A Comparative Study of Response of Control Problems to Unconstrained & Constrained Model Predictive Control

A Dissertation

Submitted in partial fulfillment of the requiremens for the award of degree of

Master of Engineering

in

Electronic Instrumentation and Control

Submitted By:

Neha Raghu

(Roll No. 801151015)

Under the esteemed guidance of

Ms. Ruchika
Lecturer, EIED
Thapar University

ELECTRICAL AND INSTRUMENTATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

(Established under the section 3 of UGC act, 1956)

PATIALA, 147004, Punjab, India

JUNE 2013

DECLARATION

I hereby certify that the work which is being presented in the dissertation entitled **"A Comparative Study of Response of Control Problems to Unconstrained and Constrained Model Predictive Control"** in partial fulfilment of award of degree of **Master in Engineering in Electronic Instrumentation and Control** submitted in Electrical and Instrumentation Department, Thapar University, Patiala, is an authentic record of my own work carried under the supervision of **Ms. Ruchika**, Lecturer, Department of Electrical and Instrumentation Engineering, Thapar university, Patiala.

Date: 12-07-2013

Neha Raghu
(Neha Raghu)

801151015

I certify that the above statement made by the student is correct to the best of my knowledge and belief.

Date: 12.07.2013

Ruchika
(Ms. Ruchika)
Lecturer, EIED
Thapar University,
Patiala

Countersigned by

S. Ghosh
Dr. Smarajit Ghosh
Head of Department, EIED
Thapar University
Patiala, Punjab

Dr. S. K. Mohapatra
Dean of Academic Affairs
Thapar University
Patiala, Punjab

II

ABSTRACT

Model Predictive Control which was originally developed to meet the control needs of petroleum refineries and power plants can now be found in a wide variety of application areas such as chemicals, food processing, automotive and aerospace applications.

The reason for increase in use of MPC technology is its ability to handle multivariable control problems and constraints. The saturation on the control signal can make the control system performance deteriorate significantly. The strength of the constrained control framework using model predictive control lies in the optimality it achieves in a systematic manner, and the generality to cope with the multi-input, multi-output system with various constraints

This dissertation deals with the Model Predictive Control, MPC, with the goal of comparing the response of control problems to constrained and unconstrained MPC. The control problem of Van de Vusse Reactor and a temperature control problem are considered to which unconstrained and constrained MPC is applied respectively. Further, a positioning platform problem is taken and its response to both unconstrained and constrained MPC is compared. In MPC strategy, a sequence of control actions (from present time onwards) is computed that minimizes a finite duration objective function that starts from the present time and extends a fixed number of time steps into the future. Although a sequence of present and future control actions is computed, only the present control action is applied to the system. At the next time step the entire process repeats. Thus, the starting and ending time steps of the objective function shifts one step forward, thus the term receding-horizon control strategy is often associated with this strategy.

ACKNOWLEDGEMENT

With deep sense of gratitude, I express my sincere thanks to my esteemed and worthy guide, **Ms. Ruchika,** Lecturer, Department of Electrical and Instrumentation Engineering, Thapar University, Patiala, Punjab, for her invaluable guidance in carrying out the work under her effective supervision, encouragement, enlightenment and cooperation. Her feedback and editorial comments were invaluable for writing this dissertation.

This note of thanks would not be complete without mentioning my gratitude for **Dr. Smarajit Ghosh**, Head of Department of Electrical and Instrumentation Engineering, Thapar University, Patiala.

I would like to extend my gratitude to **Mr. Vikram**, Lecturer, Department of Electrical and Instrumentation Engineering, Thapar University to provide me with the right direction in carrying my dissertation work.

I am also thankful to my family and friends for providing me emotional support and blessings.

Date: 12.07.2013

Neha Raghu

(Neha Raghu)

801151015

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER – 1

INTRODUCTION

1.1 Overview

Over the last three decades predictive control strategy has attained the status of being a very successful controller design strategy, both in theory and practice. The reason for this development is that it provides high performance controllers that can easily be applied to difficult high-order and multivariable processes. Also, process constraints are handled in a simple and systematic way. However a general stability and robustness theory is lacking.

Traditional feedback controllers operate by adjusting control action in response to a change in the output set-point of a system. Model predictive control (MPC) is a technique that focuses on constructing controllers that can adjust the control action before a change in the output set-point actually occurs. This predictive ability, when combined with traditional feedback operation, enables a controller to make adjustments that are smoother and closer to the optimal control action values.

Practically all processes are subject to restrictions which can be considered as constraints on the input and output variables. These constraints are effectively handled by MPC.

1.2 Motivation

It is well known that the economic operating point of a typical process unit often lies at the intersection of constraints [1.1]. A successful industrial controller must therefore maintain the process as close as possible to constraints without violating them. In addition, process units are often complex, nonlinear, constrained multivariable systems whose dynamic behaviour changes with time. This environment led to the development, in industry, of a more general model based control methodology in which the dynamic optimization problem is solved on-line at each control execution. This new methodology for industrial process modelling and control is what we refer to as Model Predictive Control (MPC) technology.

In modern processing plants the MPC controller is a part of multi-level control hierarchy of control functions. It is often difficult to translate the control requirements at this level into an appropriate conventional control structure. In MPC methodology this combination of blocks is replaced by a single MPC controller. The hierarchy of control system functions is illustrated in fig. 1.1.

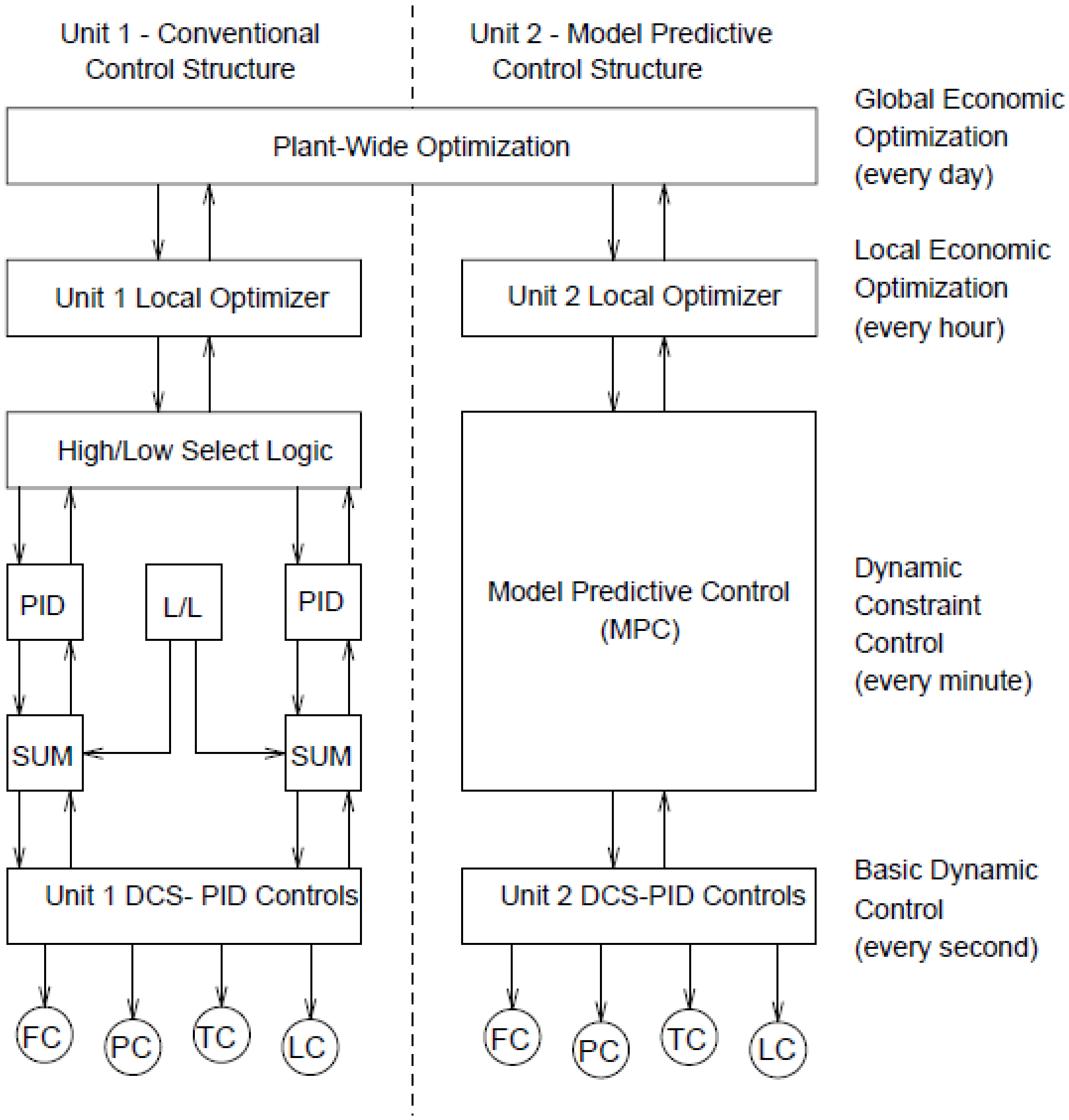

Fig. 1.1 Hierarchy of control system functions in a typical processing plant. Conventional structure is shown at the left; MPC structure is shown at the right. [1.2]

1.3 Objective and Scope of the dissertation

The objective of this dissertation can be stated in the following points:

- Study of basic control model, the unconstrained MPC. Implementation of receding horizon approach and analysis of reactor problem.
- Extension of MPC to problems containing constraints. Implementation of constrained

MPC control to temperature control problem.

- Comparison and analysis of response of a control problem to unconstrained and constrained MPC control.

1.4 Organisation of the dissertation

The dissertation is organised as follows:

Chapter 2 contains the literature review, describing the history and development of industrial MPC and stability issues of the same.

Chapter 3 is a take on MPC technology where it is compared with traditional control and the basic structure and objectives of MPC are defined. Thorough description of use of MPC in industry is given.

Chapter 4 deals with the receding horizon approach and the optimization problem wherein various types of models and objective functions are discussed.

Chapter 5 focusses on basic control model, the unconstrained MPC. Dynamic matrix control is discussed and implemented on a reactor problem.

Chapter 6 extends the MPC control to problems containing constraints and implemented on a temperature control problem.

Chapter 7 contains the problem definition for a positioning platform and the comparison of responses of unconstrained and constrained MPC control.

Chapter 8 contains the concluding remarks of the dissertation.

References

[1.1] Prett, D. M., & Gillette, R. D., *Optimization and constrained multivariable control of a catalytic cracking unit*, In Proceedings of the joint automatic control conference, 1980.

[1.2] Qin, S. J., & Badgwell, T. A., *An survey of industrial model predictive technology*, Control Engineering Practice 11, pp. 733-764, 2003.

CHAPTER 2

LITERATURE REVIEW

Model Predictive Control was originally developed for chemical applications to control the transients of dynamic systems with hundreds of inputs and outputs, subject to constraints [2.1]. It is a form of control in which current control action is computed by solving on-line, at each sampling instant, a finite horizon open-loop optimal control problem, using current state of plant as initial state. The optimization yields an optimal control sequence, the first control of which is applied to the plant.

In last two decades the MPC landscape has changed drastically. There has been a large increase in the number of reported applications and significant improvements in the technical capability.

2.1 A brief history of MPC

This section presents a brief history of industrial MPC technology. Fig. 2.1 shows an evolutionary tree for the most significant industrial MPC algorithms.

The development of modern control concept can be traced back to the work of Kalman in the early 1960's who sought to determine when a linear control system can be said to be optimal [2.2], [2.3]. A linear quadratic regulator (LQR) was designed to minimize an unconstrained quadratic objective function of states and inputs. The LQR algorithm had powerful stabilizing properties because of the infinite horizon. But it had little impact on the control technology development in the process industries. The reason for this was that there were no constraints in its formulation and the nonlinearities of the real system.

Various applications of MPC were reported in the process industry in the late 1970's. The important ones were by Richalet *et.al* which presented Model Predictive Heuristic Control (MPHC) and by Cutler and Ramaker which presented Dynamic Matrix Control [2.4], [2.5]. Model Predictive Heuristic Control was later called Model Algorithmic Control (MAC). The underlying theme of both the theories was to use a dynamic model of the process to predict the effect of future control actions, which were determined by minimizing the predicted error subject to operational restrictions. The optimization is repeated at each sampling instant with updated information from the process. These algorithms were algorithmic as well as heuristic and took advantage of the development of digital computers. The initial versions of MPC were not automatically stabilizing. However, if stable plants are considered and a horizon is chosen large enough compared with the settling time of the plant, stability is achieved.

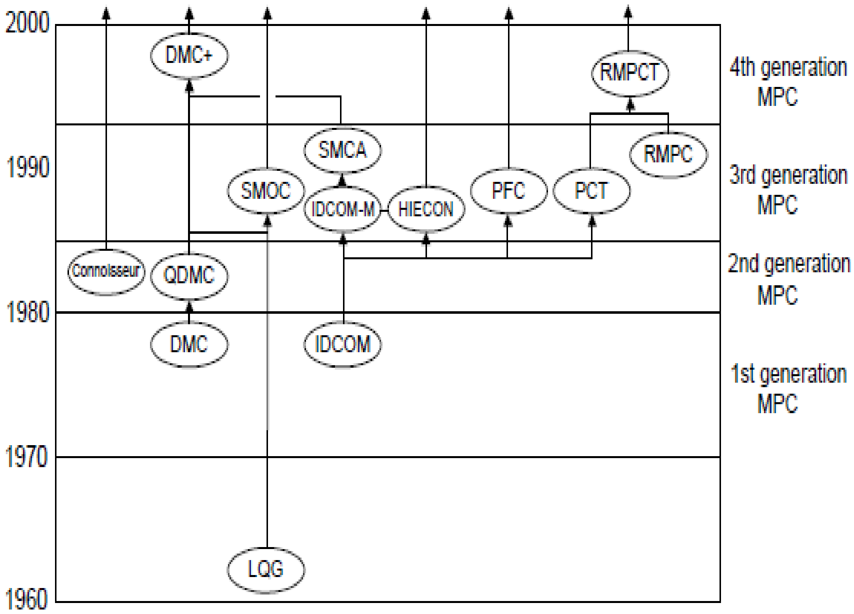

Fig. 2.1 Approximate genealogy of linear MPC algorithms [2.1]

The solution software to MPHC was referred to as IDCOM, an acronym for Identification and Command [2.6]. In this approach, impulse response model and a quadratic objective function over a finite prediction horizon was used. Optimal inputs were computed using a heuristic iterative algorithm, interpreted as the dual of identification.

DMC was presented by Cutler and Ramaker at the 1979 AlChE meeting [2.7], and at the 1980 Joint Automatic Control Conference [2.5]. In this approach, linear step response model for the plant and quadratic objective function over a finite horizon was used. The optimal inputs were computed as the solution to a least squares problem.

Prett and Gillette described an application of DMC to FCCU reactor/ regenerator in which the algorithm was modified to handle nonlinearities and constraints [2.8].

The initial IDCOM and DMC algorithms represent the *first generation* of MPC technology.

Later on a *second generation* of MPC such as quadratic dynamic matrix control (QDMC) came into picture.

Cutler *et.al*, first described the QDMC algorithm in a 1983 AlChE conference paper [2.9]. Garcia and Morshedi presented a more comprehensive description some years later [2.10]. QDMC approach used quadratic programming to solve the constrained open-loop optimal control problem where the system is linear, objective function is quadratic and the constraints are defined by linear inequalities. Although QDMC algorithm provided a systematic approach to incorporate input and output constraints, but there was no clear way to handle an infeasible solution.

To tackle this issue, engineers at Shell, Adersa, Setpoint, Inc. developed new versions of MPC algorithms. IDCOM-M controller was first described in a paper by Grosdidier, Froisy, and Hammann in 1988 [2.11]. Another paper presented by Froisy and Matsko in 1990 described an application of IDCOM-M to a Shell fundamental control problem [2.12]. The main difference from the previous algorithms is the use of two separate objective functions, one for the outputs and then, if there are extra degrees of freedom, one for the inputs. A quadratic output objective function is minimized first subject to input constraints. Each output is driven as closely as possible to a desired value at a single point in time known as the coincidence point. The desired output value comes from a first order reference trajectory that starts at the current measured value and leads smoothly to the set-point.

In the late 1980's engineers at Shell Research in France developed the Shell Multivariable Optimizing Controller (SMOC) [2.13]. It was described as a bridge between state-space and MPC algorithms. This approach uses state space models so that full range of linear dynamics can be represented. A kalman filter is used to estimate the plant states and unmeasured disturbances from output measurements. A distinction is introduced between controlled variables appearing in the control objective and feedback variables that are used for state estimation.

The IDCOM-M and SMOC represent the *third generation* of MPC technology. This generation distinguishes between several levels of constraints and provides some mechanism to recover from infeasible solution. There were other algorithms belonging to this generation such as hierarchical constraint control (HIECON), PCT algorithm sold by Profimatics and RMPC algorithm sold by Honeywell.

In the late 1995, RMPC algorithm offered by Honeywell was merged with the Profimatics PCT controller to create RMPCT, robust model predictive control technology. In 1998, SMCA and DMC technologies were merged to create DMC-plus. RMPCT and DMC-plus are representative of *fourth generation* of MPC technology. This generation is sold today with features such as, windows based graphical user interface, multiple optimization levels to address prioritized control objectives, improved identification technology based on prediction error method.

2.2 Stability

For those working with predictive control, stability has always been an important issue. Due to the finite horizon, stability is not guaranteed and is achieved by tuning the weights and horizons. Mohtadi proved specific stability theorems of GPC using state-space relationships and studied the influence of filter polynomials on robustness improvement [2.14]. However a general stability property for predictive controllers, in general, with finite horizons was still lacking. This led researchers to pursue new predictive control methods with guaranteed stability in the 1990s. With that purpose a number of design modifications have been proposed since then including the use of terminal constraints by Kwon *et.al*, the introduction of dual-mode designs by Mayne and Michalska and the use of infinite prediction horizons by Rawlings and Muske, among others [2.15], [2.16], [2.17]. Clarke and Scattolini and Mosca *et.al*, independently developed stable predictive controllers by imposing end-point equality constraints on the output after a finite horizon [2.18], [2.19]. Kouvaritakis *et.al*, presented a stable formulation for GPC by stabilizing the process prior to the minimization of the objective function [2.20]. Many of these techniques are specialized for state-space representations of the controlled plant, and achieve stability at the expense of introducing additional constraints and modifying the structure of the design.

References

[2.1] Qin, S. J., & Badgwell, T. A., *A survey of industrial model predictive control technology*, Control Engineering Practice, 11, pp. 733–764, 2003.

[2.2] Kalman, R.E., *Contribution to the theory of optimal control*, Bull. Soc. Math. Mex. 5: pp. 102-119, 1960.

[2.3] Kalman, R.E, *A new approach to linear filtering and prediction problems*, Trans. ASME, J. Basic Engineering, pp. 35-45, 1960.

[2.4] Richalet, J., Rault, A., Testud, J. L., & Papon, J., *Algorithmic control of industrial processes*, In Proceedings of the 4th IFAC symposium on identification and system parameter estimation, pp. 1119–1167, 1976.

[2.5] Cutler, C. R., Ramaker, B.C., *Dynamic Matrix Control - A Computer Control Algorithm*, Automatic Control Conference, San Francisco, CA, 1980.

[2.6] Richalet, J., Rault, A., Testud, J.L., Papon, J., *Model Predictive Heuristic Control: Application to Industrial Processes*, Automatica, 14, 2, pp. 413-428, 1978.

[2.7] Cutler, C. R., & Ramaker, B. L., *Dynamic matrix control—a computer control algorithm,* AICHE national meeting, Houston, TX, April 1979.

[2.8] Prett, D. M., & Gillette, R. D., *Optimization and constrained multivariable control of a catalytic cracking unit*, In Proceedings of the joint automatic control conference, 1980.

[2.9] Cutler, C., Morshedi, A., & Haydel, J., *An industrial perspective on advanced control*, In AICHE annual meeting, Washington, DC, October, 1983.

[2.10] Garcia, C. E., & Morshedi, A. M., *Quadratic programming solution of dynamic matrix control (QDMC),* Chemical Engineering Communications, 46, pp. 73–87, 1986.

[2.11] Grosdidier, P., Froisy, B., & Hammann, M., *The IDCOM-M controller,* In T. J. McAvoy, Y. Arkun, & E. Zafiriou (Eds.), Proceedings of the 1988 IFAC workshop on model based process control, pp. 31–36, Oxford: Pergamon Press, 1988.

[2.12] Froisy, J. B., & Matsko, T., *IDCOM-M application to the Shell fundamental control problem*, AICHE annual meeting, November 1990.

[2.13] Marquis, P., & Broustail, J. P., SMOC, *A bridge between state space and model predictive controllers: Application to the automation of a hydrotreating unit,* In T. J. McAvoy, Y. Arkun, & E. Zafiriou (Eds.), Proceedings of the 1988 IFAC workshop on model based process control, pp. 37–43, Oxford: Pergamon Press, 1988.

[2.14] Clarke, D.W., Mohtadi, C., Tuffs, P.S., *Generalized Predictive Control - Part II. Extensions and Interpretations*, Automatica, 23, 2, pp. 149-160, 1987.

[2.15] Kwon, W.H., Pearson, A.E., *A Modified Quadratic Cost Problem and Feedback Stabilization of a Linear System*, IEEE Transactions on Automatic Control, AC-22, 5, pp. 838-842, 1977.

[2.16] Mayne, D.Q., Michalska, H., *Robust Receding Horizon Control of Constrained Nonlinear Systems*, IEEE Transactions on Automatic Control, 38, pp. 1623-1633, 1993.

[2.17] Muske, K.R., Rawlings, J., *Model Predictive Control with Linear Models*, AIChE Journal, 39, pp. 262-287, 1993.

[2.18] Clarke, D.W., Scattolini, R., *Constrained Receding-horizon Predictive Control,* Proceedings IEE, 138, 4, pp. 347-354, 1991.

[2.19] Mosca, E., Lemos, J.M., Zhang, J., *Stabilizing I/O Receding Horizon Control*, IEEE Conference on Decision and Control, 1990.

[2.20] Kouvaritakis, B., Rossiter, J.A., Chang, A.O.T., *Stable Generalized Predictive Control: An Algorithm with Guaranteed Stability*, Proceedings IEE, Part D, 139, 4, pp. 349-362, 1992.

CHAPTER 3

MPC TECHNOLOGY

3.1 Introduction

Predictive control is a class of control strategies based on the explicit use of a process model to generate the predicted values of the output at future time instants, which are then used to compute a sequence of control moves that optimize the future behaviour of a plant. Predictive control is rather a methodology than a single technique. The difference in the various methods is mainly the way the problem is translated into a mathematical formulation.

The development of modern control concepts can be traced back to the work of Kalman in the early 1960s with the linear quadratic regulator (LQR) designed to minimize an unconstrained quadratic objective function of states and inputs. The infinite horizon enabled the LQR algorithm to have powerful stabilizing properties. However it had little impact on the control technology development in the process industries. The reason for this was the absence of constraints in its formulation, the nonlinearities of the real systems, and above all the culture of the industrial process control community at the time, in which instrument technicians and control engineers either had no exposure to optimal control concepts or regarded them as impractical. Thus the early proponents of MPC for process control proceeded independently, addressing the needs of the industry.

In the late 1970s various articles reported successful applications of model predictive control in the industry, principally the ones by Richalet *et al.* (1978) presenting Model Predictive Heuristic Control (later known as Model Algorithmic Control (MAC)) [3.1] and those of Cutler and Ramaker (1980) with Dynamic Matrix Control (DMC) [3.2]. The common theme of these strategies was the idea of using a dynamic model of the process (impulse response in the former and step response in the later) to predict the effect of the future control actions, which were determined by minimizing the predicted error subject to operational restrictions. The optimization is repeated at each sampling period with updated information from the process. These formulations were algorithmic and also heuristic and took advantage of the increasing potential of digital computers at the time. Stability was not addressed theoretically and the initial versions of MPC were not automatically stabilizing. However, by focusing on stable plants and choosing a horizon large enough compared to the settling time of the plant, stability is achieved after playing with the weights of the cost function.

Later on a second generation of MPC such as quadratic dynamic matrix control (QDMC; Garcia, Morshedi, 1986) used quadratic programming to solve the constrained open-loop optimal control problem where the system is linear, the cost quadratic, the control and state constraints are defined by linear inequalities [3.3].

Another line of work arose independently around adaptive control ideas developing strategies essentially for mono-variable processes formulated with transfer function models (for which less parameter are required in the identification of the model). The first initiative came from Astron *et al*. (1970) with the Minimum Variance Control where the performance index to be minimized is a quadratic function of the error between the most recent output and the reference (i.e. the prediction horizon N=1). In order to deal with non-minimum phase plants a penalized input was placed in the objective function and this became the Generalized Minimum Variance (GVM) control. To overcome the limitation on the horizon, Peterka (1984) developed the Predictor-Based Self-Tuning Control [3.4]. Extended Prediction Self-Adaptive Control (EPSAC) by De Keyser *et al*. (1985) proposes a constant control signal starting from the present moment while using a sub-optimal predictor [3.5]. Later on the input was replaced by the increment in the control signal to guarantee a zero steady-state error.

Based on the ideas of GVM Clarke *et al*. (1987) developed the Generalized Predictive Control (GPC) and is today one of the most popular methods [3.6]. A closed form for the GPC is given by Soeterboek. State-space versions of unconstrained GPC were also developed.

3.2 Traditional control and MPC

Traditional feedback controllers operate by adjusting control action in response to a change in the output set-point of a system. Model predictive control (MPC) is a technique that focuses on constructing controllers that can adjust the control action before a change in the output set-point actually occurs. This predictive ability, when combined with traditional feedback operation, enables a controller to make adjustments that are smoother and closer to the optimal control action values. MPC is often used in addition to traditional control like PID. In large plants MPC is not a replacement for traditional PID, but used in addition to PID controllers. PID controllers are used as single-loop controllers, while MPC is used as an overall system. PID handles only a single input and a single output (SISO systems), while MPC is a more advanced method of process control used for MIMO systems (Multiple Inputs, Multiple Outputs).

Fig. 3.1 shows that set-point in traditional control is far from constraint whereas in MPC, set-point is closer to constraint. MPC tries to bring the process as close as possible to the constraints without violating them. MPC is an improved process operation in comparison to traditional control being a non-optimal process operation.

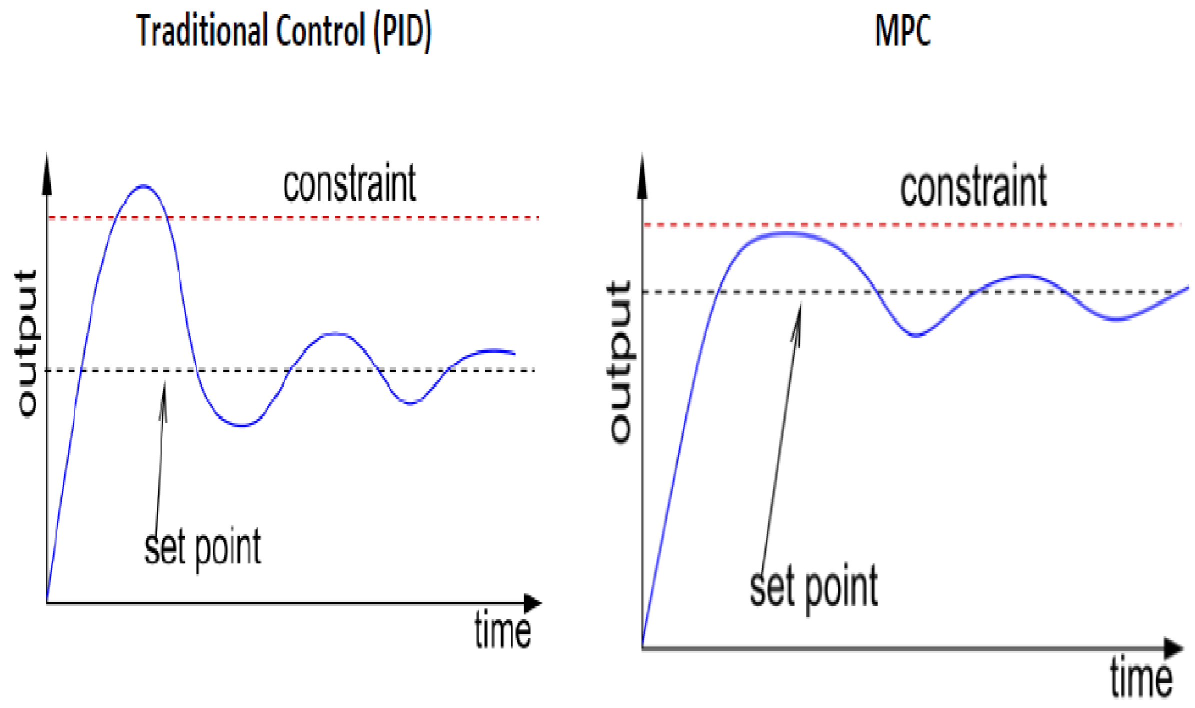

Fig. 3.1 Comparison of traditional control and MPC response [3.7]

The explicit use of a model is the main difference between predictive control and the classical PID controller. Its advantage is that the behaviour of the controller can be studied in detail, simulations can be done and performance can be evaluated. One of the drawbacks is the need of an appropriate model of the process. The benefits obtained are affected by the discrepancies existing between the real process and the model. Another drawback is that although the resulting control law is easy to implement and requires little computation, its derivation is more complex than that of the PID.

3.3 General Objectives of MPC

The general objectives of MPC controller, in order of importance, are following:

- Prevent violation of input and output constraints.
- Drive the control variables to their steady-state optimal values.
- Drive the manipulated variables to their steady-state optimal values using remaining degrees of freedom.
- Prevent excessive movement of manipulated variables.
- When signals and actuators fail, control as much plant as possible.

The translation of these objectives into mathematical problem statement involves a number of approximations and trade-offs that defines the basic character of the controller. Like any design problem, there are a number of possible solutions; there are a number of different MPC control formulations.

3.4 Structure of MPC

A model of the process is used to predict the future evolution of the process to optimize the control signal. Unlike time delay compensation methods, the predictions are made for more than one time delay ahead. Fig. 3.2 shows the basic structure of MPC controller.

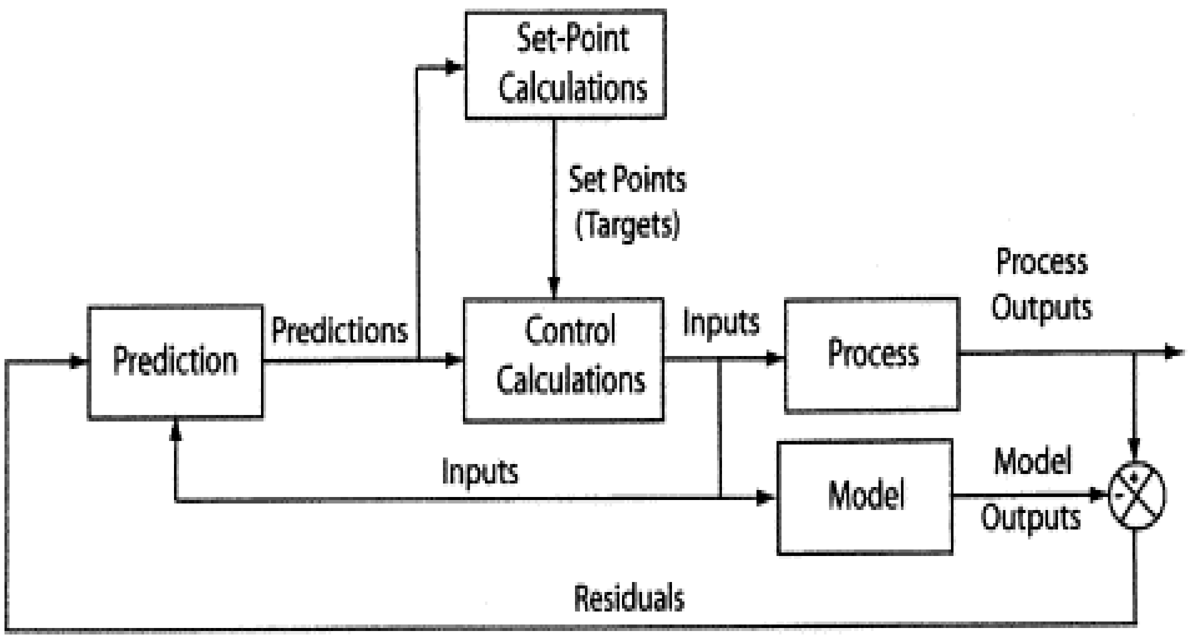

Fig. 3.2 Basic structure of MPC controller [3.8]

The MPC calculations are imbedded in the prediction and controller blocks and are carried out quite often (e.g., every 1–10 min). The prediction block predicts the future trajectory of all controlled variables, and the controller achieves the desired response while keeping the process within limits. The targets for the MPC calculations are generated by solving a steady-state optimization problem based on a linear process model, which also finds the best path to achieve new targets. These calculations may be performed as often as MPC calculations [3.9].

3.5 MPC in industry

Model predictive control was primarily developed for chemical applications, to control the transients of dynamic systems with hundreds of inputs and outputs, subject to constraints [3.10]. Such applications are characterized by slow dynamics, presence of expert human supervision, and expensive plants, causing the cost of the control system to be practically irrelevant. Also, each controller is usually deployed into a single plant, which means that each control design is customized and specifically tuned.

In the last 15 years, due to the increase of the available computational power in microprocessors, MPC has been investigated in different industries, such as automotive, aerospace, and factory automation/robotics, where each control design is deployed in hundreds, or even thousands, of final product. These domains are referred to as “large volumes” application domains [3.11]. The applications in large volumes domains differ from the ones in chemical process control for system size, significantly smaller, dynamics, significantly faster, control system supervision, unsupervised, and plant cost, significantly reduced, meaning that the control system has to have low cost. Another major difference is that, due to the large number of plants where it is deployed, the controller cannot be tuned for a single plant, and hence needs to be flexible and robust to accommodate the plant-to-plant differences. While the availability of cheap powerful processors and the development of new MPC algorithms address some of the problems are yet to be solved and limit still today the widespread use of MPC in favour of the traditional control strategies, like PIDs. Some of these problems are the computational complexity of the algorithm and the induced microprocessor cost, the need of robust calibration, the applicability to limited classes of dynamics, the lack for guaranteed robustness and bounded computational load. On the other hand there are multiple reasons for the large volumes industries to look with increased interest at MPC, such as the performance optimization, the enforcement of constraints on outputs and inputs, the ease of design for multivariable systems, the capability of dealing with time delays and future information, and the capability of easily handling certain nonlinearities.

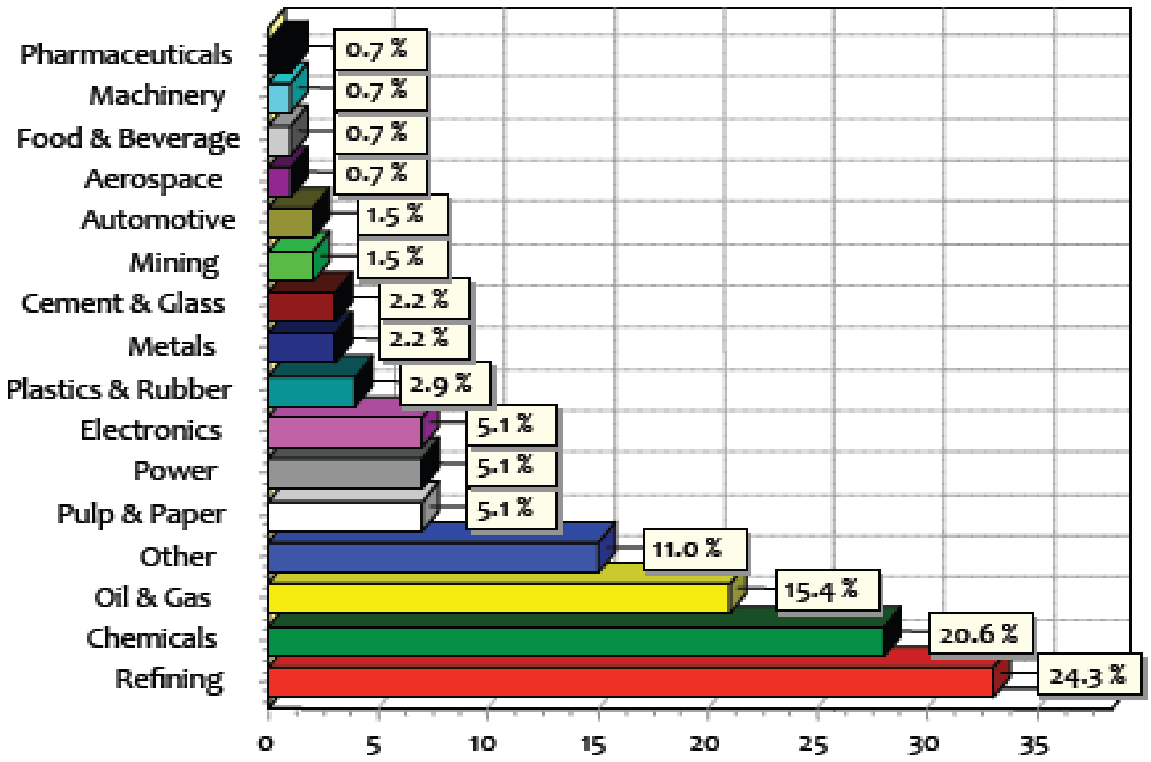

Fig. 3.3 Industrial areas of respondents to the survey.

The results of a survey (November 7, 2005) about the use of MPC techniques/real-time optimization in a set of US industries are illustrated in the figures 3.3 - 3.5 (courtesy- ARC advisory group).

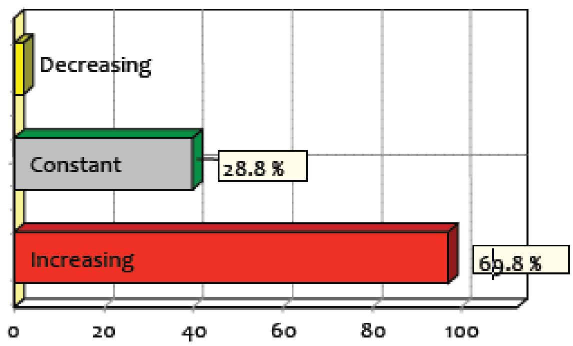

Fig. 3.4 Use of MPC accelerating, remaining constant or declining?

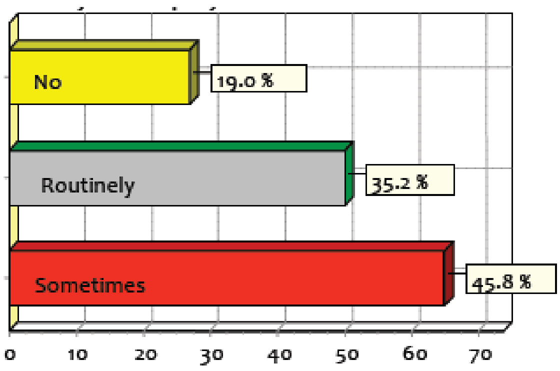

Fig. 3.5 Does the respondent's company use MPC?

In the year 2006, it was estimated that model predictive control was used to control more than 4500 chemical processes. Fig. 3.6 shows the breakdown of MPC in chemical industry.

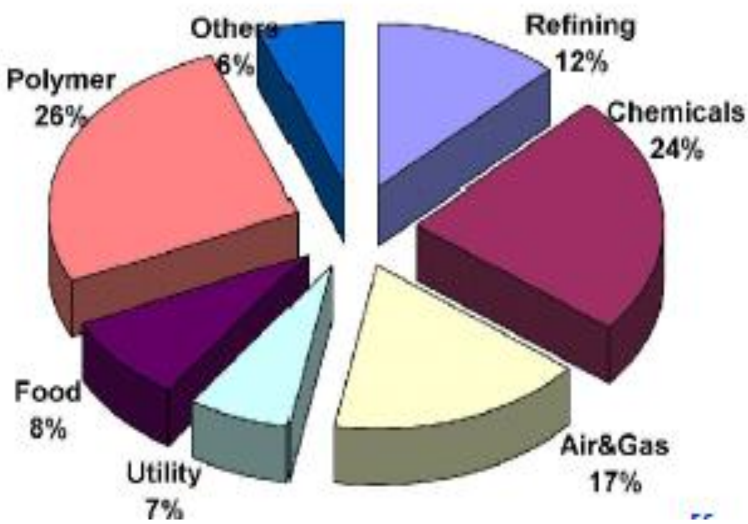

Fig. 3.6 Breakdown of MPC in chemical industry (Nagy, IJRNC, 2006)

References

[3.1] Richalet, J., Rault, A., Testud, J.L., Papon, J., *Model Predictive Heuristic Control: Application to Industrial Processes*, Automatica, 14, 2, pp. 413-428 (1978).

[3.2] Cutler, C.R., Ramaker, B.C., *Dynamic Matrix Control - A Computer Control Algorithm*, Automatic Control Conference, San Francisco, CA (1980).

[3.3] Garcia, C.E., Prett, D.M., Morari, M., *Model Predictive Control: Theory and Practice, A Survey*, Automatica, 25, 3, pp. 335-348 (1989).

[3.4] Peterka, V., *Predictor-based Self-tuning Control*, Automatica, 20, 1, pp. 39-50 (1984).

[3.5] De Keyser, R.M.C., Van Cuawenberghe, A.R., *Extended Prediction Self-adaptive Control*, IFAC: Symposium on Identification and System Parameter Estimation, York, UK, pp. 1317-1322 (1985).

[3.6] Clarke, D.W., Mohtadi, C., Tuffs, P.S., *Generalized Predictive Control - Part II. Extensions and Interpretations*, Automatica, 23, 2, pp. 149-160 (1987).

[3.7] Hans-Petter Halvorsen, *Model Predictive Control in labVIEW*, Department of Electrical Engineering, Information Technology and Cybernetics, Telemark University College, 2011.

[3.8] Seborg, D.E., Edgar, T.F., Mellichamp, D.A., *Process Dynamics and Control*, 2nd Edition, John Wiley and Sons, Inc., New Jersey, New York (2004).

[3.9] Backx, *Reduction and Predictive Control Design for a computational Fluid Dynamic model*, Vol.3, Proceedings of the 41st IEEE conference on Decision and Control (2002).

[3.10] Qin, S.J., Badgwell, T.A., *A survey of industrial model predictive control technology,* Control Engineering Practice 11, pp. 733–764 (2003).

[3.11] Stefano di Cairano, *An industry perspective on MPC in large volumes applications: Potential benefits and open challenges*, Mechatronics, Mitsubishi electric research laboratories, Cambridge, MA, USA

CHAPTER 4

THE RECEDING HORIZON

4.1 Basic Description

Future values of output variables for more than one time delay ahead are predicted using a dynamic model of process and current measurements. Control calculations are based on both future predictions as well as current measurements. The manipulated variables, u(k), at the kth sampling instant are calculated so that they minimize an objective function, ϕ. Measured disturbances are included in the control calculations. The calculated manipulated variables are implemented as set-point for lower level control loops.

MPC is based on an iterative and finite horizon optimization of a plant model. At any time 't', the current plant state is sampled and a cost minimization algorithm is applied for a relatively short time horizon in the future:[t, t+T]. Only the first step of the control algorithm is implemented and then the plant state is sampled again. The calculations are again repeated starting from the now current state thereby yielding a new control and a new predicted state path. The prediction horizon keeps being shifted forward and for this reason model predictive control is also called **receding horizon control** [4.1].

4.2 Prediction and control horizon

PREDICTION HORIZON: The number of samples in future the MPC controller predicts the plant output [4.2]. Prediction (P) is typically as far ahead as two to three times the dominant time constant of the system. Suppose the process is sampled at say one twentieth of that time constant: the output prediction horizon could then be up to some 60 steps ahead [4.3].

CONTROL HORIZON: The number of samples within the prediction horizon where the MPC controller can affect the control action [4.2]. The control horizon is always shorter than the prediction horizon.

Fig. 4.1 shows the prediction and control horizons. For time k, MPC controller predicts the plant output, $k + N_p$. The figure shows that control action does not change after the control horizon ends.

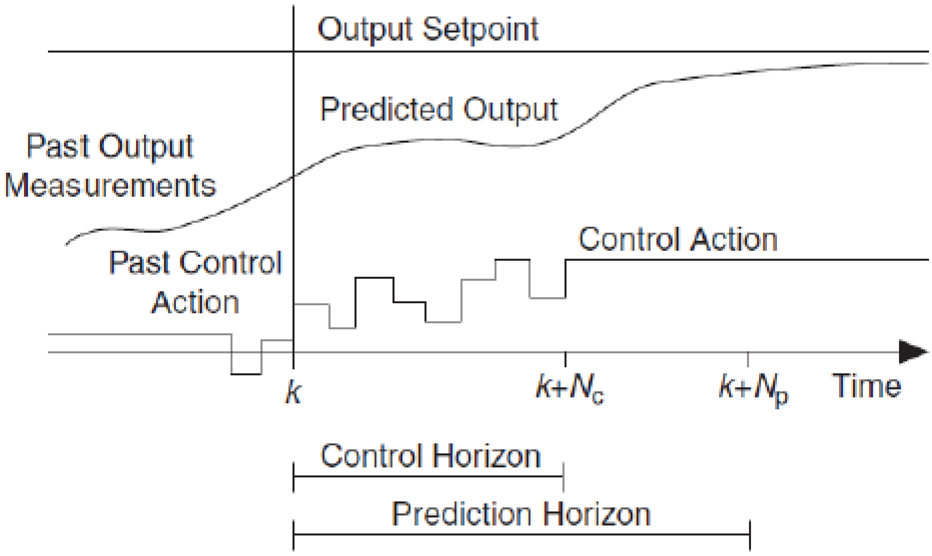

Fig. 4.1 Prediction and Control horizons [4.2]

4.3 Receding horizon approach

In the fig. 4.2, control algorithm for single-input, single-output (SISO) plant is assumed. The current time is labelled as time step k. The plant output at current time is y(k).

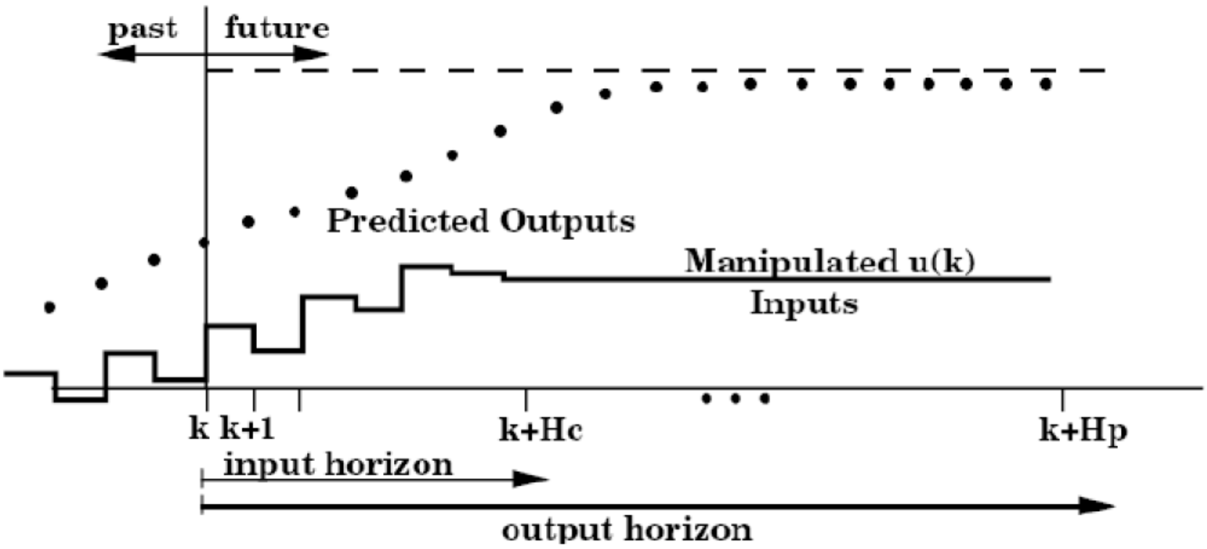

Fig. 4.2 Receding horizon control [4.4]

The figure shows the previous history of output trajectory. A set-point trajectory is shown which is the trajectory that the output should follow. The value of set-point trajectory at any time 't' is denoted by s(t). Also a reference trajectory is shown which starts at the current output y(k). It defines an ideal trajectory along which the plant should return to the set-point trajectory, for example after a disturbance occurs. It is not necessary for the plant to go back to the set-point trajectory as fast as possible. Usually the reference trajectory approaches the set-point trajectory exponentially. At the kth sampling instant, the values of the manipulated variables, u, at the next M sampling instants, {u(k), u(k+1), …, u(k+M -1)} are calculated. This set of M "control moves" is calculated so as to minimize the predicted deviations from the reference trajectory over the next P sampling instants while satisfying the constraints. Typically, an LP or QP problem is solved at each sampling instant. Then the first "control move", u(k), is implemented. At the next sampling instant, k+1, the M-step control policy is re-calculated for the next M sampling instants, k+1 to k+M, and implement the first control move, u(k+1). Then Steps 1 and 2 are repeated for subsequent sampling instants [4.5].

4.4 Optimization Problem

Here possible objective functions are discussed first, and then the possible process models that can be used for MPC are discussed.

4.4.1 Objective functions

There are several different choices for objectives functions. The first one is a standard *least-squares* or "quadratic" objective function [4.6]. The objective function is a "sum of squares" of the predicted errors (differences between the set points and model-predicted outputs) and the control moves (changes in control action from step to step).

For a prediction horizon of P and a control horizon of M, the least-squares objective function is written as

$$\Phi= \sum_{i=1}^{P}(r_{k+i} - \hat{y}_{k+i})^2 + w \sum_{i=0}^{M-1} \Delta u^2_{k+i} \tag{4.1}$$

where $\hat{y}$ represents the model predicted output, r is the set point, Δu is the change in manipulated input from one sample to the next ,w is a weight for the changes in the manipulated input, and the subscripts indicate the sample time (k is the current sample time).

Another possible objective function is to simply take a sum of the absolute values of the predicted errors and control moves [4.6].

For a prediction horizon of P and a control horizon of M, the absolute value objective function is

$$\Phi = \sum_{i=1}^{P} |r_{k+I} - \hat{y}_{k+i}| + w\sum_{i=0}^{M-1} |\Delta u_{k+i}| \qquad (4.2)$$

The optimization problem solved is usually stated as a minimization of the objective function, obtained by adjusting the M control moves, subject to modelling equations, and constraints on the inputs and outputs.

Least-squares formulations are by far the most common objective functions in MPC. Least squares yields analytical solutions for unconstrained problems and penalizes larger errors (relatively) more than smaller errors. The absolute value objective function has been used in few algorithms because a linear programming problem results. LPs are solved in large-scale scheduling and allocation problems. The LP approach is not useful for model predictive control, because the manipulated variable moves often “hop” from one constraint to another.

4.4.2 Models

Many different types of models are possible for calculating the predicted values of the process outputs, which are used in evaluating the objective function. Since the outputs are evaluated at discrete-time steps, it makes sense to use discrete models for the output prediction.

4.4.2.1 Finite Step Response

FSR models are obtained by making a unit step input change to a process operating at steady-state. The model coefficients are simply the output values at each time step. Here, s_i represents the step response coefficients for the ith sample time after the unit step input change. If a non-unit step change is made, the output is scaled accordingly.

The step response model is the vector of step response coefficients,

$$S = [s_1\ s_2\ s_3\ s_4\ s_5\ .\ .\ .\ s_N]^T \qquad (4.3)$$

where the model length N is long enough so that the coefficients values are relatively constant (i.e. the process is close to a new steady state).

4.4.2.2 FINITE IMPULSE RESPONSE

Another common form of model is a finite impulse (FIR). Here, a unit pulse is applied to the manipulated input, and the model coefficients are simply the values of the outputs the ith impulse response coefficients. h_i represents the ith impulse response coefficient. There is a direct relationship between step and impulse response models:

$$h_i = s_i - s_{i-1} \tag{4.4}$$

$$s_i = \sum_{j=1}^{i} h_j \tag{4.5}$$

The impulse response coefficients are simply the changes in the step response coefficient at each time step. Similarly, step response coefficient is the sum of the impulse response coefficients to that point. It should be noted that there are two major limitations to step and impulse response models. They can only be used to represent open-loop stable processes, and they require a large number of parameters (model coefficients) compared to state space and transfer function models.

4.4.2.3 State Space Model

The state space representation is a time domain approach which provides a convenient and compact way for model representation and analysis of systems with multiple inputs and outputs.

A very general state space model of a linear system with *p* inputs, *q* outputs and *n* state variables can be written in the following form

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{4.6a}$$

$$y(t) = Cx(t) + Du(t) \tag{4.6b}$$

where,

$x(\cdot)$ is the "state vector", $x(t) \in R^n$;

$y(\cdot)$ is the "output vector", $y(t) \in R^q$;

$u(\cdot)$ is the "input or control vector", $u(t) \in \mathbb{R}^p$;

$A(\cdot)$ is the "state matrix", $\dim[A(\cdot)] = n \times n$,

$B(\cdot)$ is the "input matrix", $\dim[B(\cdot)] = n \times p$,

$C(\cdot)$ is the "output matrix", $\dim[C(\cdot)] = q \times n$,

$D(\cdot)$ is the "direct-feed through matrix", $\dim[D(\cdot)] = q \times p$,

$\dot{x}(t) = \frac{d}{dt}x(t)$

Equation 4.6 is a generalized formula. Here all matrices are time-variant in nature (i.e., their elements can depend on time). However, in the common LTI case, matrices are time invariant. The time variable *t* can be continuous or discrete. In the latter case, the time variable is usually denoted as *k*. Hybrid systems allow for time domains which have both discrete and continuous parts.

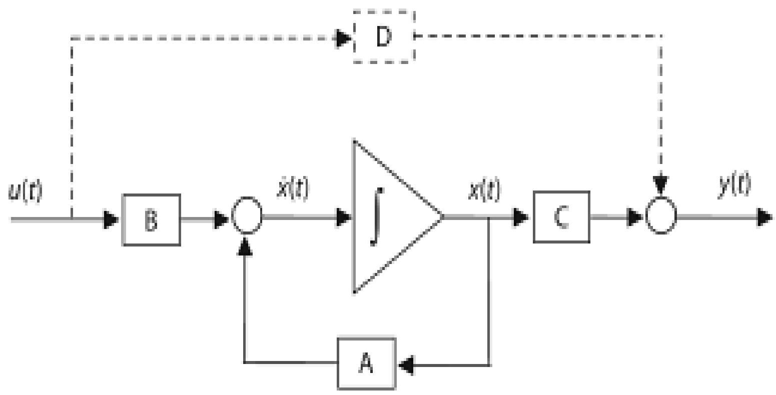

Fig. 4.3 Block diagram of state space model [4.7].

4.4.2.4 Transfer function model

For some processes, good models can be obtained on the basis of physical laws or by parametric system identification. In this case, a transfer function model is preferred. Less parameters are used than in the Impulse/Step response models.

References

[4.1] Patcharaprakiti, N., Thongpron, J., Kirtikara, K., Chenvidhya, D., & Sangswang, A., *Model Predictive Control based on identification of photovoltaic grid inverte*r, International Journal of Information and Electronics Engineering, Vol. 2, No. 4, July 2012.

[4.2] Hans-Petter Halvorsen, *Model Predictive Control in labVIEW*, Department of Electrical Engineering, Information Technology and Cybernetics, Telemark University College, 2011.

[4.3] Jonathan Love, *Process Automation Handbook, A guide to Theory and Practice,* Springer, 2007.

[4.4] José Alberto Romagnoli, Ahmet Palazoglu, *Introduction to process control*, CRC Press, 2005.

[4.5] Bemporad, A., *Model Predictive Control: Basic concepts*, Controllo di Processo e Sistemi di Produzione, 2009.

[4.6] Bequette, B. W., *Process Control* - *Modeling, Design, and Simulation,* PHI, pp. 488-509, 2003.

[4.7] Johnson, M. A., & Moradi, M. H., *PID: New implementation and design methods,* Springer, pp. 51, 2005.

CHAPTER 5

UNCONSTRAINED MPC

This section focusses on the basic control model, the unconstrained MPC. The goal of the controller is to make the difference between output and reference as small as possible. This can be done by using a least squares problem.

5.1 Dynamic Matrix Control

DMC was developed by Shell Oil Company in the 1960s and 1970s with initial application in 1973 [5.1]. Key features of DMC include

- linear step response model for the plant
- quadratic performance objective over a finite prediction horizon
- future plant output behaviour specified by trying to follow the set point as closely as possible
- optimal inputs computed as the solution to a least-squares problem

Despite the fact that most real processes are approximately linear within only a limited operating window, linear MPC approaches are used in the majority of applications with the feedback mechanism of the MPC compensating for prediction errors due to structural mismatch between the model and the process. In model predictive controllers that consist only of linear models, the superposition principle of linear algebra enables the effect of changes in multiple independent variables to be added together to predict the response of the dependent variables. This simplifies the control problem to a series of direct matrix algebra calculations that are fast and robust. Hence it is called Dynamic Matrix Control.

DMC is based on a step response model [5.2], which has the form

$$\hat{y}_k = s_1 \Delta u_{k-1} + s_2 \Delta u_{k-2} + \ldots + s_{N-1} \Delta u_{k-N-1} + s_N u_{k-N} \tag{5.1}$$

which is written in the form

$$\hat{y}_k = \sum_{i=1}^{N-1} s_i \Delta u_{k-i} + s_N u_{k-N} \tag{5.2}$$

where $\hat{y}_k$ is the model prediction at time step *k*, and u_{k-N} is the manipulated input N steps in the past.

The model predictive output is unlikely to be equal to the actual measured output at time step *k*. the difference between the measured output (y_k) and the model prediction is called the additive disturbance.

$$d_k = y_k - \hat{y}_k \tag{5.3}$$

The "corrected prediction" is then equal to the actual measured output at step k,

$$\hat{y}_k^c = \hat{y}_k + d_k \tag{5.4}$$

Similarly, the corrected predicted output at the first time step in the future can be found

$$\hat{y}_k^c = \sum_{i=1}^{N-1} s_i \, \Delta u_{k-i+1} + s_N \, u_{k-N+1} + \hat{d}_{k+1} \tag{5.5}$$

$$\hat{y}_k^c = s_1 u_k + \sum_{i=2}^{N-1} s_i \, \Delta u_{k-i+1} + s_N \, u_{k-N+1} + \hat{d}_{k+1} \tag{5.6}$$

So, for the *j*th step into the future, we find

$$\hat{y}_{k+j}^c = \hat{y}_{k+j} + \hat{d}_{k+j}$$

$$\hat{y}_{k+j}^c = \underbrace{\sum_{i=1}^{j} s_i \, \Delta u_{k-i+j}}_{\text{Effect of future control moves}} + \underbrace{\sum_{i=j+1}^{N-1} s_i \, \Delta u_{k-i+j} + s_N \, u_{k-N+j}}_{\text{effect of past control moves}} + \underbrace{\hat{d}_{k+j}}_{\text{correction term}} \tag{5.7}$$

And we can separate the effect of past and future control moves

$$\hat{y}_{k+j}^c = s_1 \, \Delta u_{k+j-1} + s_2 \, \Delta u_{k+j-2} + \ldots + s_j \, u_k \quad \{\text{effect of current and future moves}\}$$

$$+ s_N \, \Delta u_{k-N+j} + s_{j+1} \, u_{k-1} + \ldots + s_{N-1} \, \Delta u_{k-N+j+1} \quad \{\text{effect of past moves}\}$$

$$+ \hat{d}_{k+j} \;\; \{\text{correction term}\} \tag{5.8}$$

The common assumption is that the correction term is constant in the future ("constant additive disturbance assumption"):

$$\hat{d}_{k+j} = \hat{d}_{k+j-1} = \ldots = \hat{d}_k = y_k - \hat{y}_k \tag{5.9}$$

Since there are no control moves beyond the control horizon of M steps, so

$$\Delta u_{k+M} = \Delta u_{k+M+1} = \dots = \Delta u_{k+P-1} = 0 \tag{5.10}$$

In matrix-vector form, a prediction horizon of P steps and a control horizon of M steps, yields

$$\begin{bmatrix} \hat{y}^c_{k+1} \\ \hat{y}^c_{k+2} \\ \vdots \\ \hat{y}^c_{k+j} \\ \vdots \\ \hat{y}^c_{k+P} \end{bmatrix} = \underbrace{\begin{bmatrix} s_1 & 0 & 0 & \dots & 0 & 0 \\ s_2 & s_1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & & & \\ s_j & s_{j-1} & s_{j-2} & \dots & \dots & s_{j-M+1} \\ \vdots & \vdots & \vdots & & & \vdots \\ s_P & s_{P-1} & s_{P-2} & \dots & \dots & s_{P-M+1} \end{bmatrix}}_{\text{PxM, dynamic matrix, } S_f} \underbrace{\begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-2} \\ \Delta u_{k+M-1} \end{bmatrix}}_{\text{Mx1, current \& future moves, } \Delta u_f} +$$

$$\underbrace{\begin{bmatrix} s_2 & s_3 & s_4 & \dots & s_{N-2} & s_{N-1} \\ s_3 & s_4 & s_5 & \dots & s_{N-1} & 0 \\ \vdots & \vdots & & & 0 & 0 \\ s_{j+1} & s_{j+2} & \dots & s_{N-1} & 0 & 0 \\ \vdots & \vdots & \vdots & & & \vdots \\ s_{P+1} & s_{P+2} & \dots & 0 & \dots & 0 \end{bmatrix}}_{\text{Px(N-2) matrix, } S_{past}} \underbrace{\begin{bmatrix} \Delta u_{k-1} \\ \Delta u_{k-2} \\ \vdots \\ \Delta u_{k-N+3} \\ \Delta u_{k-N+2} \end{bmatrix}}_{\text{(N-2)x1 past control moves, } \Delta u_{past}} + s_N \underbrace{\begin{bmatrix} \Delta u_{k-N+1} \\ \Delta u_{k-N+2} \\ \vdots \\ \Delta u_{k-N+P} \end{bmatrix}}_{\text{Px1 past inputs, } u_P}$$

$$+ \underbrace{\begin{bmatrix} d_{k+1} \\ d_{k+2} \\ \vdots \\ d_{k+P} \end{bmatrix}}_{\text{Px1 predicted disturbances, } \hat{d}} \tag{5.11}$$

The P x 1 matrix on the L.H.S is the Corrected output predictions matrix, $\hat{Y}^c$.

The P x M matrix is the dynamic matrix, S_f.

The M x 1 matrix is the current and future moves matrix, Δu_f.

The (N – 2) x 1 matrix is the past control moves matrix, Δu_{past}.

The last two matrices are the past inputs matrix, u_P, and predicted disturbances matrix, $\hat{d}$.

The above equation can be written using matrix-vector notation

$\hat{Y}^c = S_f\ \Delta u_f + S_{past}\ \Delta u_{past} + s_N\ u_P + \hat{d}$ (5.12)

In the last equation the corrected-predicted output response is naturally composed of a "forced response" (contributions of the current and future control moves) and a "free response" (the output changes that are predicted if there are no future control moves). The difference between the set-point trajectory, r, and the future prediction is given by eq. 5.13.

$r - \hat{Y}^c = r - [S_{past}\ \Delta u_{past} + s_N\ u_P + \hat{d}\ ] - S_f\ \Delta u_f$ (5.13)

which can be written as

$E^c = E - S_f\ \Delta u_f$ (5.14)

where the future predicted errors are composed of "free response" (E) and "forced response" ($-S_f\ \Delta u_f$) contributions.

The least-squares objective function is given by eq. 5.15.

$\Phi = \sum_{i=1}^{P}(e_{k+i}^{c})^2 + w \sum_{i=0}^{M-1}(\Delta u_{k+i})^2$ (5.15)

The quadratic terms can be written in matrix-vector form given by eq. 5.16 and eq. 5.17

$$\sum_{i=1}^{P}(e_{k+i}^{c})^2 = [e_{k+1}^{c}\ \ e_{k+2}^{c}\ \ \ldots\ \ e_{k+P}^{c}\ ]\begin{bmatrix} e_{k+1}^{c} \\ e_{k+2}^{c} \\ \vdots \\ e_{k+P}^{c} \end{bmatrix}$$

$= (E^c)^T E^c$ (5.16)

and

$$w \sum_{i=0}^{M-1}(\Delta u_{k+i})^2 = w\ [\Delta u_k\ \ \Delta u_{k+1}\ \ \ldots \Delta u_{k+M-1}\ ]\begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-1} \end{bmatrix}$$

$= \Delta u_f^T\ W\ \Delta u_f$ (5.17)

Therefore the objective function can be written in the form given by eq. 5.18.

$\Phi = (E^c)^T E^c + (\Delta u_f)^T W \Delta u_f$ (5.18)

Using the value of E^c , the objective function can be written as eq. 5.19.

$$\Phi = (E - S_f \Delta u_f)^T (E - S_f \Delta u_f) + (\Delta u_f)^T W \Delta u_f \tag{5.19}$$

The solution for the minimization of this objective function is

$$\Delta u_f = \underbrace{(S_f^T S_f + W) - 1 S_f^T}_{k} \underbrace{E}_{\text{Unforced errors}} \tag{5.20}$$

The current and future control move vector (Δu_f) is proportional to the unforced error vector €. That is, a controller gain matrix, K, multiplies the unforced error vector (the future errors that would occur if there were no control move changes implemented).

Because only the current control move is actually implemented, we use the first row of the K matrix, and

$$\Delta u_k = K_1 E \tag{5.21}$$

where K_1 represents the first row of the K matrix, where $K = (S_f^T S_f + W)^{-1} S_f^T$

5.2 DMC Tuning

Every controller design has some design parameters, which can be tuned to get the desired response of the controller. These parameters are called tuning parameters of the controller. The effect of these tuning parameters is discussed for SISO systems.

- **Model length and sample time** selection are not independent. The model length should be approximately the "settling time" of the process, that is, the time required to reach a new steady state after a step input change. Value of N is normally taken between 20 to 70. For most systems, the model length will be roughly 50. The sample time is usually on the order of one tenth the dominant time constant, so the model length is roughly the settling time of the process.
- **Prediction and control horizons** differ in length. Usually, the prediction horizon is selected to be much longer than the control horizon. This is particularly true if the control weighting factor is selected to be zero. Usually, if the prediction horizon is much longer than the control horizon, the control system is less sensitive to model

error. Often P = 20 or so, while M = 1 – 3.

- **Control weighting** is often set to zero if the prediction horizon is much longer than the control horizon. As the control horizon is increased, the control moves tend to become more aggressive so a larger weight is needed to penalize the control moves

5.3 Simulation Results

Van de Vusse Reactor problem [5.2], has been considered in this section.

5.3.1 Problem Definition

The reactor is considered at an operating point where optimal yield with respect to the desired product is achieved. There are two interesting features of this problem. First is that the steady state gain changes its sign at the operating point, therefore linear controllers will not be able to stabilize this reactor and accomplish satisfactory performance. Second is that the zero dynamics changes its properties at the operating point, therefore the qualitative behaviour of CSTR differ for different set-points and disturbances [5.3].

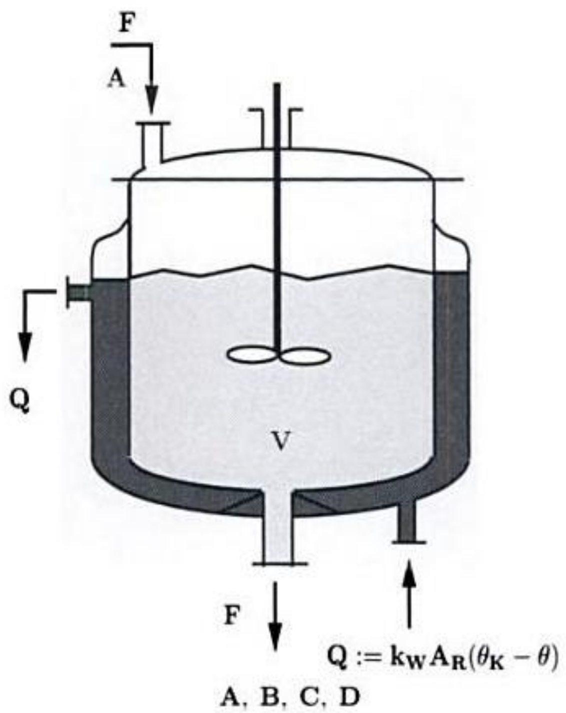

Fig. 5.1 Schematic representation of the reactor [5.3].

Van de Vusse reactions are

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$

$$2A \xrightarrow{k_3} D \qquad (5.22)$$

Fig. 5.1 shows the schematic representation of the reactor. The flow, F, fed to the reactor contains only substance A. Main reaction is given by transformation of substance A into the product B. Substance A also reacts in unwanted parallel reaction to generate a by-product D. Furthermore, substance C is formed in an unwanted decomposition of the product B.

The continuous state space model of the reactor considered in this section is,

A= [-2.4048 0; 0.8333 -2.2381];

B= [7; -1.117];

C= [0 1];

D= [0];

where the measured state (output) is the concentration of the second component and the manipulated input is the dilution rate [5.2].

5.3.2 Implementation in MATLAB

Effect of Prediction horizon

a) For Prediction horizon, P = 15

   Model length, N = 50

   Control horizon, M = 1

   Control weight, w is set to zero because P is much longer than M.

```
>> van_de_vusse

enter plant in 1.statespace 2.transferfunction 3.polezero 4.frquencyresponse 1

enter A matrix  [-2.4048 0; 0.8333 -2.2381];

enter B matrix [7; -1.117];

enter C matrix [0 1]

enter D matrix [0]


numm =
```

0   -0.0751   0.1001

Sf =

0
-0.0751
-0.0940
-0.0768
-0.0376
0.0137
0.0704
0.1281
0.1840
0.2362
0.2840
0.3270
0.3650
0.3983
0.4273
0.4522
0.4735
0.4917
0.5071
0.5202
0.5311
0.5403

0.5480

0.5544

0.5597

u =

Columns 1 through 15

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Columns 16 through 25

0	0	0	0	0	0	0	0	0	0

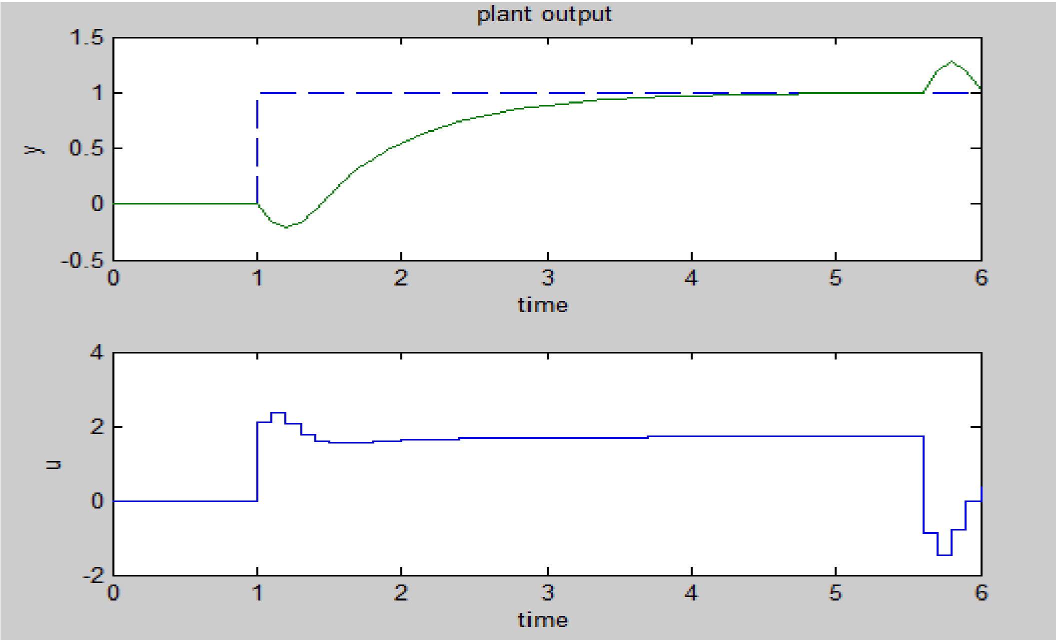

Fig. 5.2 Plant output for a step set-point change, model length, N = 50, control horizon, M = 1, prediction horizon, P = 15, no weighting on input.

b) For Prediction horizon, P = 30

Model Length, N = 50

Control horizon, M = 1

Control weight, w is set to zero because P is much longer than M.

```
>> van_de_vusse
enter plant in 1.statespace 2.transferfunction 3.polezero 4.frquencyresponse 1
enter A matrix [-2.4048 0; 0.83333 -2.2381];
enter B matrix [7; -1.117];
enter C matrix [0 1];
enter D matrix [0];
numm =


     0   -0.0751    0.1001


Sf =
     0
   -0.0751
   -0.0940
   -0.0767
   -0.0376
    0.0137
    0.0704
    0.1281
    0.1840
    0.2363
    0.2841
    0.3270
    0.3650
```

```
    0.3983
    0.4273
    0.4522
    0.4736
    0.4918
    0.5072
    0.5202
    0.5311
    0.5403
    0.5480
    0.5544
    0.5597
    0.5641
    0.5678
    0.5708
    0.5733
    0.5754


u =

  Columns 1 through 15

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0

  Columns 16 through 30

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

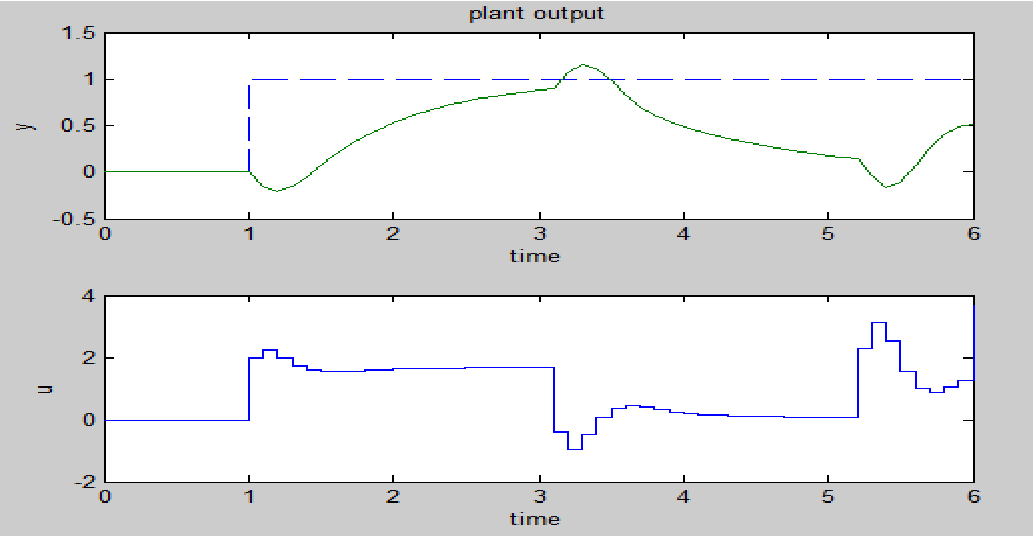

Fig. 5.3 Plant output for a step set-point change, model length, N = 50, control horizon, M = 1, prediction horizon, P = 30, no weighting on input.

c) For Prediction horizon, P = 8

    Model length, N = 50

    Control horizon, M = 1

    Weight, w = 0

```
>> van_de_vusse

enter plant in 1.statespace 2.transferfunction 3.polezero 4.frquencyresponse 1

enter A matrix [-2.4048 0; 0.8333 -2.2381];

enter B matrix [7; -1.117];

enter C matrix [0 1];

enter D matrix [0];

numm =

     0   -0.0751    0.1001

Sf =

     0
```

-0.0751

-0.0940

-0.0768

-0.0376

0.0137

0.0704

0.1281

u =

0 0 0 0 0 0 0 0

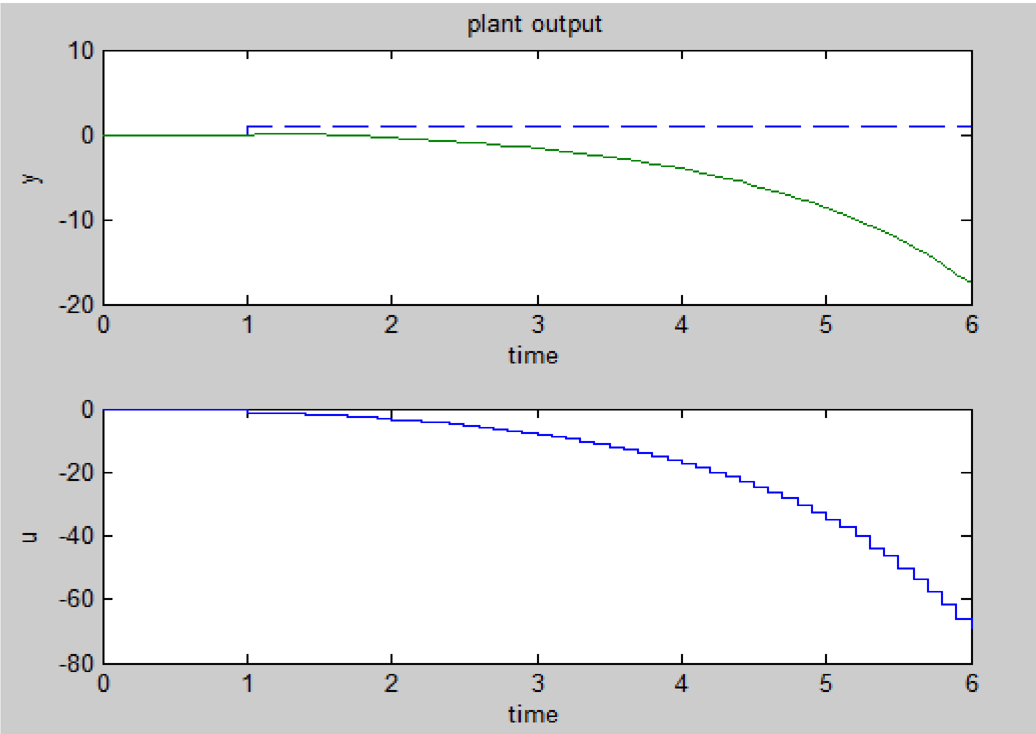

Fig. 5.4 Plant output for a step set-point change, model length, N = 50, control horizon, M = 1, prediction horizon, P = 8, no weighting on input.

d) For Prediction horizon, P = 5

Model length, N = 50

Control horizon, M = 1

Weight, w = 0

```
>> van_de_vusse

enter plant in 1.statespace 2.transferfunction 3.polezero 4.frquencyresponse 1

enter A matrix [-2.4048 0; 0.8333 -2.2381];

enter B matrix [7; -1.117];

enter C matrix [0 1];

enter D matrix [0];



numm =

      0   -0.0751    0.1001

Sf =

      0

  -0.0751

  -0.0940

  -0.0768

  -0.0376

u =

    0     0     0     0     0
```

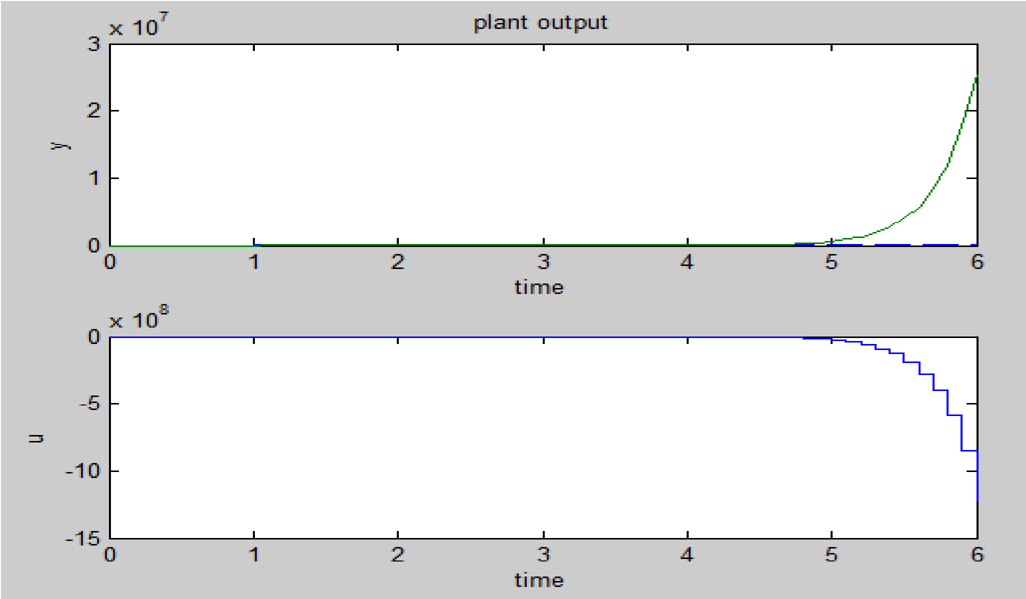

Fig. 5.5 Plant output for a step set-point change, model length, N = 50, control horizon, M = 1, prediction horizon, P = 5, no weighting on input.

Effect of Model length

e) For Prediction horizon, P = 25

Model length, N = 70

Control horizon, M = 1

Control weight, w is set to zero because P is much longer than M.

>> van_de_vusse

enter plant in 1.statespace 2.transferfunction 3.polezero 4.frquencyresponse 1

enter A matrix [-2.4048 0; 0.8333 -2.2381];

enter B matrix [7; -1.117];

enter C matrix [0 1]

enter D matrix [0]

numm =

0 -0.0751 0.1001

Sf =

0
-0.0751
-0.0940
-0.0768
-0.0376
0.0137
0.0704
0.1281
0.1840
0.2362
0.2840
0.3270
0.3650
0.3983
0.4273
0.4522
0.4735
0.4917
0.5071
0.5202
0.5311
0.5403
0.5480
0.5544

0.5597

u =

Columns 1 through 15

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 16 through 25

0 0 0 0 0 0 0 0 0 0

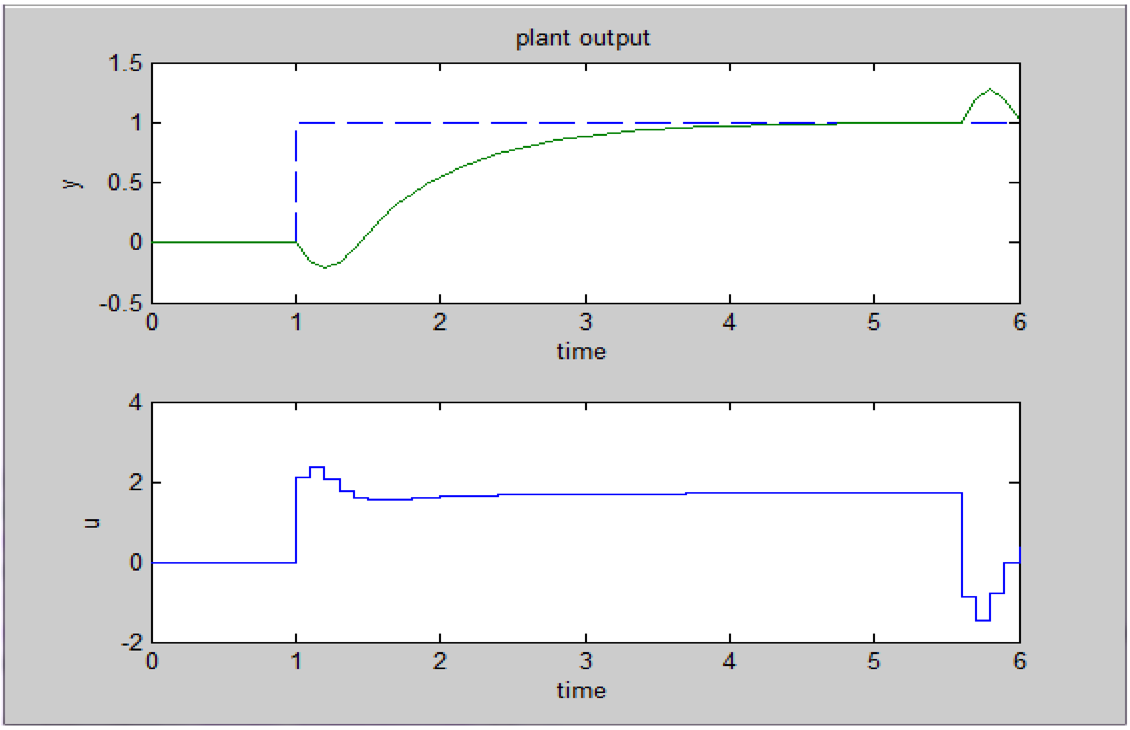

Fig. 5.6 Plant output for a step set-point change, model length, N = 70, control horizon, M = 1, prediction horizon, P = 25, no weighting on input.

>> van_de_vusse

```
enter plant in 1.statespace 2.transferfunction 3.polezero 4.frquencyresponse 1

enter A matrix [-2.4048 0; 0.8333 -2.2381];

enter B matrix [7; -1.117];

enter C matrix [0 1];

enter D matrix [0];


numm =

         0   -0.0751    0.1001


Sf =

         0
   -0.0751
   -0.0940
   -0.0768
   -0.0376
    0.0137
    0.0704
    0.1281
    0.1840
    0.2362
    0.2840
```

0.3270

0.3650

0.3983

0.4273

0.4522

0.4735

0.4917

0.5071

0.5202

0.5311

0.5403

0.5480

0.5544

0.5597

u =

Columns 1 through 15

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Columns 16 through 25

0	0	0	0	0	0	0	0	0	0

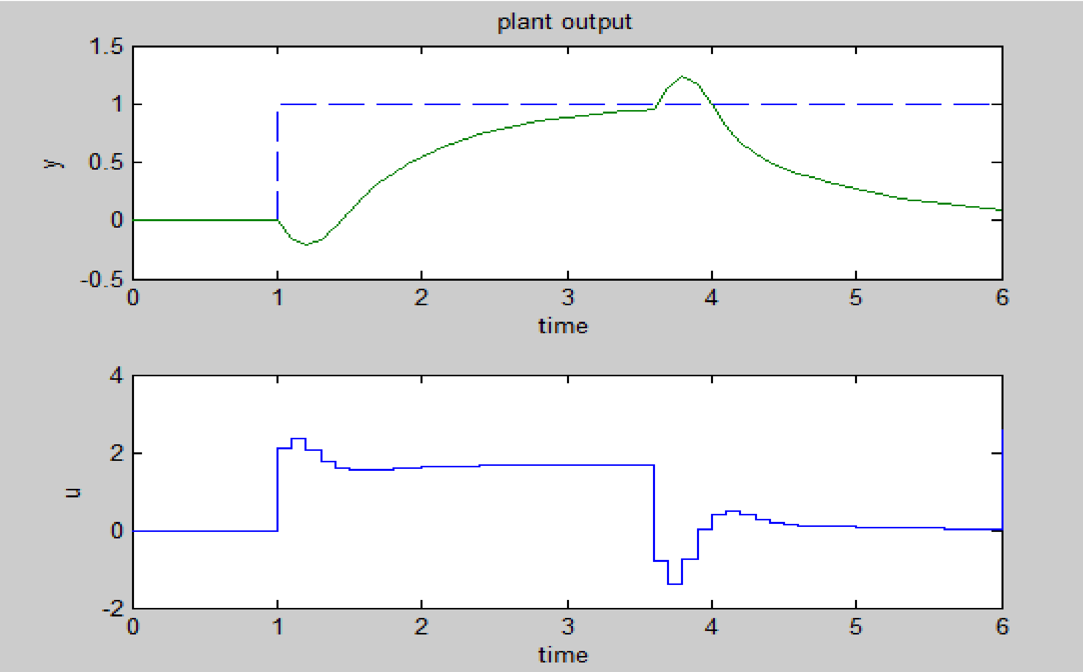

Fig. 5.7 Plant output for a step set-point change, model length, N = 50, control horizon, M = 1, prediction horizon, P = 25, no weighting on input.

5.3.3 Inference

The comparison of fig. 5.2 - 5.5 shows the effect of changing the prediction horizon while keeping the model length and control horizon fixed. When P = 30, control action required is much greater than when P = 15. If we have a fixed control horizon, then it is seen that choosing a smaller prediction horizon results set point being achieved in smaller time.

However, the shorter prediction horizon is more sensitive to model uncertainty.

The comparison of fig. 5.6 and fig. 5.7 shows the effect of changing model length while keeping prediction horizon and control horizon fixed. Choosing a smaller model length does not capture the complete dynamics of the process. This results in a model error and poor performance. It can be seen that N = 50 gives better results than N = 70. The performance degrades sharply as N increases.

References

[5.1] Cutler, C.R., and B.L. Ramaker, *Dynamic Matrix Control- A computer control algorithm*, Proceedings of Joint Automatic Control conference, San Francisco, CA, Paper

WP5-B, 1980

[5.2] Bequette, B. W., *Process Control - Modeling, Design, and Simulation,* PHI, pp. 488-509, 2003.

[5.3] Simone Loureiro De Oliveira, *Model Predictive Control for Constrained Nonlinear System*, IfA, Publication No. 6, pg 107.

CHAPTER 6

CONSTRAINED MPC

In this chapter MPC is extended such that it contains constraints. The MPC has to consider the limits of physical system being controlled. It is rarely that a physical system does not have boundaries. The performance of control system can deteriorate significantly when the control signals from the original design meet with operational constraints. With a small modification, the degree of performance deterioration can be reduced if the constraints are incorporated in the implementation, leading to the idea of constrained problem [6.1].

6.1 Operational constraints

There are three major types of operational constraints. The first two types deals with constraints imposed on the control variable u(k), third type deals with constraint on output variable y(k) or state variable x(k) constraints.

6.1.1 Constraint on rate of change of control variable

These are hard constraints on the size of control signal movements, i.e., on the rate of change of control variables ($\Delta u(k)$). Suppose that for a single input, the upper limit is Δu^{max} and the lower limit is Δu^{min}, the constraints are specified in the form given by eq. 6.1.

$$\Delta u^{min} \leq \Delta u(k) \leq \Delta u^{max} \qquad (6.1)$$

The rate of change constraints can be used to impose directional movement constraints on control variable. If $\Delta u(k)$ can only increase and not decrease, then we can select $0 \leq \Delta u(k) \leq \Delta u^{max}$. The constraint on Δu can be used to cope with the cases where the rate of change of control amplitude is limited in value [6.1].

6.1.2 Constraint on amplitude of control variable

These are the physical hard constraints on the system and are most commonly encountered. They are of the form given by eq. 6.2.

$$u^{min} \leq u(k) \leq u^{max} \qquad (6.2)$$

Here, u(k) is only an incremental variable, not the actual physical variable. These are the most commonly encountered constraints among all the constraint types [6.1].

6.1.3 Output Constraints

The actual operating range on the plant output can also be specified. Suppose that the output y(k) has an upper limit y^{max} and a lower limit y^{min}, then the output constraints are specified by eq. 6.3.

$$y^{min} \leq y(k) \leq y^{max} \tag{6.3}$$

Output constraints are often implemented as "soft" constraints in the way that a slack variable $s_v > 0$ is added to the constraints, forming

$$y^{min} - s_v \leq y(k) \leq y^{max} + s_v \tag{6.4}$$

Output constraints often cause large changes in both the control and incremental control variable when they are enforced or become active. When that happens, the control and incremental control variables can violate their own constraints and the problem of constraint conflict occurs. In the situations where the constraints on control variables are more essential to plant operation, output constraints are often relaxed by selecting a larger slack variable, s_v to solve the conflict problem [6.1].

6.1.4 Constraints in a MIMO setting

If there is more than one input, then the constraints are specified for each input independently. In the multi-input case, suppose that the constraints are given for the upper limit by eq. 6.5

$$[\ \Delta u_1^{max}\ \ \Delta u_2^{max}\ \ \ldots\ldots\ \ \Delta u_m^{max}\ ] \tag{6.5}$$

and lower limits is given by eq. 6.6.

$$[\ \Delta u_1^{min}\ \ \Delta u_2^{min}\ \ldots\ldots\ \Delta u_m^{min}\ ] \tag{6.6}$$

Each variable with rate of change is specified as

$$\Delta u_1^{min} \leq \Delta u_1(k) \leq \Delta u_1^{max}$$

$\Delta u_2^{min} \leq \Delta u_2(k) \leq \Delta u_2^{max}$

$\vdots$

$$\Delta u_m^{min} \leq \Delta u_m(k) \leq \Delta u_m^{max} \tag{6.7}$$

Similarly, suppose that the constraints for the upper limit of control signal are given by eq. 6.8.

$$[\ u_1^{max}\ \ u_2^{max}\ \ \ldots\ldots\ \ u_m^{max}\ ] \tag{6.8}$$

and lower limit are given by eq. 6.9.

$$[\ u_1^{min}\ \ u_2^{min}\ \ldots\ldots\ u_m^{min}\ ] \tag{6.9}$$

Then the amplitude of each control signal is required to satisfy the constraints

$u_1^{min} \leq u_1(k) \leq u_1^{max}$

$u_2^{min} \leq u_2(k) \leq u_2^{max}$

$\vdots$

$$u_m^{min} \leq u_m(k) \leq u_m^{max} \tag{6.10}$$

6.2 Hard and Soft Constraints

Hard constraints are the physical limitations of the real processes e.g. actuator extreme positions.

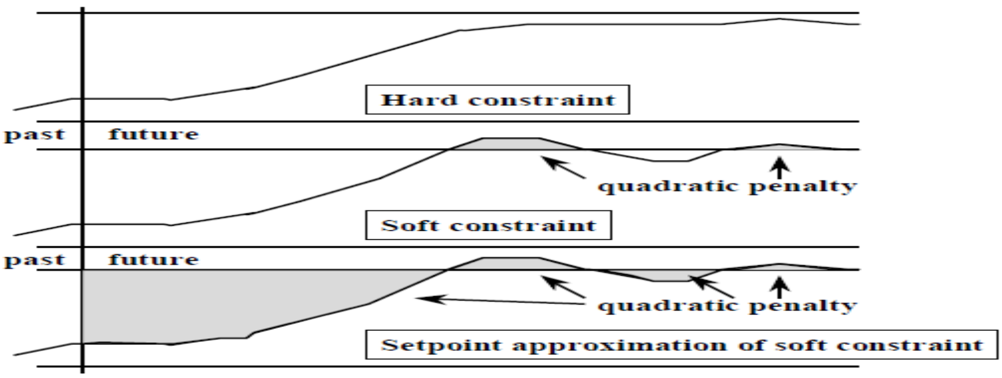

Fig. 6.1 Hard and soft constraints [6.2]

Hard constraints must not be violated. Soft constraints can be violated though at some penalty e.g. loss of product quality, constraints for the system inner variables. The violation can be penalised in the objective function. Fig. 6.1 shows the hard and soft constraints and also the quadratic penalty for violation of soft constraint.

($\Delta u^{min} \leq \Delta u(k) \leq \Delta u^{max}$) and ($u^{min} \leq u(k) \leq u^{max}$) are the hard constraints.

($y^{min} \leq y(k) \leq y^{max}$) is the soft constraint.

6.3 Quadratic DMC (QDMC)

QDMC considers constraints on the manipulated inputs. To use a standard quadratic program (QP), the input constraints need to be written in terms of the control moves, Δu_{k+i}. since the previously implemented control action (Δu_{k-l}) is known, it can be written in the form of equation 6.1.

$u_k = u_{k-l} + u_k$

$u_{k+l} = u_{k-l} + \Delta u_k + \Delta u_{k+l}$ (6.11)

Since the manipulated input constraints are enforced over the control horizon of M steps, the input constraints and eq. (6.11) yield

$$\begin{bmatrix} u_{min} \\ u_{min} \\ \vdots \\ u_{min} \end{bmatrix} \leq \begin{bmatrix} u_{k-1} \\ u_{k-1} \\ \vdots \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & & & \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-1} \end{bmatrix} \leq \begin{bmatrix} u_{max} \\ u_{max} \\ \vdots \\ u_{max} \end{bmatrix}$$

Most QP codes use a "one-sided" form

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & & & \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-1} \end{bmatrix} \geq \begin{bmatrix} u_{min} - u_{k-1} \\ u_{min} - u_{k-1} \\ \vdots \\ u_{min} - u_{k-1} \end{bmatrix} \tag{6.12a}$$

$$-\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & & & \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-1} \end{bmatrix} \geq \begin{bmatrix} u_{k-1} - u_{max} \\ u_{k-1} - u_{max} \\ \vdots \\ u_{k-1} - u_{max} \end{bmatrix} \tag{6.12b}$$

which have the form $A\Delta u_f \geq b$.

The velocity constraints are implemented as bounds on the control moves given by eq. 6.13a.

$$\begin{bmatrix} \Delta u_{min} \\ \Delta u_{min} \\ \vdots \\ \Delta u_{min} \end{bmatrix} \leq \begin{bmatrix} \Delta u_{k} \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-1} \end{bmatrix} \leq \begin{bmatrix} \Delta u_{max} \\ \Delta u_{max} \\ \vdots \\ \Delta u_{max} \end{bmatrix} \tag{6.13a}$$

The majority of constrained MPC problems can be solved based on the input constraints considered above. It is desirable to predict the process outputs to be within a range of minimum and maximum values

$$y_{min} \leq \hat{y}^{c}_{k+i} \leq y_{max} \tag{6.14}$$

The equation for unconstrained control (5.13) is rewritten as eq. 6.16.

$$\hat{Y}^{c} = S_f \; \Delta u_f + S_{past} \; \Delta u_{past} + s_N \, u_P + \hat{d} \tag{6.15}$$

$$\hat{Y}^{c} = S_f \; \Delta u_f + f \tag{6.16}$$

where *f*, the free response of the “corrected-predicted output” (if no current and future control moves are made) is given by eq. 6.17.

$$f = S_{past} \, \Delta u_{past} + s_N \; u_P + \hat{d} \tag{6.17}$$

so that eq. 6.14 can be written in the form of eq. 6.18.

$$y_{min} - f \leq S_f \; \Delta u_f \leq y_{max} - f \tag{6.18}$$

or, in terms of one-sided inequalities given by eq. 6.19.

$$S_f \; \Delta u_f \geq y_{min} - f$$

$$- S_f \; \Delta u_f \;\geq - \, y_{max} + \; f \tag{6.19}$$

Using matrix-vector notation, the quadratic programming problem is stated by eq. 6.20.

$$\min \phi = \frac{1}{2} \Delta u_f^T \; H \Delta u_f + c^T \; \Delta u_f \tag{6.20}$$

$$\text{s.t.} \quad A \Delta u_f \geq b$$

$\Delta u_{min} \le \Delta u_f \le \Delta u_{max}$ (6.21)

where the matrices and vectors in the objective function are given by equations 6.22 and 6.23.

$H = S_f^T \ S_f + W$ (6.22)

$c^T = E^T S_f$ (6.23)

and the inequality matrices, A and b, in eq. 6.21 incorporate the matrices in eq. 6.12 and eq. 6.19, with appropriate dimensionality [6.3].

6.4 Simulation Results

A temperature control problem has been considered. MATLAB's model predictive toolbox has been used. Default input constraints are applied.

6.4.1 Problem Definition

The water temperature in a heated boiler is related to the heater power, q, and the ambient air temperature, θ, is related to the heater input power, q, and the ambient air temperature, θ_a, according to the eq. 6.24.

$T(d\theta/dt) = kq + \theta a - \theta$ (6.24)

where it is assumed that T=1 hour and k=0.2 °C/kW .

Predictive control is to be applied to keep the water at a desired temperature, and a sampling interval Ts=0.25 hour is to be used.

Suppose the air temperature follows a sinusoidal variation with amplitude 10°C given by eq. 6.25.

$\theta a(t) = 15 + 10\sin(2\pi t/24)$ (6.25)

Consider the problem if there is constraint on the input power given by eq. 6.26.

$-50 < q < 50$ (6.26)

6.4.2 Implementation in MATLAB

The MPC toolbox function ***scmpc*** is used. It simulates closed loop problems by designing an MPC controller for constrained problems. The function usage is as follows:

[y,u,ym]= scmpc(pmod,imod,ywt,uwt,M,P,tend, ... r,ulim,ylim,Kest,z,v,w,wu)

pmod- plant model

imod- internal model used as basis for controller design

ywt- weight for setpoint tracking

uwt- weight for change in manipulated variable

ulim- input constraints

ylim- output constraints

z- measurement noise

v- measured disturbance

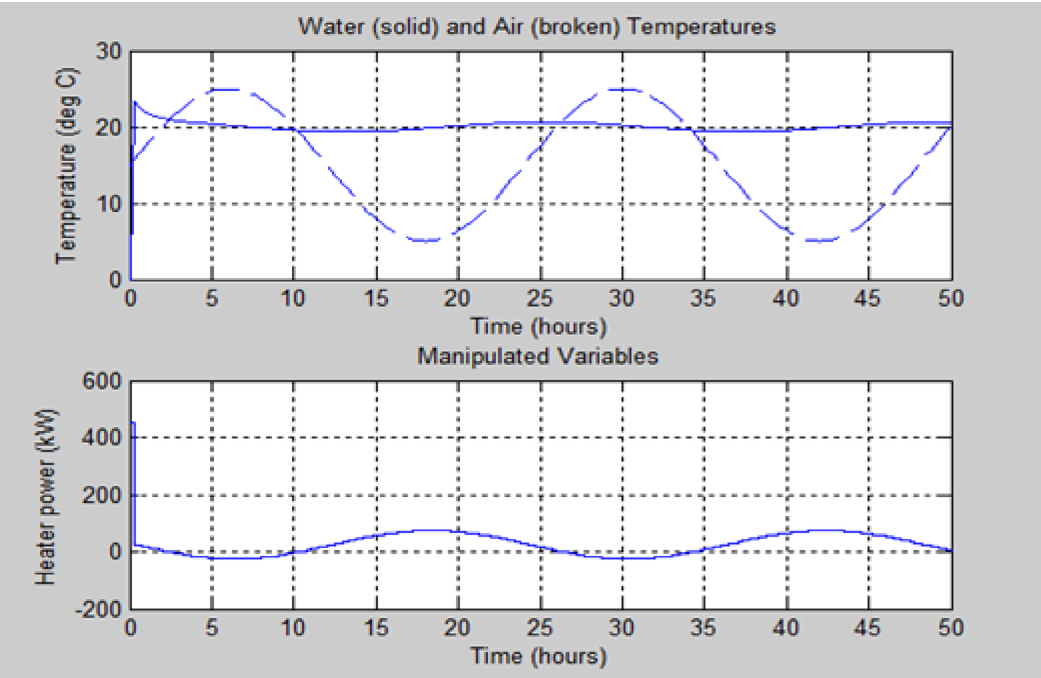

Fig. 6.2 Output after applying MPC to control water temperature.

6.4.3 Inference

Fig. 6.2 shows that a predictive controller with standard 'DMC' disturbance model does not compensate perfectly for a sinusoidal variation of the air temperature if the air temperature is not measured. The water temperature oscillates with amplitude of about 0.5 deg C if there are no constraints on the heater power. It is seen that if the air temperature is measured and used for feed forward control, then air temperature is perfectly compensated, providing that model is perfect. If some modeling error occurs, then residual oscillation remains. But if we want to remove it completely, then we have to model the sinusoidal disturbance and design a suitable observer, even if the air temperature is not measured.

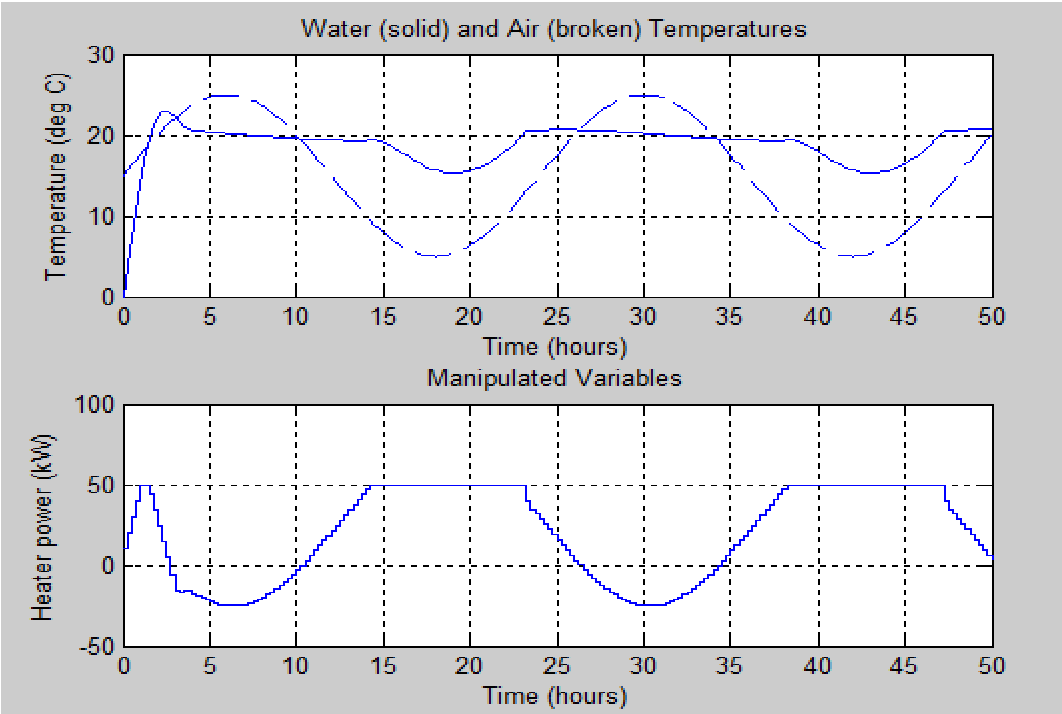

Fig. 6.3 Output after applying MPC to control of water temperature, with constraint on input power.

Fig. 6.3 shows that water temperature oscillates with amplitude of about 5 deg C when there is an input constraint on the heater power. The control action does not exceed 50 kW.

References

[6.1] Wang, L., *Model Predictive Control system Design and Implementation using MATLAB*, Springer, pp. 43-83, 2009.

[6.2] Qin, S. J., Badgwell, T. A., *Industrial Model Predictive Control – An Updated Overview*, Survey hand-out, AspenTech, 2002.

[6.3] Bequette, B. W., *Process Control - Modeling, Design, and Simulation*, PHI, pp. 488-509, 2003.

CHAPTER 7

COMPARISON OF UNCONSTRAINED AND CONSTRAINED CONTROL BY MPC

7.1 Problem Definition

A tool positioning platform is which is commonly used in industries is considered.

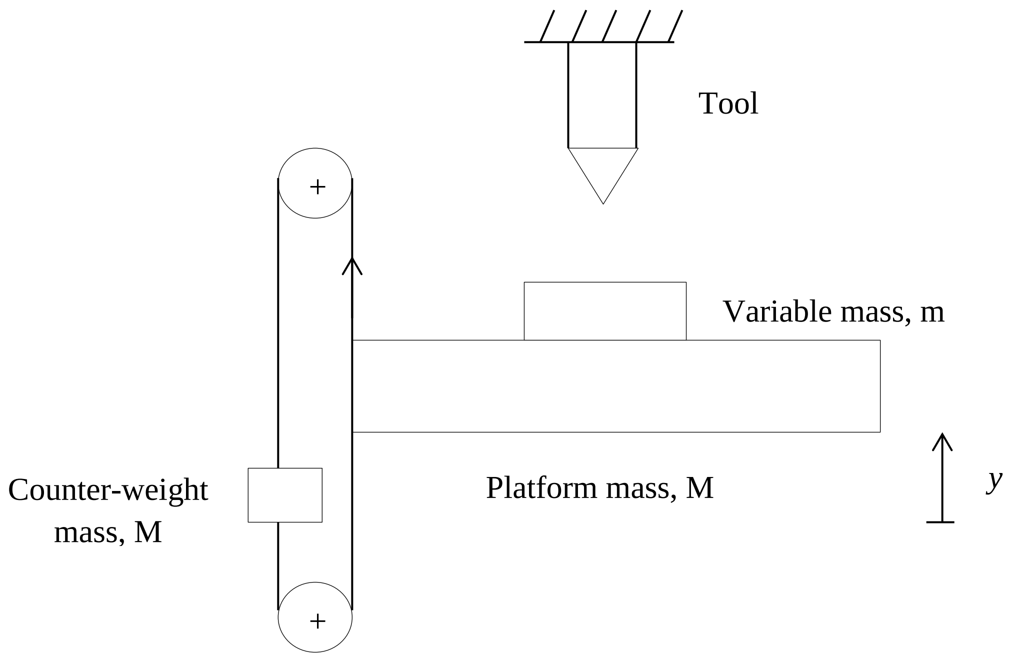

Fig. 7.1 Assembly of positioning platform

The vertical position y of a positioning platform is controlled by a motor which applies a vertical force F to the platform (Figure 7.1). The platform has mass M and carries a variable load of mass m; the unloaded weight of the platform is balanced by a counter-weight. The force F is proportional to the voltage v applied to the motor, so that $F = K_v v$ where K_v is a fixed gain.

Assuming m is small enough that $M + m \approx M$, the unknown load constitutes a constant disturbance in the discrete-time model of the system for sampling interval T:

$$x(k + 1) = A\,x(k) + B\,u(k) + Dd \quad (7.1)$$

$$e(k) = C\,x(k) \quad (7.2)$$

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (7.3)$$

$$B = \frac{k_v}{2M}\begin{bmatrix} T^2/2 \\ T \end{bmatrix} \quad (7.4)$$

$$C = [1 \quad 0] \quad (7.5)$$

$$D = -\frac{g}{2M}\begin{bmatrix} T^2/2 \\ T \end{bmatrix} \quad (7.6)$$

where e is the error in y relative to a desired steady-state height y^0, and

$$x(k) = \begin{bmatrix} y(kT) - y^0 \\ y(\dot{k}T) \end{bmatrix} \quad (7.7)$$

$$u(k) = v(kT) \quad (7.8)$$

$$d = m \quad (7.9)$$

The motor input voltage is subject to constraints given by eq. 7.10.

$$-1 \le v \le 1 \quad (7.10)$$

7.2 Implementation in MATLAB

Unconstrained Control

Assuming the model parameters M = 20, K_v = 10 NV^{-1}, T = 0.1s, closed loop platform response under MPC is observed.

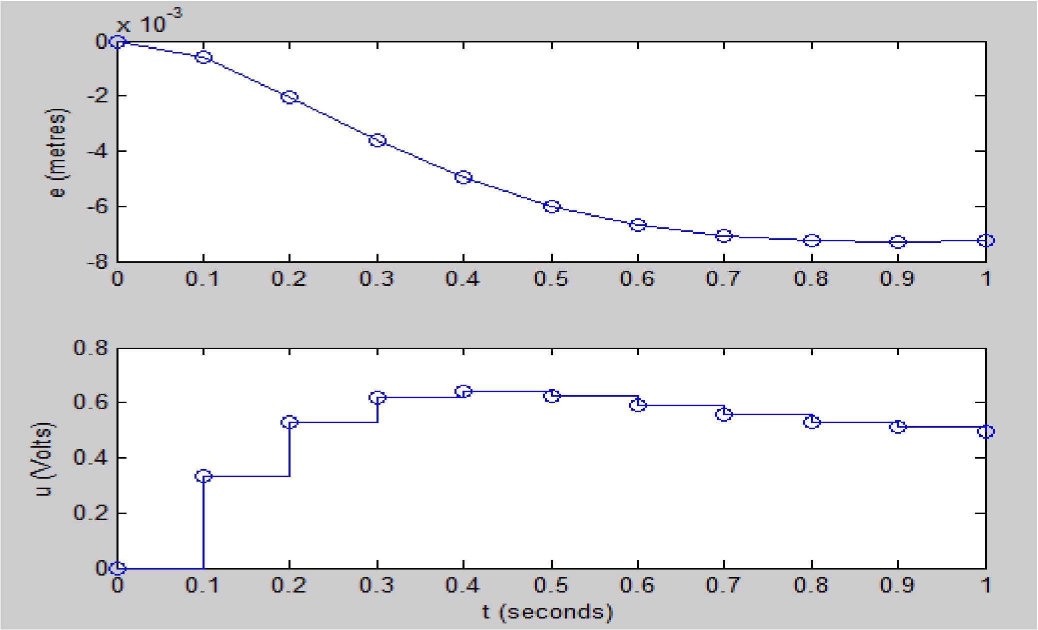

Fig. 7.2 Closed-loop platform response for unconstrained MPC

Constrained Control

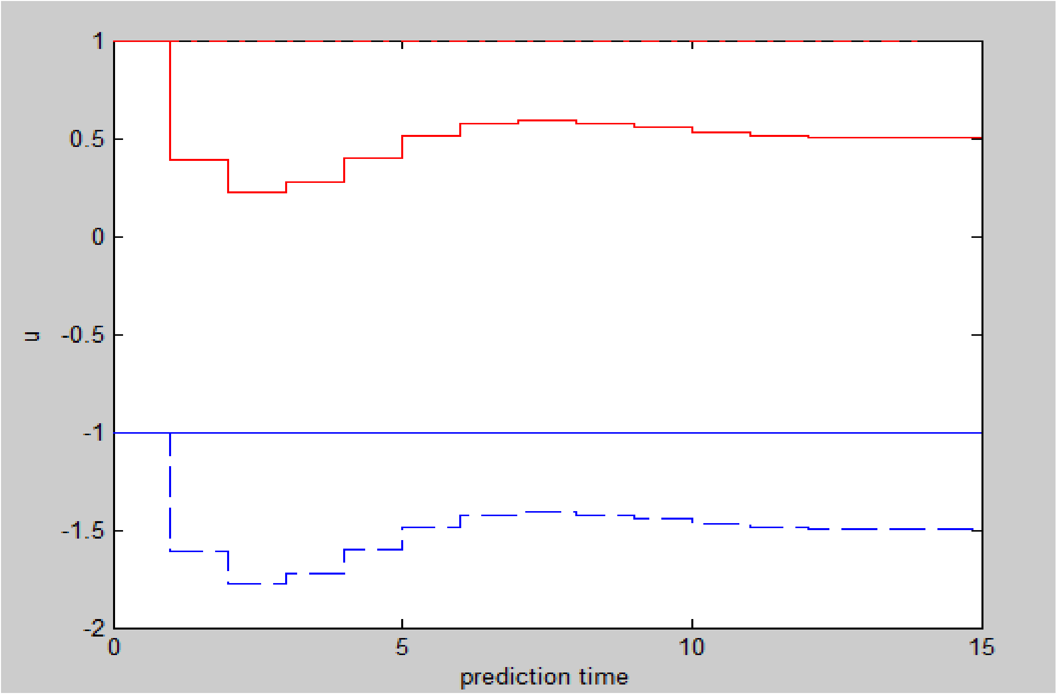

Fig. 7.3 Bounds on input predictions

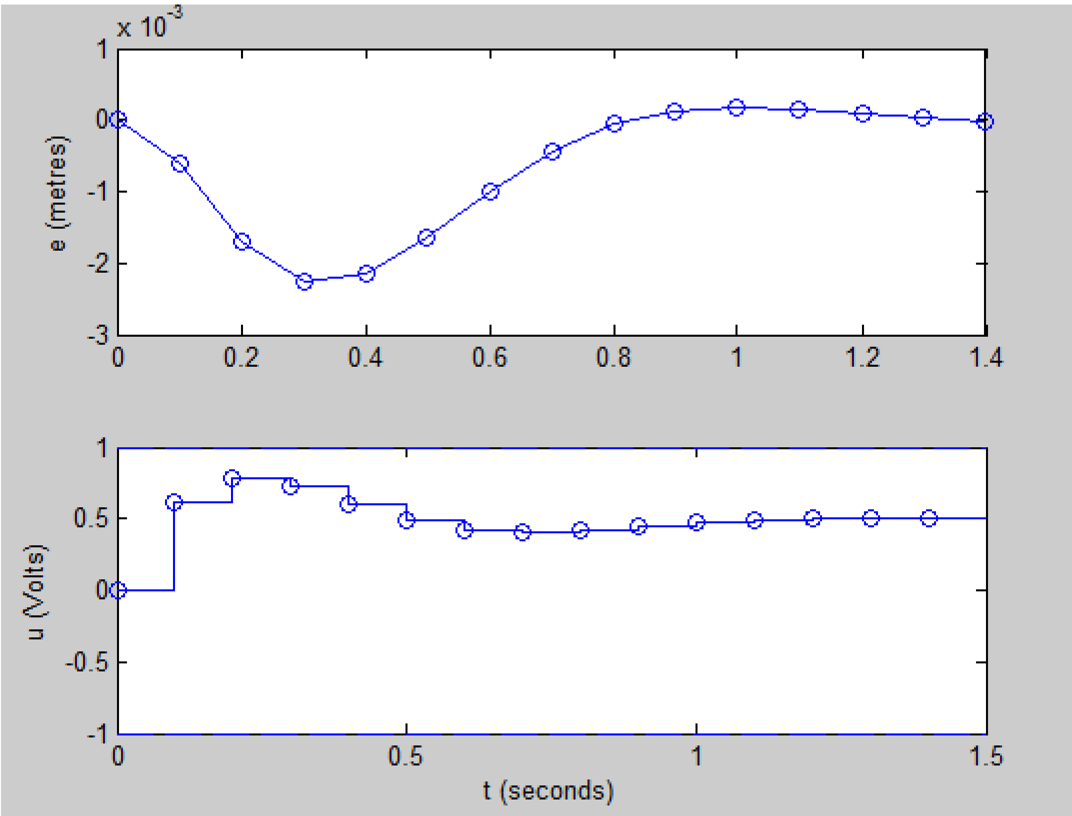

Fig. 7.4 Closed-loop platform response under MPC incorporating constraints

7.3 Inference

Fig. 7.2 shows the closed-loop platform response for unconstrained MPC. The error, e, in y relative to desired steady-state height y^0 is plotted.

To ensure constraint satisfaction, the constraints must be imposed on the predictions that correspond to the worst-case disturbance values (i.e. d = 0 and d = 0.5). This is achieved by generating two sets of predictions (i.e. the predictions that are obtained with d = 0 and d = 0.5).

Fig. 7.3 shows the bounds on input predictions for all the values of disturbance in the range $0 \le d \le 0.5$

Fig. 7.4 shows the closed-loop platform response under constrained MPC. The error, e, in y relative to desired steady-state height y^0 is plotted for constrained MPC too. The response is having zero steady-state error.

Comparing the fig. 7.2 and fig. 7.4 the steady-state error for unconstrained MPC does not return to zero unlike in the case of constrained control. In presence of constraints the closed-loop performance can deteriorate severely. The modification in the predictive law enables the system to handle constraints without significant performance deterioration.

CHAPTER 8

CONCLUSION

MPC technology has progressed steadily in the past years since the first IDCOM and DMC applications. It has large number of applications in industry, with a solid foundation in refining and petrochemicals, and significant penetration into a wide range of application areas from chemicals to food processing. It has already been established that the reason for its success is its ability to handle constraints and multivariable control problems.

Current generation MPC technology offers significant new capabilities but the controllers still retain, for the most part, an IDCOM-like or a DMC-like personality. Most MPC techniques have been based on step or impulse response models, but there has been a recent trend toward state space models. Regardless of the model type, the same basic ideas are used.

This dissertation focusses on analysing the response of system under unconstrained as well as constrained MPC.

Van de Vusse reactor problem established the basic understanding of tuning of model predictive controllers. It is important to make certain that model length (N) is long enough to capture the steady-state change. It is also important to make certain that the prediction horizon (P) is long enough to avoid stability problems, particularly for systems with inverse response behaviour. Also, control horizon (M) is much shorter than prediction horizons, yielding more robust performance.

In the temperature control problem the behaviour of system with input constraints under constrained model predictive control is shown.

The tool positioning platform problem establishes that if we do not pay attention to the saturation of the control, then in the presence of constraints, the closed-loop control performance could severely deteriorate. A modification in the predictive control law will enable the system to handle the constraint without significant performance deterioration.

There are some problems present with the existing techniques like the DMC and QDMC uses a constant disturbance assumption. It may not yield good performance if the real disturbances occur at the plant input. Also, there is the finite horizon problem. Control performance deteriorates if the prediction or control horizons are not chosen correctly, even if the model is perfect.

Several technical advances have not yet been incorporated into industrial MPC technology. These include use of an infinite prediction horizon to guarantee nominal closed loop stability, and using linear estimation theory to improve output feedback. In addition, robust stability conditions have been developed for a modified QDMC algorithm. It would seem that the company which first implements these advances will have a significant marketing and technical advantage.

Next-generation MPC technology is likely to include multiple objective functions, an infinite prediction horizon, non-linear process models and better use of model uncertainty estimates.

LIST OF PUBLICATIONS

[1] Ruchika, Neha Raghu, *Analysis of Van de Vusse Reactor using Model Predictive Control,* International Journal of Emerging Technology and Advanced Engineering, Vol. 3, Issue 6, June 2013.

[2] Ruchika, Neha Raghu, *Model Predictive Control – History and Development,* International Journal of Engineering Trends and Technology, Vol. 4, Issue 6, 2013.