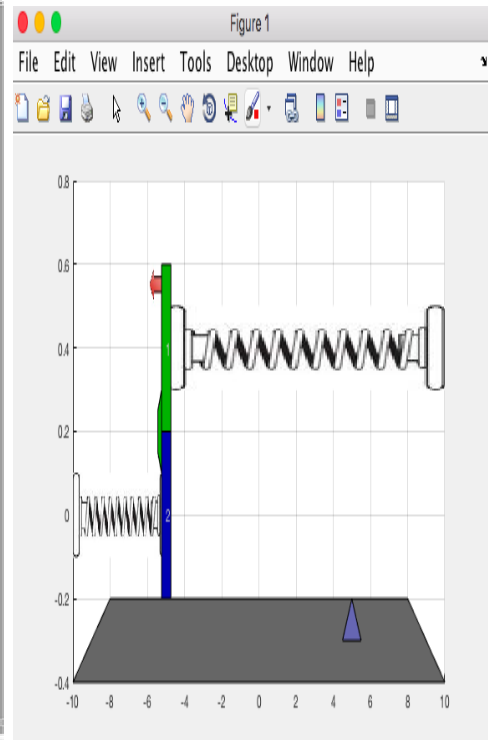

Fig. 7.5

We can see from the results that the output is very close to the reference value when the force is applied as well as when force is removed.

8 DISCUSSIONS:

The results obtained from implementing both the Techniques have been discussed here:

1. IMPLICIT MPC CONTROL TECHNIQUE:
   In this control technique, we implemented the simulation model using three cases:
   a) Using MPC1 controller only
   b) Using MPC2 controller only
   c) Using MPC1 as well as MPC2 together for the twofold dynamics along with switching signal.

   In case a) output , which is position of mass M1 is close to the reference value when the force F is in action and Implicit MPC1 is activated all the time but when Mass M1 and M2 move together there is no MPC2 activated hence there is no proper control. Hence the performance of the system is degrading whenever the two masses join.

   In case b) output which is position of Mass M1 is close to the reference value whenever the Force F is released and Implicit MPC2 is activated all the time but when Mass M1 is getting pulled by the force, MPC1 is deactivated hence the performance of the system degrades whenever the Mass M1 is pulled.

   In case c) which is the best case we have both MPC1 and MPC2 are activated with a scheduling signal and we can see from the output curve that position of mass M1 is close to the reference value for all the instances. Hence gain scheduling is giving the best results when the plant cannot be controlled by the single controller.

2. EXPLICIT MPC CONTROL TECHNIQUE:

   In the Explicit MPC control technique, MPC1 and MPC2 controllers are designed similar like implicit MPC controller design and output is generated and we can see that output which is position of mass M1 is very close to the reference value for all the instances. We can also see that the output is coming nearly as same as case c of Implicit control design.

9 CONCLUSION:

1. The gain-scheduled Explicit MPC controllers provide the same performance as gain-scheduled Implicit MPC controllers
2. For this project, we can use any of the above explained methodology to get the same results
3. Techniques used in this project can be used to improve the performance of Aircraft control system.
4. These techniques can be used to improve performance of dampers in cruise control system.

10 REFERENCES:

1. ' Model Predictive Control System Design and Implementation Using MATLAB' by luipang wang.

2. MATLAB Documentation on ' Gain Scheduled Implicit and Explicit MPC Control of Mass Spring System'