PERFORMANCE ANALYSIS OF LP-MPC CASCADE CONTROL SYSTEMS

PERFORMANCE ANALYSIS OF LP-MPC CASCADE CONTROL SYSTEMS

by

ALEXEI NIKANDROV, M.Sc. (Eng)

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Master of Applied Science

McMaster University

MASTER OF APPLIED SCIENCE (2007)	McMaster University
(Chemical Engineering)	Hamilton, Ontario, Canada

TITLE:	Performance Analysis of LP-MPC Cascade Control Systems
AUTHOR:	Alexei Nikandrov, M.Sc(Eng)
	(State Technical University, Saint-Petersburg, Russia)
SUPERVISOR:	Dr. C. L. E. Swartz
NUMBER OF PAGES:	xiii, 126

ABSTRACT

Model Predictive Control (MPC) algorithms are widely applied in the chemical process industry. The main advantage of these controllers over others is their ability to provide multivariable control of the process subject to specified constraints. The presence of degrees of freedom in the plant provide an opportunity for the introduction of an optimization level (Real-Time Optimization (RTO) level), to determine optimal set points and target values for controlled variables and manipulated variables respectively, and the constraints the plant should follow to provide maximum profit.

Industrial MPC controllers typically include an upper level steady-state optimizer, which usually comprises a linear programming (LP) or quadratic programming (QP) problem. This local optimizer may serve either as an integrating level between the low frequency nonlinear steady-state RTO and regulatory level, or as an independent optimizer with an economic objective function. Many researchers have reported success of LP-MPC cascade control system implementations (Sorensen and Cutler, 1998; Verne *et al.*, 1999). However, despite its apparent success, poor LP-MPC cascade system performance and possible instability have also been reported. In particular, Shah *et al.* (2002) show that in the presence of a steady-state LP optimizer, the set-points could have a large variation relative to the controlled variable variation; thus the LP could degrade the MPC performance by sending highly variable set-points to the controller. Kozub (2002) indicates that in a control system with an LP steady-state optimizer, an LP instability problem may arise under certain conditions.

These observations motivated research which aims to investigate the effect of the various factors on the stability and performance of the two-level LP-MPC cascade control system. Such factors include plant/model mismatch, the frequency of LP implementation, the LP objective function, constraints and type of disturbances.

Since the optimization can be executed at different frequencies, two most common scenarios are considered: (i) when the LP is implemented at steady-state only and (ii) when the LP is implemented at every MPC iteration. Initially, steady-state LP optimization only is considered and it is shown that the set-points may fail to converge to constant values in the absence of external disturbances under certain conditions. Then, the effects of optimization frequency and control structure on the closed-loop properties of the LP-MPC control system are investigated. Results of a number of case studies are shown, and root causes for observed behavior discussed.

As a part of the regulatory level analysis, the calculation of the closed-loop equilibrium of a process controlled by constrained MPC is studied. This problem arises in process design and operations, and is often applied within an optimization framework. It is shown that the effect of the control system on the resulting steady-state must be explicitly accounted for, and that in the general case, the use of a steady-state process model only is not sufficient for this calculation to be correctly executed. Two solution strategies, sequential and simultaneous, are presented and evaluated.

The effect of high frequency noise-like disturbances on the two-level control system behavior is analyzed. The analysis which verified by case studies, showed that the LP may have an effect of amplifying the system noise through the bias term which is used for the model update. Such amplification may result in high variation of the LP set points provided to the MPC, thereby degrading the overall performance of the two-level system.

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor Dr. Christopher Swartz for his guidance and encouragement throughout the period of my studies. I am very grateful for his vision and advice, which inspired me and helped me gain insight into the research process.

I also would like to thank Dr. Thomas Marlin and Dr. John MacGregor for the many interesting discussions and for the knowledge they have imparted to me. I consider myself very fortunate to have taken several of their courses.

I wish to put on record my appreciation for the financial support I received from the McMaster Advanced Control Consortium and the McMaster Chemical Engineering Department, without which this research would not have been possible.

I am very grateful to all my fellow students in the Chemical Engineering Department and, in particular, in the Process Control Group for their friendship and support which have made these years brighter and more memorable.

I would like to thank all my friends for their support. I would especially like to thank Galini Gavrilidou, for being such a wonderful person, for her kindness and for the encouragement she gave me during my studies.

And last, but certainly not least, I would like to express a deep gratitude towards my family, especially to my parents Elena and Alexander, and my sister Maria, for their love and care. Their contributions are immeasurable, and I cannot thank them enough for everything they have done for me.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

For a long time process control was considered as a two-layered structure, with the control system above the plant; its main purpose was to bring and maintain plant outputs at specified set points in the presence of measured and unmeasured disturbances. In a situation of strict product specifications, high resource costs and fast-changing market demands, such a structure is not able to work effectively.

Recent progress in control algorithms, numerical calculations for simulation, and optimization allowed this two-layered structure to be extended to a multi-level plant-wide optimization and scheduling system (Qin and Badgwell, 2003; Marlin and Hrymak, 1997). This structure aims to provide optimal operation at different levels of the entire plant. Each level in the automation hierarchy is executed at higher frequency than the level above. The top level of this hierarchy is plant production and scheduling which is executed once in several weeks according to changes in product demand and prices. The next level is real-time optimization (RTO), which comprises several optimizers each appointed to a single unit or a group of units. The optimizers at this level possess sophisticated nonlinear process models and the calculations are executed in terms of hours. The main goal of this level is to provide optimal set points and target values in the presence of degrees of freedom to the regulatory level below. The regulatory level interacts directly with the plant and is responsible for process control. It is executed at a frequency of minutes or even seconds. Such a multilevel structure is able to provide efficient and economically optimal plant operation; it monitors the entire plant and makes required corrections according to observed changes.

Within the past two to three decades Model Predictive Control (MPC) has become the advanced control strategy of choice within the chemical process industry. Key advantages of MPC over the other control algorithms are its ability to accommodate process interactions and dead time directly; and its ability to explicitly handle constraints on manipulated and controlled variables. MPC is a control algorithm in which a dynamic model of the plant is explicitly incorporated. This permits future outputs to be predicted based on a future set of input moves, an estimate of the process state, and predicted disturbances. An optimization problem is solved at every iteration to yield the set of input moves that minimize a (typically quadratic) performance objective. The inputs that correspond to the first time interval are implemented, and the process repeated at the end of the sampling period, with the difference between the measured and predicted plant outputs used to formulate a new disturbance estimate. In this thesis Quadratic Dynamic Matrix Control (Garcia and Morshedi, 1986), using a state-space formulation of the internal dynamic model (Maciejowski, 2002) is considered as the regulatory level control algorithm.

An important issue in the design of the hierarchical control system is integration of its layers, since each of them has different models and frequency of execution. Great care must be given to the integration of RTO and the regulatory level (MPC) because improper design may result in poor operation performance and even instability which maybe unsafe for personnel and equipment.

A linear programming (LP) and quadratic programming (QP) optimizer has been proposed as an integrating link between nonlinear infrequently executed RTO and MPC. Also, a local LP optimizer may be used separately from RTO to accommodate plant degrees of freedom according to a specified economic objective function (Sorensen and Cutler, 1998). Performance of the LP-MPC cascade control system is analyzed in this thesis.

Motivation and Goals

Despite its apparent success, unsatisfactory performance of LP-MPC cascade control systems have been reported. Shah *et al.* (2002) showed that in the presence of a steady-state LP optimizer, the set points could have a large variation in comparison with that of the controlled variables, which means that the LP can degrade MPC performance by sending highly variable set points to the controller.

Economically optimal plant operation usually occurs at the intersection of constraints. If a disturbance enters the plant, the optimum operating point may shift; the LP updates the set points which push the plant to operate at a new set of constraints. Kozub (2002) mentioned that in an LP-MPC control system, the set of manipulated variables at constraints may fluctuate with unexpectedly high frequency. He further noted that it was unlikely that disturbances were responsible for such frequent optimum operating point shifts and there may be a stability issue in such a cascade system.

These observations motivated the present investigation of the effect of the LP level on the two-level control system behavior. The major focus is on control performance with some consideration of the overall system stability. The main goal of the research presented in this thesis is to reveal primary factors which can affect the two-level control system behavior and show scenarios when undesirable behavior may occur.

Main Contributions

Main thesis contributions include the following:

- The performance of LP-MPC cascade control systems was explored under a variety of conditions. This included investigation of the effects of plant/model mismatch, bias update scheme, constraints, control structure, LP objective function and LP execution frequency.

- LP sensitivity to high frequency disturbances was investigated. It was shown that poor two-level control system performance may be caused by output noise and the particular LP design. Three such designs for a 2×2 system were presented and discussed.

- To facilitate simulations in the case of steady-state optimization, dynamic simulation can be substituted by plant steady-state calculation that includes the effects of constrained predictive controllers. It was shown that the use of steady-state models may not be sufficient and dynamic models should be incorporated into the calculation procedure. This thesis presented and evaluated two methods for calculating the closed-loop steady-state of a plant under MPC control in the general case.

Thesis overview

- **Chapter 2 – LP-MPC Cascade Control Systems**

  An introduction to the LP-MPC cascade control system with a literature review on the topic is given. The role and importance of using a local LP optimizer together with the standard MPC controller are highlighted. In this thesis, a state-space formulation of the MPC algorithm and an LP with incorporated economics were chosen as components of the two-level LP-MPC cascade control system. Their formulations are presented in this chapter.

- **Chapter 3 – Stability and Performance of LP-MPC Control Systems**

  The effect of the optimization level (LP) on the regulatory level (MPC) is studied in this chapter and possible factors which may affect closed-loop stability and performance are revealed and discussed. The cascade control system behavior was considered for two methods of the model bias update and their efficiency and performance are presented and compared. The effect of the control structure on the cascade system stability and performance is shown.

- **Chapter 4 – Calculation of Closed-Loop Steady-State of Constrained MPC System**

  This chapter is devoted to the problem of calculation of the closed-loop equilibrium point of a process controlled by constrained MPC. It was shown first, that the steady-state calculation methods which use steady-state models only may not be sufficient in some cases. Then, two approaches, a sequential and simultaneous solution strategy are presented and analyzed. As a case study, the simultaneous solution method was used to study the interaction between LP and MPC levels in an iterative steady-state optimization procedure.

- **Chapter 5 – LP Sensitivity Analysis**

  The effect of high frequency noise-like disturbances on the performance of the cascade system is studied in this chapter. The dependency of a linear programming problem solution on the bias term is presented first. Then, the effect of output noise on the two-level control system behavior in SISO and MISO systems is considered. Finally, sensitivity of a 2×2 cascade control system to nonconstant perturbations is investigated, and possible scenarios of performance degradation in the presence of output white noise are shown and discussed.

- **Chapter 6 – Conclusions and Recommendations**

  This chapter summarizes the main thrust of the thesis, highlighting the major results and achievements. Some recommendations and directions for future work are given.

Chapter 2

LP-MPC Cascade Control Systems

2.1 The Basics of RTO

Progress in digital computers and advances in modelling and optimization algorithms have allowed the application of plant-wide optimization systems, which aim to increase profitability of the plant. Real-time optimization (RTO) is a feedback control system that maximizes a calculated, inferred estimate of the plant profit by adjusting selected optimization variables within specified bounds (Marlin and Hrymak, 1997). The scheme of general RTO is presented in Figure 2.1. It consists of three main components: data reconciliation and update of model parameters, optimization problem solution and results analysis. Since RTO is a closed-loop system, it obtains the data from the plant, and this data is used to update model parameters. The model is a necessary component of any RTO system, since it is used in the optimization process for plant behavior imitation. Since model accuracy plays a crucial role in the optimization process, it is necessary to check that the data from the process coincides with that predicted by the model. If it is not true, the parameters in the model should be updated, to make the model valid. Clearly, such parameters update is possible only at steady-state, since the models used for optimization are usually nonlinear fundamental models (material and energy balances, thermodynamic and physical property equations and others) and they are valid for the stationary state of the plant. This gives

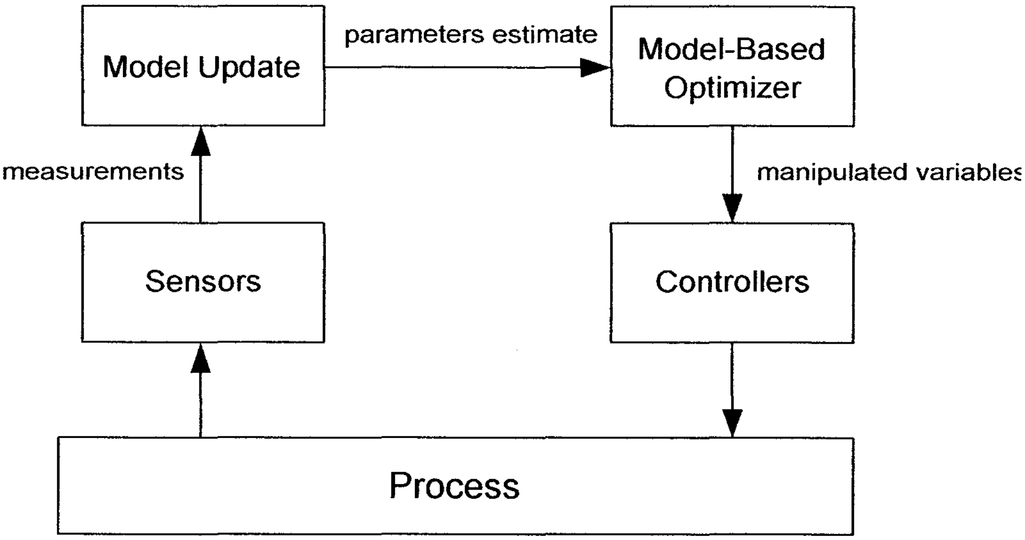


Figure 2.1: Process optimization system structure

rise to the following considerations. First, steady-state detection should be accomplished, which is not easy in the presence of noisy data and poor measurements. Second, active disturbances must be identified, otherwise the updated model will not be valid when these disturbances disappear. Also, it is important to detect and remove gross errors, such as instruments malfunctions, poor sampling, leaks *etc*, which also degrade the model quality.

The optimization procedure employs the updated model to predict the optimum set points and target values. Generally this is a nonlinear optimization problem. The solution of this problem becomes even more difficult when ordinary differential equations (ODE) appear in the model. There exist several methods for solving such problems. Marlin and Hrymak (1997) cite the most commonly used as being augmented Lagrangian methods and reduced sequential quadratic programming. If the control system experiences saturation, the resulting process steady-state would be different than expected. To take the control system effects into account, the model should contain the steady-state effects of the process control system and any routine actions by the operators. It is not a problem for simple controllers which never experience saturation; however it is not straightforward for multivariable controllers with constraints, which can experience saturation effects. The effect of the control system on the steady-state plant behavior is considered in more detail in Chapter 4.

After a solution of the optimization procedure has been found, it must be analyzed to determine whether or not it should be passed down to the control level. For example, it should be checked that the plant is still at the steady-state, that the bounds of the optimization variables are unchanged, i.e. the conditions under which the optimal solution has been found, and that the optimization variables are still available for manipulation. Also, the plant profit improvement should be evaluated and in the case when it is not important, the entire solution can be rejected. It is important to mention that it is inappropriate to apply a part of the solution, because it can lead to poor performance or even infeasible process operation.

As it was mentioned above, the model in RTO is its “heart” and it must meet necessary requirements in order for RTO to work properly. Besides the fact that the model should be accurate, it must be adequate. Model adequacy means that it has such structure, that there exists a set of parameters (calculated operating conditions), which will coincide with the true plant optimal operating conditions. It does not mean that the calculated parameters provide true plant optimum, but it means that the model profit curve has a maximum at the same parameter values as the true plant profit curve. Since the RTO calculates the operating conditions which maximize the profit, adequacy should be guaranteed at the optimum point, while beyond this point adequacy may not be preserved. This definition of model adequacy is called “point-wise” model adequacy and is well studied in Forbes and Marlin (1994a) (Forbes and Marlin, 1994b). The main requirement here is that for the RTO model based optimizer to yield the plant optimum, the optimality conditions for the model must be valid at the optimum plant conditions. The model is adequate if there exists a set of adjustable parameters, such that the statement above is preserved.

2.1.1 Main approaches for model-based RTO

In the previous section is was shown that the general RTO consists of two main levels: a lower level, comprised of a plant under a control system and an upper level, comprised of a model update block and an optimizer. There can be more optimization levels above this two-level system, which would range in objectives, type of models used, frequency of implementation and so forth. The problem of coordination of these levels is one of the most challenging problems in the RTO research field. The difficulties here arise, since the levels employ different models for the calculation of optimal values, and each of these models is not perfect and must be regularly updated.

According to the types of objective function formulation, the methods of the model update, and the model type used, the following groups of RTO can be classified (Marlin and Mudt, 2004) (here and further, Model Predictive Control (MPC) is assumed as the control system):

1. RTO is an LP(QP) optimization problem on top of MPC. This optimization level does not optimize the plant economics directly and is used to select the least costly manipulated variables (MVs). It uses the bias model update. The advantage of this approach is that it is simple in implementation, fast and does not cost much to install. The disadvantage is that it optimizes not the plant profit, but a surrogate variable (feed rate, conversion, *etc*)

2. RTO is an LP(QP) economic optimization problem on top of MPC. Coefficients in the optimization level are updated from the nonlinear model and a bias update is also employed. The optimum is usually at the corner point of the feasible region. Vermeer and Pederson (1996) give a blending process example. Since the parameters in the optimizer level are changed, the gains in the MPC controller should be changed also, which can affect the performance. This approach is also comparably cheap, since it requires for operation a nonlinear model obtained once.

3. Nonlinear open-loop RTO. Here, RTO is represented by a nonlinear optimization problem which is not updated empirically. This RTO is placed on the top of MPC, which has its own local optimizer. Such an RTO scheme can optimize the set points for more than one MPC and if so, it is sometimes referred to as a "composite" level. The reported application of this method (Verne *et al.*, 1999) uses Sequential Linear

Programming (SLP).

4. Full closed-loop RTO. RTO employs a nonlinear model for optimization and is placed above MPC which has its own local optimizer. Also, RTO provides parameter estimation using plant data. An example given in the literature is optimization of a hydrocracker complex (Pederson *et al.*, 1995).

This thesis considers primarily the first type of RTO, i.e. the RTO with a linear programming problem and bias model update.

2.2 Composite LP

The composite LP application usually contains several MPC's, each having a local set of constraints. This LP is responsible for managing the distribution of plant-wide constraints among these advanced process controllers (APCs).

The application of this technique was presented in Jakhete *et al.* (1999), where the authors consider a composite LP optimizer which was designed and implemented to work with both FCCU (fluid cat cracking unit) and GPU (gas processing unit) multivariable predictive controllers at Sunoco's Toledo refinery. Placing one LP optimizer on the top of both units controllers allows all the process operation constraints to be considered simultaneously at the same frequency as the multivariable controllers. If units form a sequence in their operation, then a key requirement of the composite LP (CLP) is that independent variables in a downstream controller should not affect dependant variables in the upstream controller. The advantage of the composite LP is that it is able to optimize the production across both units and, therefore, there is no need to solve the problem of monitoring and control of intermediate stream variables. In this project, it was designed that the GPU controller handles its own constraints first, until all of its degrees of freedom are exhausted; the CLP then directs the FCCU controller to start cutting feed severity. The main objective of CLP here was to maximize feed or keep feed at or near maximum target rate, while meeting all constraints in the FCCU and GPU. The authors report that for this project CLP normally does its job and decreases the feed from its maximum only to ensure that all constraints are satisfied.

Friedman (2000) mentioned that economically speaking, CLP is not a proper optimization technique because it relies on approximate linear models rather than rigorous ones. Also, as it was mentioned above, many industrial MPC's are already equipped with LP optimizer and therefore composite LP is just an extension of local LP. However, this approach allows the steady-state anticipation to be avoided and to implement set point changes in small steps. Also, the use of composite LP gives the opportunity to work with reasonable size MPC's and still keep a global view of the plant.

The Honeywell solutions, such as Profit Controller and Profit Optimizer can be considered as applications of the composite LP. Indeed, the Process Optimizer contains the model of entire plant. At each controller iteration, the optimizer obtains the plant data and this frequent process feedback compensates for the plant/model mismatch. Nath *et al.* (1999) describes their algorithm in the following way: since the Optimizer and controller are implemented at about the same frequencies, the plant is driven in small steps toward the optimum. Also, there is no need to implement a local optimizer at the MPC level, since the technology embeds so-called "bridge models" which relate interaction between multiple Profit Controllers.

2.3 LP-MPC Integration

In modern processing plants the MPC controller is a part of a multi-level hierarchy of control functions (Qin and Badgwell, 2003). At the top of this structure a plant-wide optimizer determines optimal steady-state settings for each unit in the plant. The optimizer computes an economic steady-state and passes this to the dynamic constraint control system for implementation. In many practical applications, the number of inputs is not equal to the number of outputs and some techniques for the management of extra degrees of freedom should be applied. One approach is to use as many manipulated variables for control as it is necessary while driving the remaining to specified target values. It can be achieved posing the controller objective function in the following way:

$$J_k = ||\hat{y}_k - y^{set}||_Q^2 + ||\hat{u}_k - u^{tar}||_R^2 + ||\Delta\hat{u}_k||_S^2 \tag{2.1}$$

where Q and R are symmetric positive definite penalty matrices, and S is a symmetric positive semidefinite matrix with the norms in (2.1) defined as $||x||_Q^2 = x^T Qx$. $\hat{y}_k$, $\hat{u}_k$ and $\Delta\hat{u}_k$ represent predicted outputs, future inputs and future input changes respectively over specified horizons and are related through a dynamic process model.

Assume that the decisions about control structure (which manipulated variables have targets and which are used for control) are made, and that set points and target values are given. The following potential problems arise:

- Since constrained predictive control is used, it is very possible that some of the manipulated variables will encounter constraints. In this case degrees of freedom are lost and it is no longer possible to track or keep all controlled variables at their set points, and non-zero offset will appear.
- Output set points and input target values relate to each other through the steady-state process model. Assume that initially process is at the steady-state and no unmeasured or measured disturbances affect the plant. Then the inputs and the outputs of the plant satisfy the equation:

  $$y_{ss} = K_m u_{ss} \tag{2.2}$$

  and the set points and target values in objective function (2.1) satisfy this equation. If suddenly output unmeasured disturbance (for simplicity, step disturbance) d affects the plant, the steady-state parameters will shift and the new steady-state equation is:

  $$y_{ss}^* = K_m u_{ss}^* + d \tag{2.3}$$

Comparing (2.2) and (2.3) it is obvious that either $y_{ss} = y^*_{ss}$ or $u_{ss} = u^*_{ss}$ but never at the same time. Therefore, it is not possible to achieve the set points for outputs and target values for inputs at the same time if they were calculated without knowledge of disturbance but it does appear during operation. This again leads to non-zero offset.

- If the set points y^{set} and target values u^{tar} are given from upper optimization level then they satisfy the steady-state model used at that level. Since the upper optimization levels use more sophisticated (often nonlinear) models these values may not be consistent with controller steady-state model i.e.:

$$y^{set} \neq K_m u^{tar}$$

Because of these potential problems and some others, an MPC controller with objective function (2.1) should not be implemented in practice without some modification.

A local steady-state optimizer for MPC systems was proposed by many researchers as a remedy to avoid possible problems which can appear during control of excessive inputs (Yousfi and Tournier, 1991; Brosilow and Zhao, 1988; Muske, 1997) or as a link between optimization and regulatory levels (Ying and Joseph (1999)). Its role is not to substitute the unit optimizer but to coordinate the interaction between these levels.

Such coordination first of all has **supervisory** purposes. This means that the local optimizer must guarantee operability of the MPC in the presence of disturbances and set point changes. Supervisory properties at the upper level must be present in any non-square control systems. Among the supervisory "skills" of the controller are: monitoring of available degrees of freedom and checking if any of control variables are lost (for example, saturated), making decisions about involving new variables into control process (for instance, based on analysis of plant directionality or interaction). However, in the presence of degrees of freedom, multiple choice is possible. Then, a decision can be based not only on the plant characteristics but also on the operation economics i.e. using cheaper resources for control while keeping the consumption of more expensive resources at a low level. Therefore, the upper level can also have **economic** objectives. It can be expressed in a form of an optimization problem (usually LP or QP) and solved at different frequencies of MPC implementation. Since different formulations of supervisory and economic designs are possible, the two stage LP-MPC cascade control system can have different modifications.

This thesis considers an LP with economics only. It is assumed that the sets of MVs which are used for optimization and for control do not change during simulations. The upper level makes decisions which are based only on the optimization problem solution and no other analysis is done then. The inclusion of the supervisory objectives into local optimizer has not been studied here; these are issues for future research.

Soufian and Sandoz (1996) considered the use of a multivariable controller together with an LP steady-state optimizer for distillation process control. The linear programming optimizer is used for on-line evaluation of the economic optimum operating point. This runs at every controller execution time to provide a consistent optimum steady-state solution within the specified limits of the actuators and the measured variables. The optimization is carried out at a steady-state operating point using steady-state models. Calculated controlled variable targets are transferred to the set points of the controller. The objective function here represents the economics of the process and is a linear combination of inputs and outputs. The simulation in this paper was implemented using incremental and absolute LP formulations. The incremental LP optimizer uses the long range of predicted values of the MVs and the actuator values to update its reference point at each control interval. This allows unconsidered disturbances to be taken into account. The absolute LP optimizer does not update its reference steady-state level, i.e. it is implemented at steady-states only. The comparison between the absolute and incremental LP formulations was done and the authors concluded that in terms of the quality of the composition control, the absolute LP optimizer is as good as the incremental form, but it is more expensive to operate since it uses more resources to achieve the same specification.

Shah *et al.* (2002) considered the performance of an MPC controller with a steady-state LP optimizer. It was found here, that the use of such an optimizer can lead to performance degradation. Among the reasons for such degradation the following were mentioned: bad models, inadequately designed LP in that the LP operates at the control frequencies, inappropriate choice of weightings, ill-posed constraints and steady-state bias updates. Several techniques for controller performance diagnosis and its improvement are also presented here.

Sorensen and Cutler (1998) mention that embedding a linear program (LP) in dynamic matrix control brings many powerful new features to the algorithm. Using the equal concern errors, the LP provides a feasible steady-state solution to the controller, which they report is necessary for the controller stability. However, the authors define the stability of the controller as the ability of the LP level to provide a feasible steady-state solution that satisfies all constraints. The definition does not consider the behavior of the process outputs directly.

In many cases the steady-state gains between the manipulated variables (independent LP costs) and the controlled variables (dependent LP costs) can be calculated from the plant tests through identification procedures. Generally, the steady-state gain information identified from plant test data is sufficient to determine the cost factors in the LP. However, in the case when product yield and property data are not available from the plant test, an off-line engineering model can be used instead. Unfortunately, the independent variables used in the engineering model are usually not consistent with the manipulated variables in the controller. Sorensen and Cutler (1998) propose two methods to rearrange the variables to make them consistent. The first method uses a partial inversion technique that is done with basic algebra. The second method uses matrix algebra. Also, the authors devote some attention to the robustness and stability of such two-level controllers. They report that many stability problems for general multivariable controllers come from the inconsistency between the target values for the controller and what is actually obtained at the steady-state. Here, in the presence of optimizer, if an unmeasured disturbance enters the system, the update of the prediction vector is able to pick up the shift and change the predicted steady-state. The authors report that the controller is inherently stable since the LP provides a feasible set of targets and the dynamic part of the controller calculates the trajectory the process will follow in sufficient amount of steps. However, this statement is not substantiated by any proof and, as it was mentioned above, the stability definition is inadequate.

Optimization of a Residue Cracking Unit (RCU) at the BP refinery at Kwinana, Western Australia is presented in Hall and Verne (1993). The uniqueness of this project is that the controller was placed within a distributed control system without a separate host computer. This project consisted of seven manipulated variables and 22 controlled variables. The role of optimizer was to find such constraints of the controlled variables (CVs), which would guarantee safe operation as well as maximized profit. In 1980, an advisory optimizer was installed at the refinery, developed by BP Central Engineering. The optimizer used nonlinear process models, which were initially fitted to the observed plant data. However this optimizer was not able to handle the complex control issues, which precluded good unit operation performance. In 1989, after a plant inspection, a multivariable controller was recommended that would improve control performance by regulating all the MVs while considering the complex dynamics and variable interactions. To prioritize the constraints, it was required that an optimizer be integrated with the controller. The user-defined objective function could be any linear or/and quadratic combination of manipulated and controlled variables. By using prices as coefficients, the objective function has been set up to optimize the added value by the RCU.

Two main problems appeared here. First, direct product rates measured off the fractionator had too much dead time and noise to be used directly as measured variables. To overcome this, the production was predicted based on the cracking severity. Second, the product value is dependent on the product utilization for the tail gas, C_3 and C_4 products. To solve this problem, a special pricing technique was used to appropriately value production. An external program then modified the user objective function coefficients on-line according to the predicted steady-state production. Then the optimizer converges to the optimal solution using successive linear programming.

Even though the multivariable controller improved the unit performance, the use of linear controller models limited its ability to achieve a true plant optimum. Consequently, a method for extending the degree of optimization beyond what can be achieved with the multivariable predictive controller alone was developed and described in Verne *et al.* (1999). A proprietary cracking model was integrated with the controller to calculate optimum solutions. There were 11 MVs and 30 potential constraints. The controller contained process models, which provide steady-state and dynamic information about the process. These models had two purposes. The first was to achieve effective regulatory control and, the second to determine the best combination of limits that the unit should run against, the combination which corresponds to the maximum profit. This constraints combination is typically calculated by using a linear programming (LP) problem. The objective function of such LP is as follows:

$$\$rcu = \sum_{i=1}^{m} a_i \cdot I_i + \sum_{j=1}^{n} b_j \cdot J_i$$

where a_i is the value of the i th independent variable, I_i, and b_j is the value of the j th dependent variable, J_j.

The aim of LP is to effectively push the plant operation to the most profitable constraints. The productivity of this LP depends on the models it uses and, therefore, the predicted values need to be verified with the observed parameters and corrected if necessary. The important assumptions are that the model gains are at least the correct sign and that the relative magnitude between gains is correct.

A study on LP (QP) - MPC cascade control is described in Ying and Joseph (1999). They consider a two stage optimizer presented in Figure 2.2. The second level is a local MPC optimizer and the top level is an RTO system, containing a nonlinear model of the plant. The local optimizer uses the disturbance estimate, obtained from MPC at every sampling time, and determines the set points and target values for MPC. In the absence of a local optimizer, the controller set points are calculated from RTO, which uses an economic objective function. According to the authors, the RTO infrequently updates the optimal nominal "targets" y^* and u^* and the cost parameters guiding the LP (QP). It can also update the

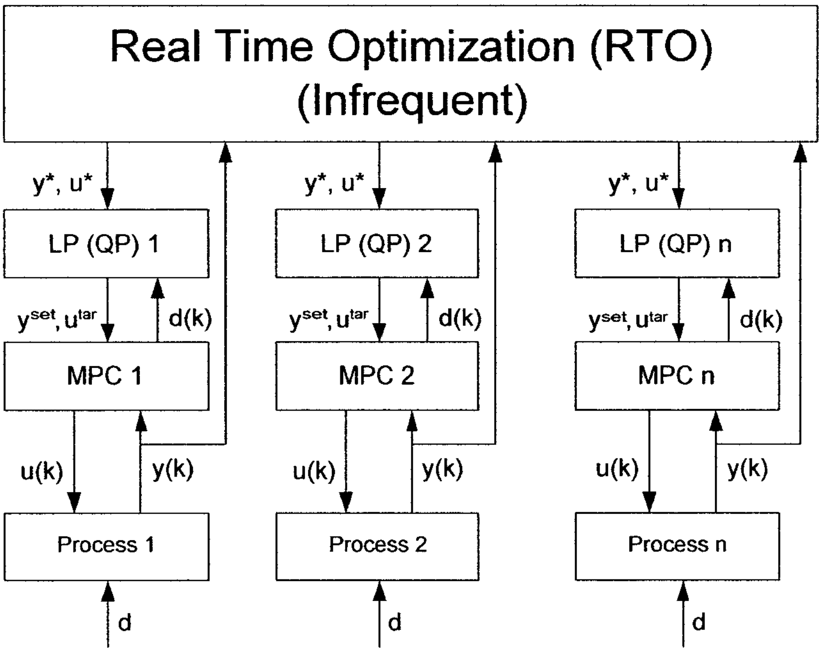


Figure 2.2: Two-stage RTO system hierarchy

constraints for the LP (QP) if necessary. The RTO is based on nonlinear steady-state models. The LP (QP) is executed at the same frequency as the lower stage MPC. The objective function in the LP (QP) level does not contain any economics, but just penalizes the deviation of the controller set points around nominal "targets" y^* and u^* from RTO. The authors cite an LP of the form:

$$\min_{y^{set},u^{tar}} c_y^T(y^{set} - y^*) + c_u^T(u^{tar} - u^*) + c_\epsilon^T \epsilon$$

subject to:

$$y^{set} = A_s u^{tar} + d(k)$$

$$d(k) = d(k-1) + \Delta(k)$$

$$y^{min} - \epsilon \le y^{set} \le y^{max} + \epsilon$$

$$u^{min} \le u^{tar} \le u^{max}$$

$$A_s = C(I - A)^{-1}B$$

$$\epsilon \ge 0$$

where $d(k)$ is the estimated disturbance at time k, $\Delta(k)$ is the estimated disturbance change which is the difference between measurement $y(k)$ and predicted output $y(k|k-1)$, c_y and c_u are cost parameters, ϵ is used to guarantee a feasible solution to the LP, and c_ϵ^T is a tuning parameter. This LP formulation is a linear approximation of the RTO objective function. A quadratic objective may alternatively be used. They provide three stability theorems on the cascade LP(QP)-MPC stability for the case when the model in the LP and MPC are perfect. From this assumption it follows that the estimate disturbance at any time will be exact. This assumption would be unrealistic in practice because of unavoidable process uncertainty. The authors remark that in the case when an LP is used as the optimization problem, a disturbance entering the process may result in a jump from one intersection of constraints to another. This would cause the set points to change in an abrupt manner which is detrimental to the stability of the MPC. A case study with plant/model mismatch was considered in their paper, and even though an instability effect was not shown, the authors mention that the proposed stability theorems do not apply there.

2.4 LP with Incorporated Economics

If the local optimizer is used separately from the upper level RTO then it can also be based on the economics of the plant operation. Many researchers (Moro and Odloak, 1995; Ramos *et al.*, 2002) point out that the optimal conditions of the operation appear at the intersection of constraints. Also, in many cases, the controlled variables do not have exact set points but upper limits (for example, impurity or pressure control) or lower limits (temperature or level control). Therefore, employing some economics into local optimizer, the plant can be pushed to operate at such a set of constraints that is economically optimal according to the specified objective function.

One of the possible formulations was proposed by Sorensen and Cutler (1998) and can be described in the following way. A local optimizer is placed on the top of the regulatory control level and executed at the same frequency. Using the current values of the manipulated variables, the LP calculates predicted steady-state values and then based on these calculates new set points for the controller by solving an optimization problem. Since optimization is performed using the predicted values, waiting for steady-state can be avoided. Assume that the steady-state process model is:

$$y^{ss} = A^{ss}u^{ss} + d$$

If the manipulated variables are changed by Δu then the outputs will change by:

$$\begin{aligned} u &= u^{ss} + \Delta u \\ y &= A^{ss}u + d = A^{ss}\left(u^{ss} + \Delta u\right) + d \\ &= y^{ss} + A^{ss}\Delta u \end{aligned}$$

If the variables are presented in deviation form ($y = y^{ss} + \Delta y$ and $u = u^{ss} + \Delta u$) and disturbance d is constant at the steady-state than changes in outputs will be proportional to the changes in inputs and the coefficient of proportionality is just the steady-state gain:

$$\Delta y = A^{ss}\Delta u$$

The LP objective function was proposed as the plant profit function:

$$\begin{aligned} \text{Profit} = \sum_i \text{Product}_i \cdot \text{Pvalue}_i \quad &- \quad \sum_j \text{Feedstock}_j \cdot \text{Fcost}_j - \\ &- \quad \sum_k \text{Utility}_k \cdot \text{Ucost}_k \end{aligned}$$

where: Profit = Plant profit function ($\$/day$); Product_i = Product flow rate, 'i' (quantity/day); Pvalue_i = Product value, 'i' (\$/quantity); Feedstock_j = Feedstock flow rate, 'j' (quantity/day); Fcost_j = Feedstock cost, 'j' (\$/quantity); Utility_k = Utility usage, 'k' (quantity/day); Ucost_k = Utility cost, 'k' (\$/quantity).

Some researchers (Soufian and Sandoz, 1996; Moro and Odloak, 1995) proposed the use of a linear combination of manipulated and controlled variables at the predicted steady-state as the objective function such that it can be configured to economize the use of particular manipulated variables and increase production of specific outputs.

The formulation of the LP which was used in the thesis is as follows:

$$
\begin{aligned}
\min_{u_i^{tar}, y_j^{set}} \quad & \sum_{i=1}^{p} a_i y_i^{set} + \sum_{i=1}^{m} b_i u_i^{tar} \\
\text{subject to:} \quad & \\
& y^{set} = K_m u^{tar} + d \\
& y^{min} \leq y^{set} \leq y^{max} \\
& u^{min} \leq u^{tar} \leq u^{max}
\end{aligned}
\tag{2.4}
$$

where: K_m is the model steady-state gain; a_i and b_j are profit and cost coefficients respectively; d is the LP bias.

In all two-level LP-MPC cascade control system simulations throughout the thesis, it was assumed that the model steady-state gain matrix at the LP level is the same as the steady-state matrix of the model which the controller uses for calculations, and that the constraints on inputs and outputs at both levels are identical unless noted otherwise. All simulations presented in this thesis were run with hard constraints on the outputs at both levels which sometimes may result in infeasibility of the corresponding optimization problems under certain circumstances. This may be avoided by the use of soft output constraints, but since infeasibility problems were not encountered in the case studies conducted in this thesis, this formulation was not considered.

2.5 State-Space Formulation of Model Predictive Control

Nowadays, Model Predictive Control (MPC) enjoys wide industrial application and significant interest from academia. This control algorithm belongs to the class of Model Based controllers, where the plant model is explicitly employed in the control calculation. The main advantages of MPC are its natural ability to regulate multivariable plants and handle different types of constraints directly.

MPC arose from the research by Richalet *et al.* (1978) and Cutler and Ramaker (1979), which led to commercialized control packages Identification and Command (IDCOM) and Dynamic Model Control (DMC) respectively. These control algorithms represent the plant by impulse response and step response models respectively. Since these models were chosen, integrating processes require special consideration. Also, treating the discrepancies between model predictions and true output values as a constant output disturbance may result in poor performance in some cases.

Taking these shortcomings into account, the use of state-space models within the MPC algorithm was proposed (Li *et al.*, 1989; Ricker, 1990; Yu *et al.*, 1994). MPC with state-space plant models is able to operate not only with linear stable but also linear unstable and integrating processes. Besides the constant output disturbances, the algorithm can handle input and state disturbances of various nature. Also, the state-space approach has been well studied in the linear optimal feedback control framework with ample theoretical results, and its combination with MPC can bring many potential benefits (Marquis and Broustail, 1998).

2.5.1 Models

The controller employs the process model together with estimations of the states and disturbances for future plant output predictions. The controller calculates the sequence of inputs $u(k+i)$ such that the predicted output trajectories are optimal according to a specified criterion. This model is a part of the control system and its parameters are specified.

Let the internal model have the following linear, time-invariant, state-space form:

$$\begin{aligned} x^m(k+1) &= Ax^m(k) + Bu(k) \\ y^m(k) &= Cx^m(k) + Du(k) \\ x^m(0) &= x_0^m \end{aligned} \tag{2.5}$$

where k is the sampling instant, $x^m \in \Re^n$ is a vector of model states, $u \in \Re^m$ is a vector of plant inputs, $y^m \in \Re^p$ is a vector of model outputs, $A \in \Re^{n\times n}$, $B \in \Re^{n\times m}$, $C \in \Re^{p\times n}$ and $D \in \Re^{n\times m}$ are constant matrices. The model steady-state gain matrix can be found from (2.5) using the following equation:

$$K_m = C(I - A)^{-1}B + D \tag{2.6}$$

Real plant dynamics are never known exactly. For simulation purposes the plant is modelled as a linear state-space model which may be different from the controller model:

$$\begin{aligned} x(k+1) &= A_p x(k) + B_p u(k) \\ y(k) &= C_p x(k) + D_p u(k) \\ x(0) &= x_0 \end{aligned} \tag{2.7}$$

where $x \in \Re^{n_p}$ is a vector of plant states, $u \in \Re^m$ is a vector of plant inputs and $y \in \Re^p$ is a vector of measured outputs. $A_p \in \Re^{n_p\times n_p}$, $B_p \in \Re^{n_p\times m}$, $C_p \in \Re^{p\times n_p}$ and $D_p \in \Re^{n_p\times m}$ are constant matrices.

Analogously to (2.6), process steady-state gain matrix can be calculated from (2.7) via the following equation:

$$K_p = C_p(I - A_p)^{-1}B_p + D_p \tag{2.8}$$

2.5.2 State and disturbance estimation

In order to calculate future output predictions, the current state and disturbances need to be estimated.

State estimation techniques are well known because of their wide application in Linear Quadratic Gaussian (LQG) control (Åström and Wittenmark, 1990; Kwakernaak and Sivan, 1972). The theory of the state estimation leads to the concept of the observer - an element of the control system which provides the state estimation and the state predictions. There are two mainstreams in state estimation which are the stochastic approach with Kalman filtering and the deterministic approach using the Luenberger observer.

State estimation is related to unmeasured disturbance modelling. Since the exact value of the disturbance is never known, it is only possible to describe its nature, or its type of behavior. For quantitative analysis it is assumed that the process is considered at time instant k and the following parameters are known: $y(k)$ - current plant output; $u(k-1)$ - last implemented input; $\hat{x}(k|k-1)$ - the estimation of the current state, which has been made at the previous step $k-1$.

Although there are several approaches for unmeasured disturbance modelling and estimation, this thesis employs the method which is implemented in the standard DMC and QDMC algorithms (Cutler and Ramaker, 1979; Garcia and Morshedi, 1986). Here, the disturbance is modelled as a step output disturbance, the size of which is equal to the difference between measured and predicted output values. The state estimate used in the calculation of the predicted output is based on the controller model (2.5) and performed in the following manner:

$$\begin{aligned}\hat{x}(k|k) &= \hat{x}(k|k-1)\\ \hat{x}(k|k-1) &= A\hat{x}(k-1|k-1) + Bu(k-1)\\ \hat{d}(k|k) &= y - \hat{y}(k|k-1)\end{aligned}$$

2.5.3 Prediction calculation

Output predictions can be calculated using the state and disturbance estimations. In MPC with the state-space formulation, the predictions of the output trajectories are calculated using the internal model (2.5) for P steps ahead expressed in terms of M control moves. P is called "prediction horizon" and it shows how many steps ahead the future plant behavior is considered and the plant outputs are predicted, and M is the "control move horizon" and it shows how many control moves ahead are calculated. Generally, P is much larger than M and they both are tuning parameters of the controller.

Since the DMC disturbance estimation scheme is used, it is assumed that the future disturbance predictions are equal to current disturbance estimation:

$$\hat{d}(k+i|k) = \hat{d}(k|k) \quad i = 1, ..., P$$

Output predictions can be calculated using the internal model (2.5):

$$\begin{aligned} \hat{x}(k+i|k) &= A\hat{x}(k+i-1|k) + B\hat{u}(k+i-1|k) \\ \hat{y}(k+i|k) &= C\hat{x}(k+i|k) + D\hat{u}(k+i|k) + \hat{d}(k+i|k) \end{aligned} \tag{2.9}$$

where $\hat{y}(k+i|k) \in R^p$ represents the predicted values of the outputs at time step $k+i$, based on information available at time step k.

2.5.4 Optimization problem formulation

Since the main purpose of the control is to track the output trajectory as close as possible to the reference trajectory, an optimization problem can be formulated, where the optimization variables are unknown control moves. In the case when extra inputs are available, the same objective function can penalize their deviation from the target values given from the optimization level.

At each sampling time the controller solves an optimization problem with the following objective function:

$$\begin{aligned} \min J_k = \; & \sum_{i=1}^{P} ||\hat{y}(k+i|k) - y^{set}||^2_{Q_i} + \sum_{i=0}^{N-1} ||\hat{u}(k+i|k) - u^{tar}||^2_{R_i} + \\ & \sum_{i=0}^{M-1} ||\Delta\hat{u}(k+i|k)||^2_{S_i} \end{aligned} \tag{2.10}$$

Here, N is a horizon over which the future inputs are penalized from their target values; Q_i and R_i are symmetric positive definite penalty matrices; S_i is a symmetric positive semidefinite matrix. Such matrices make the objective function convex. Even though it is possible to use different penalties at different sampling instants, usually these matrices are constant over the horizons.

Vector Δu represents the change in manipulated input, defined as:

$$\begin{aligned} \Delta\hat{u}(k|k) &= \hat{u}(k|k) - u(k-1) \\ \Delta\hat{u}(k+i|k) &= \hat{u}(k+i|k) - \hat{u}(k+i-1|k),\ i = 1, ..., M-1 \end{aligned} \tag{2.11}$$

To regulate the smoothness of output responses, reference trajectories can be used instead of fixed set points. These trajectories may have different rates of approaching the set points with which they coincide at steady-state. In all simulations presented in the thesis, reference trajectories have not been used and actual set points were used in the objective function.

Since the controller solves the optimization problem at every iteration, the upper and lower bounds of the controlled and manipulated variables can be taken into account explicitly. There are three sets of constraints: output constraints, input constraints and constraints on the input rates of change. The optimization problem searches for such solution that all future output predictions $\hat{y}(k+i|k)$, future inputs $\hat{u}(k+i|k)$ and future input moves $\Delta\hat{u}(k+i|k)$ do not exceed their limits over the corresponding horizons.

$$\begin{aligned} y^{min} \leq &\quad \hat{y}(k+i|k) &\leq y^{max}, \quad i = 1, ..., P \\ u^{min} \leq &\quad \hat{u}(k+i|k) &\leq u^{max}, \quad i = 0, ..., M-1 \\ \Delta u^{min} \leq &\quad \Delta\hat{u}(k+i|k) &\leq \Delta u^{max}, \quad i = 0, ..., M-1 \end{aligned} \tag{2.12}$$

The optimization problem which the controller solves at every iteration is to minimize objective function (2.10) subject to equalities (2.9) and inequalities (2.12):

$$\begin{aligned}
\min J_k = \quad & \sum_{i=1}^{P} ||\hat{y}(k+i|k) - y^{set}||^2_{Q_i} + \sum_{i=0}^{N-1} ||\hat{u}(k+i|k) - u^{tar}||^2_{R_i} + \\
& \sum_{i=0}^{M-1} ||\Delta\hat{u}(k+i|k)||^2_{S_i}
\end{aligned}$$

subject to:

$$\begin{aligned}
\hat{x}(k+i|k) &= A\hat{x}(k+i-1|k) + B\hat{u}(k+i-1|k), \quad i = 1, ..., M \\
\hat{x}(k+i|k) &= A\hat{x}(k+i-1|k) + B\hat{u}(k+M-1|k), \quad i = M+1, ..., P \\
\hat{y}(k+i|k) &= C\hat{x}(k+i|k) + D\hat{u}(k+i|k) + \hat{d}(k+i|k), \quad i = 1, ..., P \\
\Delta\hat{u}(k|k) &= \hat{u}(k|k) - u(k-1) \\
\Delta\hat{u}(k+i|k) &= \hat{u}(k+i|k) - \hat{u}(k+i-1|k), \ i = 1, ..., M-1 \\
\hat{u}(k+i|k) &= \hat{u}(k+M-1|k), \quad i = M, ..., N-1 \\
y^{min} &\leq \hat{y}(k+i|k) \leq y^{max}, \quad i = 1, ..., P \\
u^{min} &\leq \hat{u}(k+i|k) \leq u^{max}, \quad i = 0, ..., M-1 \\
\Delta u^{min} &\leq \Delta\hat{u}(k+i|k) \leq \Delta u^{max}, \quad i = 0, ..., M-1
\end{aligned} \tag{2.13}$$

The solution of the optimization problem is the sequence of M future control moves. However, only the first input move will be implemented and at the next iteration, the entire calculation procedure is repeated.

2.5.5 Compact formulation

We present here a formulation of MPC optimization problem expressed in terms of the manipulated variable changes, and in which the predicted states are eliminated. Equation (2.11) can be used to express the inputs in terms of input changes as

$$\hat{u}(k+i|k) = u(k-1) + \sum_{j=0}^{i} \Delta u(k+j|k)$$

Also, since only M first control actions are considered:

$$\Delta\hat{u}(k+i|k) = 0, \quad i = M, ..., P$$

Then, using internal model (2.5), the following prediction equations can be written:

$$\begin{aligned}
\hat{x}(k+1|k) &= A\hat{x}(k|k) + B\hat{u}(k|k) \\
\hat{x}(k+2|k) &= A\hat{x}(k+1|k) + B\hat{u}(k+1|k) = A^2\hat{x}(k|k) + AB\hat{u}(k|k) + B\hat{u}(k+1|k) \\
\hat{x}(k+3|k) &= A\hat{x}(k+2|k) + B\hat{u}(k+2|k) = \\
&= A^3\hat{x}(k|k) + A^2B\hat{u}(k|k) + AB\hat{u}(k+1|k) + B\hat{u}(k+2|k) \\
&\dots \\
\hat{x}(k+M|k) &= A^M\hat{x}(k|k) + A^{M-1}B\hat{u}(k|k) + \ldots + AB\hat{u}(k+M-2|k) + B\hat{u}(k+M-1|k) \\
&\dots \\
\hat{x}(k+P|k) &= A\hat{x}(k+P-1|k) + B\hat{u}(k+P-1|k) = \\
&= A^P\hat{x}(k|k) + A^{P-1}B\hat{u}(k|k) + A^{P-2}B\hat{u}(k+1|k) + \ldots + \sum_{i=0}^{P-M} A^iB\hat{u}(k+M-1|k)
\end{aligned} \tag{2.14}$$

This can be written in matrix-vector form:

$$\begin{pmatrix} \hat{x}(k+1|k) \\ \hat{x}(k+2|k) \\ \vdots \\ \hat{x}(k+M|k) \\ \vdots \\ \hat{x}(k+P|k) \end{pmatrix} = \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^M \\ \vdots \\ A^P \end{pmatrix} \hat{x}(k|k) + \tag{2.15}$$

$$+ \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ A^{M-1}B & A^{M-2}B & \cdots & B \\ \vdots & \vdots & \cdots & \vdots \\ A^{P-1}B & A^{P-2}B & \cdots & \sum_{i=0}^{P-M} A^iB \end{bmatrix} \begin{pmatrix} \hat{u}(k|k) \\ \hat{u}(k+1|k) \\ \vdots \\ \hat{u}(k+M-1|k) \end{pmatrix}$$

According to (2.11), the vector of future inputs can be expressed as a linear combination of last implemented input and future control moves:

$$\begin{pmatrix} \hat{u}(k|k) \\ \hat{u}(k+1|k) \\ \vdots \\ \hat{u}(k+M-1|k) \end{pmatrix} = \begin{pmatrix} I \\ I \\ \vdots \\ I \end{pmatrix} u(k-1) + \begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ I & I & \cdots & I \end{bmatrix} \begin{pmatrix} \Delta\hat{u}(k|k) \\ \Delta\hat{u}(k+1|k) \\ \vdots \\ \Delta\hat{u}(k+M-1|k) \end{pmatrix} \tag{2.16}$$

where I is the identity matrix of size m. The same equation in compact form is:

$$\hat{u}_k = D_u^* u(k-1) + E^* \Delta\hat{u}_k \tag{2.17}$$

Substitution of equation (2.16) into (2.15) produces the following expression for the future state predictions:

$$\hat{x}_k = A^* \hat{x}(k|k) + B^* u(k-1) + C^* \Delta\hat{u}_k \tag{2.18}$$

where:

$$\hat{x}_k = \begin{pmatrix} \hat{x}(k+1|k) \\ \hat{x}(k+2|k) \\ \vdots \\ \hat{x}(k+M|k) \\ \vdots \\ \hat{x}(k+P|k) \end{pmatrix}; \ \Delta\hat{u}_k = \begin{pmatrix} \Delta\hat{u}(k|k) \\ \vdots \\ \Delta\hat{u}(k+M-1|k) \end{pmatrix}; \ A^* = \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^M \\ \vdots \\ A^P \end{pmatrix}$$

$$B^* = \begin{pmatrix} B \\ AB+B \\ \vdots \\ \sum_{i=0}^{M-1} A^i B \\ \vdots \\ \sum_{i=0}^{P-1} A^i B \end{pmatrix}; \ C^* = \begin{bmatrix} B & \cdots & 0 \\ \vdots & \cdots & \vdots \\ \sum_{i=0}^{M-1} A^i B & \cdots & B \\ \vdots & \cdots & \vdots \\ \sum_{i=0}^{M} A^i B & \cdots & AB+B \\ \vdots & \cdots & \vdots \\ \sum_{i=0}^{P-1} A^i B & \cdots & \sum_{i=0}^{P-M} A^i B \end{bmatrix}$$

If P is the prediction horizon then the vector of output predictions in matrix-vector form is:

$$\begin{pmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+P|k) \end{pmatrix} = \begin{bmatrix} C & 0 & \cdots & 0 \\ 0 & C & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C \end{bmatrix} \begin{pmatrix} \hat{x}(k+1|k) \\ \hat{x}(k+2|k) \\ \vdots \\ \hat{x}(k+P|k) \end{pmatrix} + \begin{pmatrix} D \\ D \\ \vdots \\ D \end{pmatrix} u(k-1) +$$
$$+ \begin{bmatrix} D & 0 & \cdots & 0 \\ D & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ D & D & \cdots & D \end{bmatrix} \begin{pmatrix} \Delta\hat{u}(k|k) \\ \Delta\hat{u}(k+1|k) \\ \vdots \\ \Delta\hat{u}(k+M-1|k) \end{pmatrix} + \begin{pmatrix} I \\ I \\ \vdots \\ I \end{pmatrix} \hat{d}(k|k)$$

or using the notation, proposed in equation (2.17) this can be represented as:

$$\hat{y}_k = F^*\hat{x}_k + D_y^* D u(k-1) + K^*\Delta\hat{u}_k + D_y^*\hat{d}(k|k) \tag{2.19}$$

where $D_y^* = [I_P \; I_P \; \ldots \; I_P]^T$ with I_P the identity matrix of dimension P.
Substitution (2.18) into (2.19) produces:

$$\begin{aligned} \hat{y}_k &= F^*\hat{x}_k + D_y^* D u(k-1) + K^*\Delta\hat{u}_k + D_y^*\hat{d}(k|k) \\ &= F^*\left[A^*\hat{x}(k|k) + B^*u(k-1) + C^*\Delta\hat{u}_k\right] + D_y^* D u(k-1) + K^*\Delta\hat{u}_k + D_y^*\hat{d}(k|k) \\ &= F^*A^*\hat{x}(k|k) + \left(F^*B^* + D_y^* D\right) u(k-1) + \left(F^*C^* + K^*\right)\Delta\hat{u}_k + D_y^*\hat{d}(k|k) \end{aligned}$$

resulting in:

$$\hat{y}_k = A^{**}\hat{x}(k|k) + B^{**}u(k-1) + C^{**}\Delta\hat{u}_k + D_y^*\hat{d}(k|k) \tag{2.20}$$

Matrices A^*, B^* and C^* in equation (2.17) and matrices A^{**}, B^{**}, C^{**} and D_y^* in equation (2.20) are constant during the entire simulation period and, therefore, they can be calculated off-line just once. Then, once these matrices are given the output predictions can be calculated at every iteration.

The final formulation in matrix-vector form of the optimization problem which the controller solves at every iteration is:

$$J_k = ||\hat{y}_k - y^{set}||_Q^2 + ||\hat{u}_k - u^{tar}||_R^2 + ||\Delta \hat{u}_k||_S^2$$

subject to:

$$\hat{y}_k = A^{**}\hat{x}(k|k) + B^{**}u(k-1) + C^{**}\Delta\hat{u}_k + D_y^*\hat{d}(k|k)$$

$$\hat{u}_k = D_u^* u(k-1) + E^*\Delta\hat{u}_k$$

$$y^{min} \leq \hat{y}_k \leq y^{max}$$

$$u^{min} \leq \hat{u}_k \leq u^{max}$$

$$\Delta u^{min} \leq \Delta\hat{u}_k \leq \Delta u^{max} \qquad (2.21)$$

Chapter 3

Stability and Performance of LP-MPC Control Systems

The aim of this chapter is to investigate the effect of the real-time optimization (RTO) level on the stability, performance and robustness on the combined RTO-MPC control system. The key focus here is on the LP-MPC scheme because of its widespread industrial implementation. Different formulations at the optimization and control levels will be considered and recommendations for appropriate two-level control system design will be determined and summarized. To consider the behavior of the two-level control system, simulations with different scenarios have been run and analyzed.

A process steady-state model for use at the RTO level comprises equations (material and energy balances, *etc*) that contain variables as well as fixed and adjustable parameters. A simple and commonly used method in industry for model updating is the bias update, which is suitable for the model structure:

$$f(y, u, \alpha) - \beta = 0$$

where: f is a vector function, y is a vector of dependant variables, u is a vector of manipulated variables, α are fixed parameters and β are adjustable parameters ("bias" term)

(Forbes and Marlin (1994a)). Two methods for the bias update are presented and analyzed in the chapter.

The main research in this chapter is focused on the problem of stability and performance. Asymptotic stability, rather than input-output stability is referred to here, although the classification is based on observations rather than rigorous proof. In the LP-MPC cascade control system, if there are no disturbances, the sequence of the set points and target values calculated by the LP and sent to the regulatory level should eventually stabilize at some constant values. If a disturbance suddenly enters the process, the plant optimum may change and it should take a limited number of re-optimizations to bring the plant to a new operating point by following a sequence of set points from the LP. If it does not occur i.e. the set points and/or target values from the LP do not become constant in time, the system is considered as unstable. If the system is stable but requires a significant number of LP runs for stabilization, then the performance of the cascade system is poor. If mild changes in the two-level control system design (for example, variation in constants, plant/model mismatch) cause significant degradation in performance or possible instability, then the system is considered as not robust.

Two forms of steady-state model bias update were considered: (i) bias update using LP set points, $d = y - y^{set}$, and (ii) bias update using model predictions, $d = y - K^{ss}u$ for steady-state optimization only (Forbes and Marlin, 1994a) or $d(k) = y(k) - \hat{y}(k|k-1)$ for optimization during the transient (Ying and Joseph, 1999). Here, K^{ss} is the model steady-state gain, $y(k)$ is plant output measurement at step k and $\hat{y}(k|k-1)$ is the predicted plant output at step k based on information available at step $k-1$. In both cases, the bias is the difference between the measured and predicted outputs; though, in the first case, the predicted output corresponds to the set point, calculated by the LP, while in the second case, the predicted output is calculated by applying the process inputs to the process model. Application of the first type of bias update was not found in open literature sources and its efficiency is studied in this chapter.

The stability and performance of the two-level control system with steady-state optimization only is analyzed in Section 3.1. The effect of more frequent optimization is studied in Section 3.2. Then, in Section 3.3, the effect of the control structure on the performance and robustness of the overall cascade system is presented. Finally, the chapter's summary and the conclusions are given in Section 3.4.

3.1 Steady-State Optimization

The first issue addressed was to determine whether the cascade system was able to exhibit unstable behavior, and if so, what design parameters may cause it.

In this section, the optimization at steady-state only was investigated. That is, the plant was allowed to reach steady-state before the next LP calculation was executed.

Different simulation scenarios have been run to understand and analyze the two-level control system behavior, and determine the factors which can affect the stability, performance and robustness of the system.

3.1.1 SISO case

Case Study 3.1.

$$\begin{array}{cc} Model & Plant \\ y(s) = \frac{G_m}{3s+1}u(s) & y(s) = \frac{0.5}{3s+1}u(s) \end{array}$$

Constraints (at both levels): $-1.0 \leq y \leq 1.0$, $-0.5 \leq u \leq 1.0$

MPC weights: $Q = 1.0$; $S = 2.0$;

Simulation parameters: prediction horizon $P = 50$; control horizon $M = 2$; sampling time $T_s = 0.3$;

LP objective function: $\max_{y,u} y$

First, plant/model mismatch was considered. The bias was calculated using the set points and measured outputs. To represent the observations in a compact form, the observation results are presented in tables with the following notation:

- "0" - both set points from the LP and controlled variables are unstable;
- "0.5" - set points from the LP and controlled variables stabilize; response initially oscillatory;
- "1" - Set points from the LP are unstable; controlled variables are constant;
- "2" - Controlled variables and given set points are constant or approach constants without oscillation.

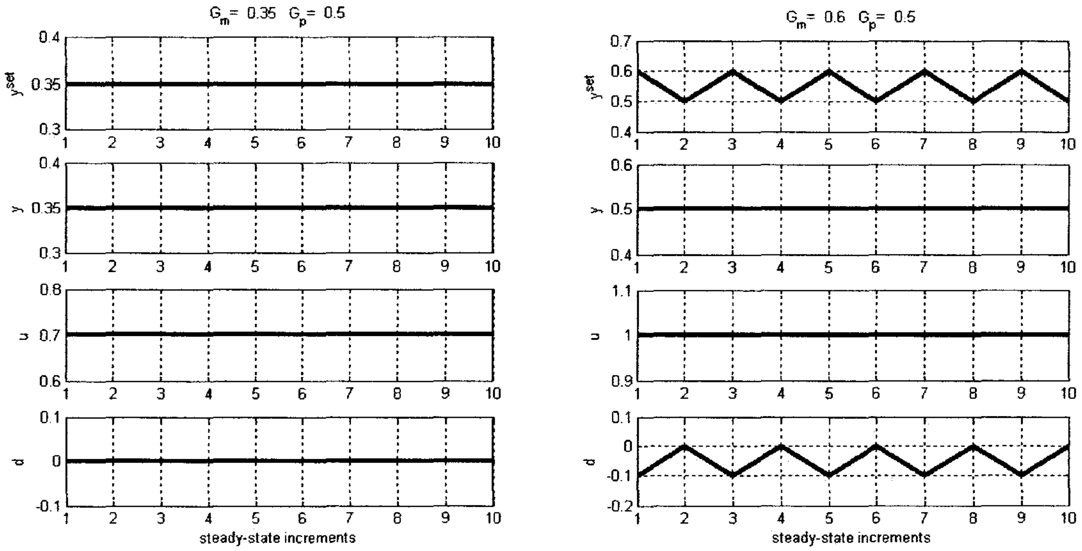
By "unstable" we mean that the response shows persistent variation and does not approach a constant value. For different values of the model steady-state gain, the resulting responses have been observed and summarized in Table 3.1.

Gm	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6
Observation	"2"	"2"	"2"	"2"	"2"	"2"	"2"	"2"	"1"	"1"

Table 3.1: Cascade System Responses for Different Values of Plant/Model Mismatch (Bias Uses the Set Points for Update)

With G_m less than or equal to the plant steady-state gain, the set points from LP and plant output are constant (see Figure 3.1(a)). However, with the model gain larger than the plant gain, chattering in the LP set point is observed (see Figure 3.1(b)).

The stability of the system response with the model gains smaller than the plant gains may be explained as follows. With identical constraints at both levels and the chosen objective function, the LP gives set points which can be archived at the regulatory level. The plant at each new steady-state arrives at the desired values and the bias which is updated at



(a) (b)

Figure 3.1: Two-level cascade system response. Bias uses the set points for update

the LP level is practically zero. The bias plot in Figure 3.1(a) confirms that. Conversely, when the model gain is larger than the plant gain, chattering in the set point from the LP is observed. It can be simply shown using a numerical example. Assume that the same case study is considered with model gain $g_m = 0.55$. Then at the first iteration the set point from the LP is: $y_1^{set} = g_m \cdot u_{max} = 0.55 \cdot 1.0 = 0.55$. The plant gain is smaller and therefore saturation takes place, and $y_1^{plant} = g_p \cdot u_{max} = 0.5 \cdot 1.0 = 0.5$. Then the bias is: $d_1 = y_1^{plant} - y_1^{set} = -0.05$. After the LP has been updated the new set point is: $y_2^{set} = g_m \cdot u_{max} + d_1 = 0.55 \cdot 1.0 - 0.05 = 0.5$. This set point is feasible for the MPC level and the new plant steady-state is: $y_2^{plant} = y_2^{set} = 0.5$ and $bias_2 = 0$. With the zero bias the new cycle begins. This explains why the chattering occurs with a period of 2 steady-state increments.

The simulations were repeated with the objective functions $\min_{y,u} y$, $\max_{y,u} u$ and $\min_{y,u} u$, and the qualitative behavior obtained is identical with the results presented in Table 3.1.

Sometimes the LP constraints for the outputs are tighter than the constraints at the MPC level. This is made to reduce the variability of the output product at steady-state. The effect of the constraints at the LP level can be analyzed in the same manner.

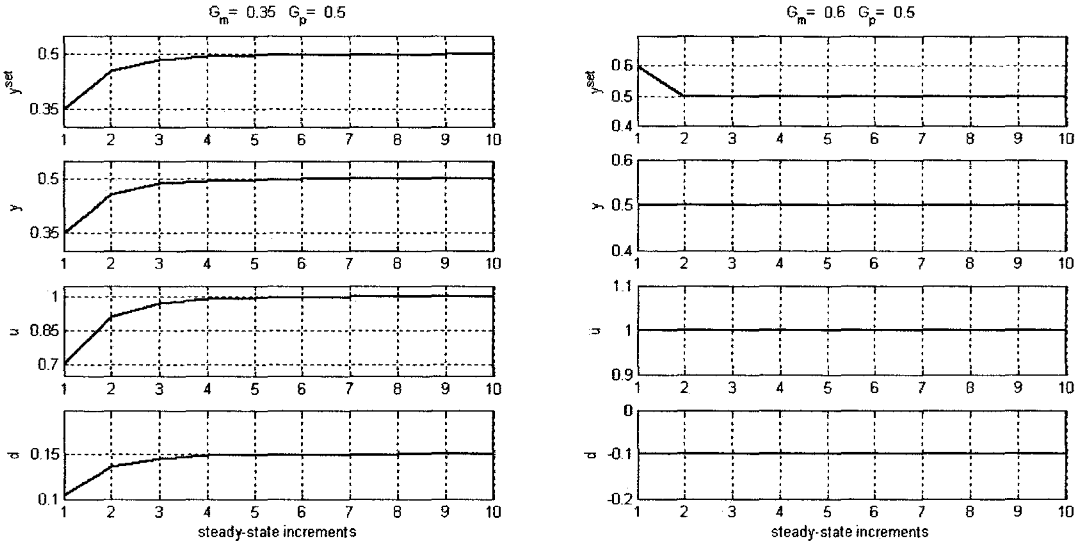
The output disturbance can also contribute to the stability of the two level system. With the premises made above assume that there is an output disturbance with the steady-state value $d_{ss} = 0.2$. That means that now maximum achievable value for y is $G_p \cdot u_{max} + d_{ss} = 0.5 \cdot 1.0 + 0.2 = 0.7$ and for all model gains in Table (3.1) the cascade system will be stable. However, with the model gains $G_m > 0.7$ the LP will generate non constant set points. The effect of the output disturbances with a negative steady-state value is straightforward.

Often the bias is considered as the difference between the actual and predicted plant outputs (Forbes and Marlin (1994a)). At every steady-state, the obtained inputs and the plant model can be used to find the predicted outputs. Then, the bias is the difference between the plant outputs and these predictions. Using this form of the optimizer update the effect of plant/model mismatch on the stability and performance of the cascade control system has been studied. The same case study as described above was considered. The results are presented in Table (3.2). As we can see from the table, this LP bias update method can significantly improve the stability properties of the LP set points. The resulting sequence of the LP set points and plant responses are presented in Figure 3.2 for the model gains corresponding to Figure 3.1.

Gm	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6
Observation	"2"	"2"	"2"	"2"	"2"	"2"	"2"	"2"	"2"	"2"

Table 3.2: Cascade System Responses for Different Values of Plant/Model Mismatch (Bias Uses Model Predictions for Update)

A great difference between Figures 3.1(a) and 3.2(a) is that the plant is maintained at different operating points, and since the objective function is to maximize the output, then it indicates that bias update with model predictions is more profitable. When the plant gain



(a) (b)

Figure 3.2: Two-level cascade system response. Bias uses predictions for update

is smaller than the model gain, the plant operates at the same point; however in Figure 3.2(b) the set points from LP do not experience chattering.

When the plant gain is larger than the model gain (see Figure 3.2(a)), several iterations are needed to achieve a constant value in the set points and plant output. This can be explained using numerical values. At the first iteration when the bias is zero, the set point is determined by the model gain only and is equal to: $y_1^{set} = G_m \cdot u_{max} = 0.35 \cdot 1 = 0.35$. Since the plant gain is larger, this set point can be achieved without input saturation ($y_1^{plant} = y_1^{set}$) and the steady-state input value becomes equal to: $u_1^{plant} = y_1^{plant}/G_p = 0.35/0.5 = 0.7$. Then the output prediction for this input is: $\hat{y}_1^{plant} = G_m \cdot u_1^{plant} = 0.35 \cdot 0.7 = 0.2450$. The bias is simply the difference between the measured plant value and the predicted output using the model: $d_1 = y_1^{plant} - \hat{y}_1^{plant} = 0.35 - 0.2450 = 0.1050$. This value coincides with the value in Figure 3.2(a). After the bias has been updated, the solution of the LP at the next iteration is: $y_2^{set} = G_m \cdot u_{max} + d_1 = 0.35 \cdot 1 + 0.105 = 0.455$ etc. After several iterations, the set point from the LP approaches a constant value and that is consistent with the measured plant output. At the same time bias value reaches a constant value:

$d_\infty = (G_p - G_m) \cdot u_{max} = 0.15 \cdot 1 = 0.15$. This shows that such a method of bias update sequentially introduces to the top level information about the plant and, eventually the LP set points and plant output settle down at the plant optimum level.

When the model gain is larger than the plant gain one iteration is enough for the LP to start providing constant set points for the plant. Assume, that $G_p \leq G_m$. At the first iteration the set point from the LP is: $y^1_{set} = G_m \cdot u_{max}$. Since the plant gain is smaller, this set point cannot be achieved, the plant input will saturate and the measured plant output will be: $y_1^{plant} = G_p \cdot u_{max}$. The predicted outputs are: $\hat{y}_1^{plant} = G_m \cdot u_{max}$. Then the bias is: $d_1 = y_1^{plant} - \hat{y}_1^{plant} = G_p \cdot u_{max} - G_m \cdot u_{max}$. After this bias has been substituted into the LP, the new LP set point is: $y^2_{set} = G_m \cdot u_{max} + d_1 = G_m \cdot u_{max} + G_p \cdot u_{max} - G_m \cdot u_{max} = G_p \cdot u_{max}$. After the first bias update, the LP will have effectively the same steady-state model as the plant does, and all further set points given from the LP will be consistent with the plant. Therefore, the bias value is: $d_1 = G_p \cdot u_{max} - G_m \cdot u_{max} = 0.5 - 0.6 = -0.1$ and keeps constant after the first iteration.

With such a bias update, the analysis of the effect of step-like output disturbances is straightforward. The steady-state disturbance value will be simply compensated by the bias value and it should not affect stability and performance.

3.1.2 MISO case (1 × 2 system)

After the SISO system has been considered, it is of interest to devote some attention to a MISO system. Since there are more inputs than outputs, some of them can be provided specified "ideal resting values". Also, even though several variables are controlled (output and some inputs), only one variable - the output, is involved in the bias update.

Case Study 3.2.

Model	*Plant*
$y(s) = \frac{G^m_{11}}{3s+1} u_1(s) + \frac{G^m_{12}}{3s+1} u_2(s)$	$y(s) = \frac{0.4}{3s+1} u_1(s) + \frac{0.3}{3s+1} u_2(s)$

Constraints (at both levels): $-1.0 \leq y \leq 1.0$; $\quad -0.5 \leq u_1,\ u_2 \leq 1.0$

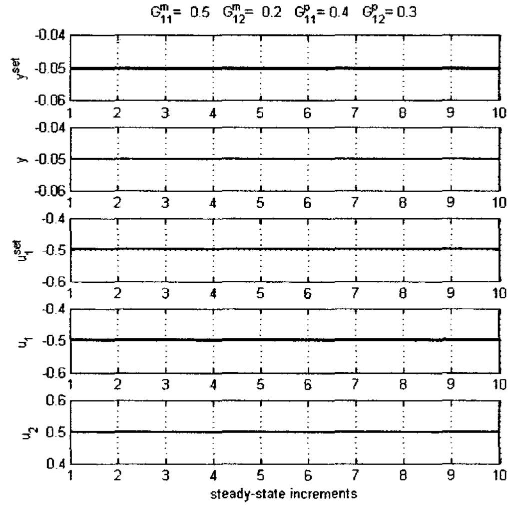
MPC weights: $Q = 1.0$; $R = \text{diag}(1.0,\ \ 0.0)$; $S = 5.0 I_m$;

Simulation parameters: prediction horizon $P = 50$; control horizon $M = 2$; sampling time $T_s = 0.3$;

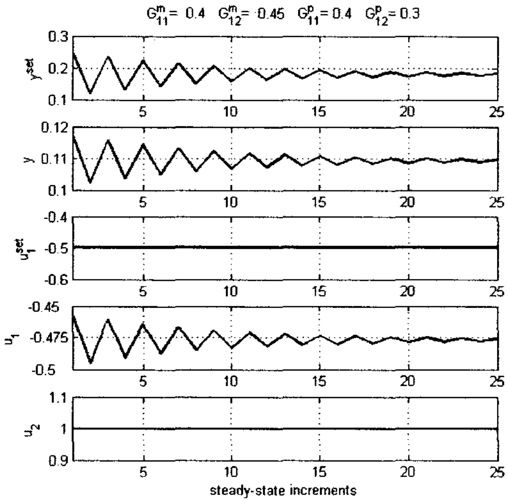
Controlled variables: y and u_1

Effect of objective function and plant/model mismatch on the stability and performance of the cascade control system.

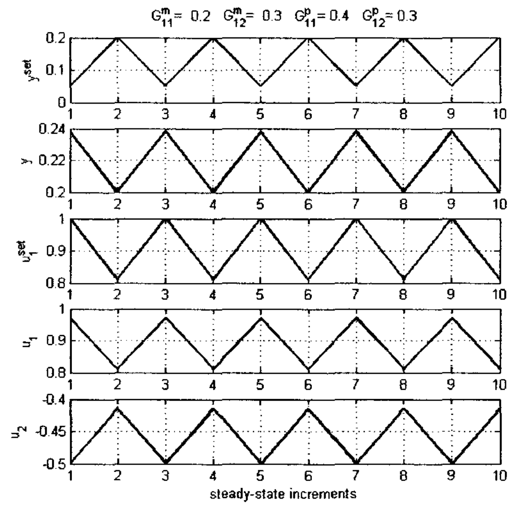
Simulations for different combinations of plant/model mismatch have been run to determine the effect of the objective function on the cascade control system behavior. The simulations have shown that the two-level control system can exhibit different responses. The responses were classified into four different cases, similar to those listed in Section 3.1.1. Examples of such responses are presented in Figure 3.3. Bias update scheme (i) was used, and different responses were obtained depending on the objective function and plant/model mismatch. Results of the simulations are presented in Figure 3.4. These figures indicate that two-level cascade control system could have different behavior for different values of the mismatch. Also, different types of behavior are not scattered randomly but form some distinct areas on the plots. In the case presented in Figure 3.4(a), for the amount of mismatch considered, the controlled variables are constant, although chattering in the set points is possible. In the case when u_2 is not included in the objective function (Figure 3.4(b)), the two-level control system can exhibit severe oscillation (i.e. set points and plant outputs reach constant values after many iterations of steady-state optimization. The number of iterations can be 20 and more). Some transient regions can be seen where changes in the mismatch can shift the system from stable operation to operation with sustained oscillation in the set points which could decay over long periods, resulting in poor performance. Although simulations were run for many different objective functions, the case when the set points from the LP and any of the controlled variables were unstable (never reach constant values as in Figure 3.3(c)) was not observed (for identical constraints at both levels). Even though



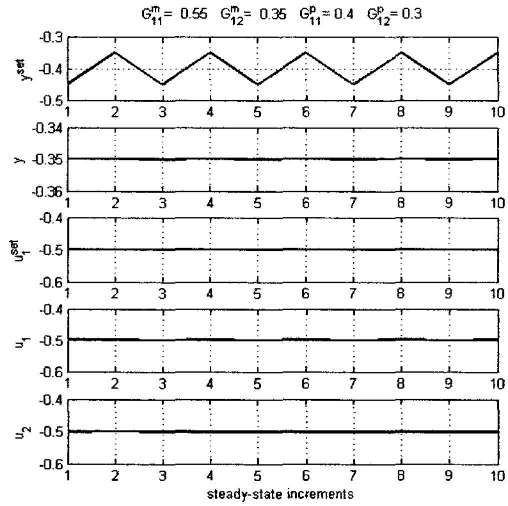
(a) Controlled variables and given set points are constant. Objective function: $\max_{y,u_1} y - u_1$



(b) Set points from LP and controlled variables stabilize; response initially oscillatory. Objective function: $\max_{y,u_1} y - u_1$

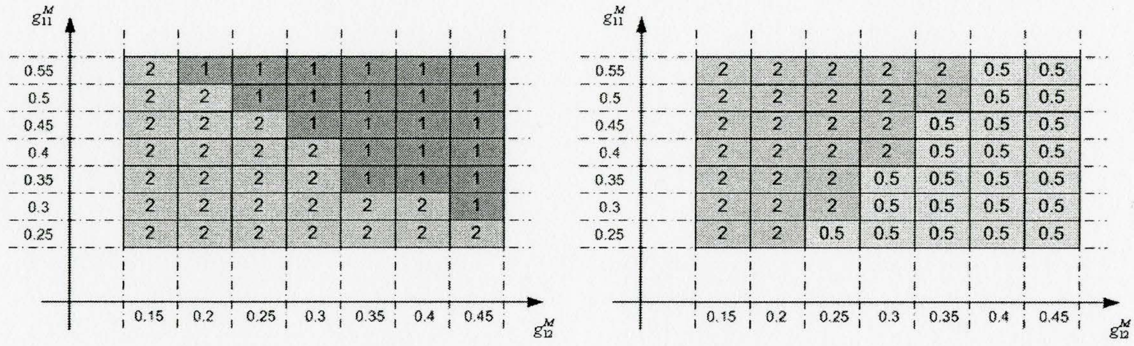


(c) Set points from LP and controlled variables are unstable. Objective function: $\max_{y,u_1} y - u_2$, at the LP level: $-0.2 \leq y \leq 0.2$



(d) Set points from LP are unstable; controlled variables are stable. Objective function: $\max_{y,u_1} y - u_1 - u_2$

Figure 3.3: Two level control system responses. Bias uses set points for update



(a) Objective function $\max_{y,u_1} y - u_1 - u_2$

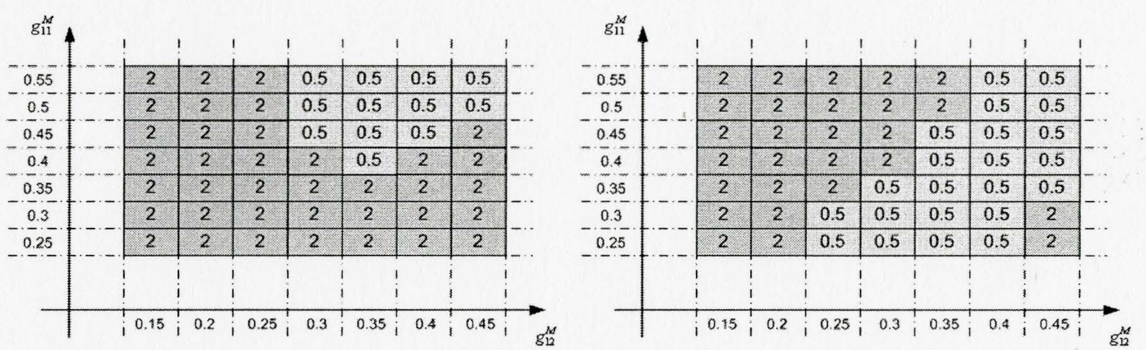
(b) Objective function $\max_{y,u_1} y - u_1$

Figure 3.4: Cascade system behavior for different values of model gains (effect of objective function)

the performance was very poor and it may have taken many iterations, the set points and the controlled variables eventually achieved constant values over time.

Effect of the constraints and plant/model mismatch on the stability and performance of the cascade control system.

The use of different sets of constraints also has effect on the cascade control system behavior. In the simulations here, the constraints at the LP level for the output variable y have been shrunk to: $-0.2 \leq y \leq 0.2$. The results of the simulations are presented in Figure 3.5. The presented plots indicate that the performance of the two-level system depend on the constraints at the LP level. The simulation results for the case when both inputs and the output are included in the objective function are presented in Figures 3.4(a) and 3.5(a). Results presented in Figure 3.4(a) are similar to the results presented in Table 3.1. If the model steady-state gains are smaller than the plant steady-state gains, the two-level control system is stable. Large model gains with this bias update method cause the set points from the LP to chatter although the plant outputs are constant. However, if the LP constraints for the output variable are shrunk, different two-level system responses for the same range of the plant/model mismatch were observed (see Figure 3.5(a)). With the original constraints, the solution of the LP lies at the input variables constraint and the output arrived at the steady-state with the offset which affected the LP solution at the next iteration causing chattering in the LP set points. Since now the LP constraints are tighter, the LP solution lies at the output constraint and chattering in the set point for the output was not observed. The resulting two level control system behavior now depends on the variable u_2 which is used for control. If it does not saturate, the output will be driven to the set point without offset providing a stable response for the two-level system. If saturation of u_2 occurs, steady-state offset will appear causing oscillation in the target value for u_1 and consequently in the plant output (since u_2 is saturated). Saturation of u_2 depends on the composition of the objective function and the value of the plant/model mismatch. This explains why the area with chattering set points in Figure 3.4(a) parted into areas with stable response and oscillatory responses in Figure 3.5(a). With u_2 not included into the objective function the effect of the constraints tightening at the LP level is not significant which can be seen in the similarity of Figures 3.4(b) and 3.5(b)).



(a) Objective function $\max_{y,u_1} y - u_1 - u_2$ (b) Objective function $\max_{y,u_1} y - u_1$

Figure 3.5: Cascade system behavior for different values of model gains (effect of the constraints)

A case when instability of the two-level system with this bias update was observed occurs in the present case study with objective function $\max_{y,u_1} y - u_2$ and shrunk output constraints at the LP: $-0.2 \leq y \leq 0.2$ (see Figure 3.6).

A possible reason for such behavior is the absence of the controlled input in the objective

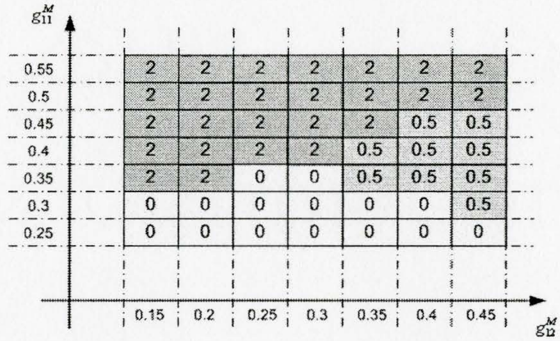
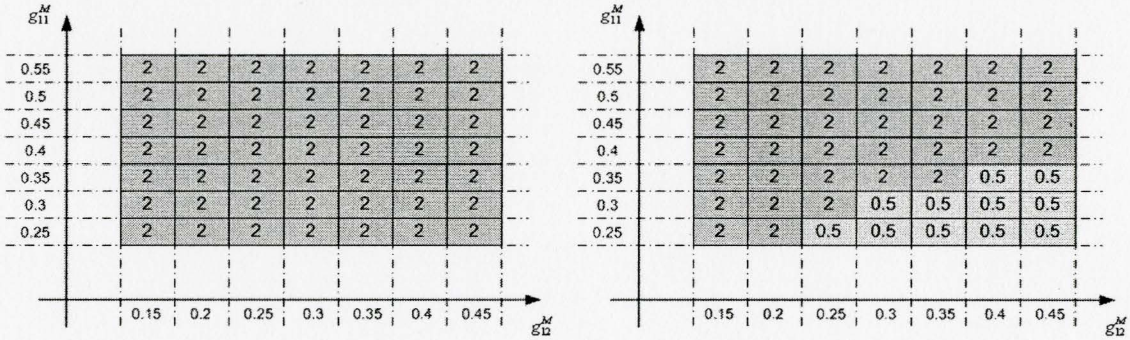


Figure 3.6: Effect of the constraints on the two-level system behavior. Objective function $\max_{y,u_1} y - u_2$; LP constraints for the output: $-0.2 \leq y \leq 0.2$

function.

The same case studies were considered with the bias update using the model predictions. With the bias update through the model predictions, instability was not observed. The results which correspond to problem parameters as in Figures 3.4 and 3.5 are presented in Figure 3.7(a). In this figure it can be seen that for the considered constraint shifts and the objective functions, the two-level control system with the bias update using model predictions is stable and there is no oscillation. Figure 3.7(b) corresponds to problem parameters as in Figure 3.6. Here, for a large g_{12}^m model gain and tighter output constraints, the set points and the controlled variables exhibit oscillation decaying in time.



(a) Case studies presented in Figures 3.4, 3.5

(b) Case study presented in Figure 3.6

Figure 3.7: Bias update with model predictions

This indicates that the bias update through model predictions significantly improves performance and stability of the two-level cascade control system. This result is consistent with the bias update formulation and use in Forbes and Marlin (1994a), and results presented in Ying and Joseph (1999), and accounts for discrepancies between the outputs and set points when input saturation occurs. It is also consistent with the use of the DMC disturbance estimation scheme, and is directly applicable when the LP is executed at higher frequencies.

3.1.3 MIMO case (2×2 system)

As the case study here, the Shell Standard Control Problem (Prett and Morari, 1986) is considered. This is a multivariable problem concerning control of a heavy oil fractionator. The column setup is shown in Figure 3.8.

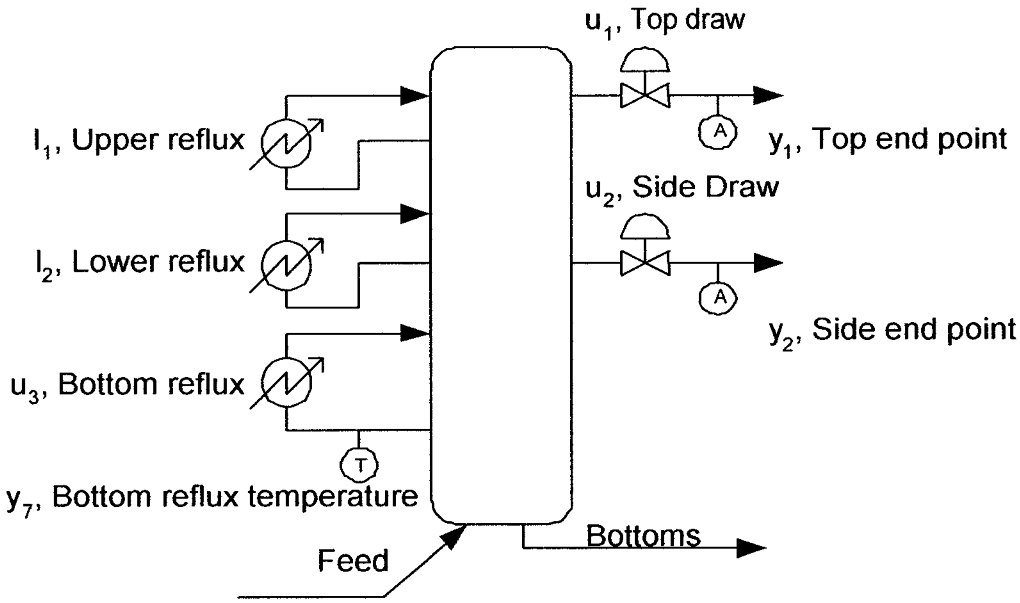


Figure 3.8: Shell heavy oil fractionator

Case Study 3.3. Plant and model descriptions were taken from (Prett and Morari, 1986).

Plant transfer matrix:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_7 \end{bmatrix} = G^p \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad G^p = \begin{bmatrix} \frac{4.0500e^{-27s}}{50.0s+1} & \frac{1.7700e^{-28s}}{60.0s+1} & \frac{5.8800e^{-27s}}{50.0s+1} \\ \frac{5.3900e^{-18s}}{50.0s+1} & \frac{5.7200e^{-14s}}{60.0s+1} & \frac{6.9000e^{-15s}}{40.0s+1} \\ \frac{4.3800e^{-20s}}{33.0s+1} & \frac{4.4200e^{-22s}}{44.0s+1} & \frac{7.2000}{19.0s+1} \end{bmatrix}$$

Uncertainties in the gains of the model:

$$G^m = \begin{bmatrix} \frac{(4.0500+2.11\epsilon_1)e^{-27s}}{50.0s+1} & \frac{(1.7700+0.39\epsilon_2)e^{-28s}}{60.0s+1} & \frac{(5.8800+0.59\epsilon_3)e^{-27s}}{50.0s+1} \\ \frac{(5.3900+3.29\epsilon_1)e^{-18s}}{50.0s+1} & \frac{(5.7200+0.57\epsilon_2)e^{-14s}}{60.0s+1} & \frac{(6.9000+0.89\epsilon_3)e^{-15s}}{40.0s+1} \\ \frac{(4.3800+3.11\epsilon_1)e^{-20s}}{33.0s+1} & \frac{(4.4200+0.73\epsilon_2)e^{-22s}}{44.0s+1} & \frac{(7.2000+1.33\epsilon_3)}{19.0s+1} \end{bmatrix}$$

where $-1.0 \leq \epsilon_1, \epsilon_2, \epsilon_3 \leq 1.0$.

In this case study it was assumed that $\epsilon_1 = 0.85$, $\epsilon_2 = -0.6375$, $\epsilon_3 = -0.6375$, which resulted in the following model transfer matrix:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_7 \end{bmatrix} = G^m \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad G^m = \begin{bmatrix} \frac{5.8435e^{-27s}}{50.0s+1} & \frac{1.5214e^{-28s}}{60.0s+1} & \frac{5.5039e^{-27s}}{50.0s+1} \\ \frac{8.1865e^{-18s}}{50.0s+1} & \frac{5.3566e^{-14s}}{60.0s+1} & \frac{6.3326e^{-15s}}{40.0s+1} \\ \frac{7.0235e^{-20s}}{33.0s+1} & \frac{3.9546e^{-22s}}{44.0s+1} & \frac{6.3521}{19.0s+1} \end{bmatrix}$$

The control objectives and constraints were taken from Ying and Joseph (1999) and are stated as follows.

Regulatory level constraints:

$$-0.5 \leq u_i \leq 0.5, \quad |\Delta u_i| \leq 0.05/min, i = 1,2,3$$

$$-0.5 \leq y_1 \leq 0.5; \quad -0.5 \leq y_7$$

LP constraints:

$$-0.5 \leq u_i \leq 0.5, \quad i = 1,2,3;$$

$$-0.005 \leq y_1 \leq 0.005; \quad -0.005 \leq y_2 \leq 0.005; \quad -0.5 \leq y_7$$

LP objective:

$$\min_{y_1,y_2,u_3} \quad u_3$$

Controlled variables:

$$y_1, \quad y_2, \quad u_3$$

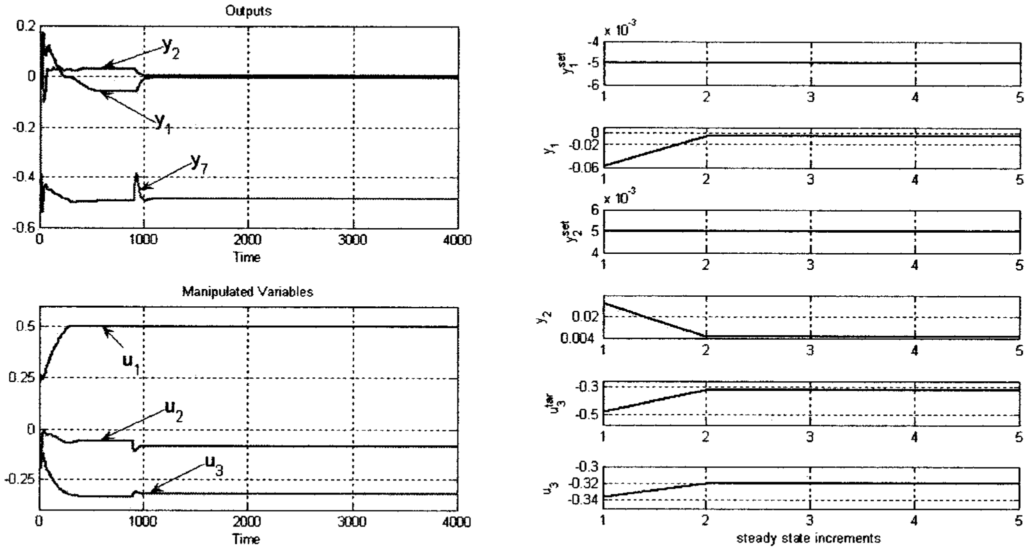
Weights: $Q = \text{diag}(2.0,\ 2.0,\ 0.0)$; $R = \text{diag}(0.0,\ 0.0,\ 2.0)$, $S = 2.0 I_m$;
Simulation parameters: prediction horizon $P = 30$, control horizon $M = 2$, sampling time $T_s = 6.0$

In the previous section is was shown that the bias calculated using model predictions (scheme (ii)) results in significant improvement of the two-level control system performance and, therefore, this scheme is used in all further simulations throughout the thesis.

Case Study 3.3 was considered for different values of the plant/model mismatch, and the simulations showed that the two-level cascade system is stable. Figure 3.9 shows the response for the mismatch described in the case study; the controlled variables approach the given set points after the second iteration and the overall two-level control system performance is good.
Next, one of the constraints at the LP level was changed:

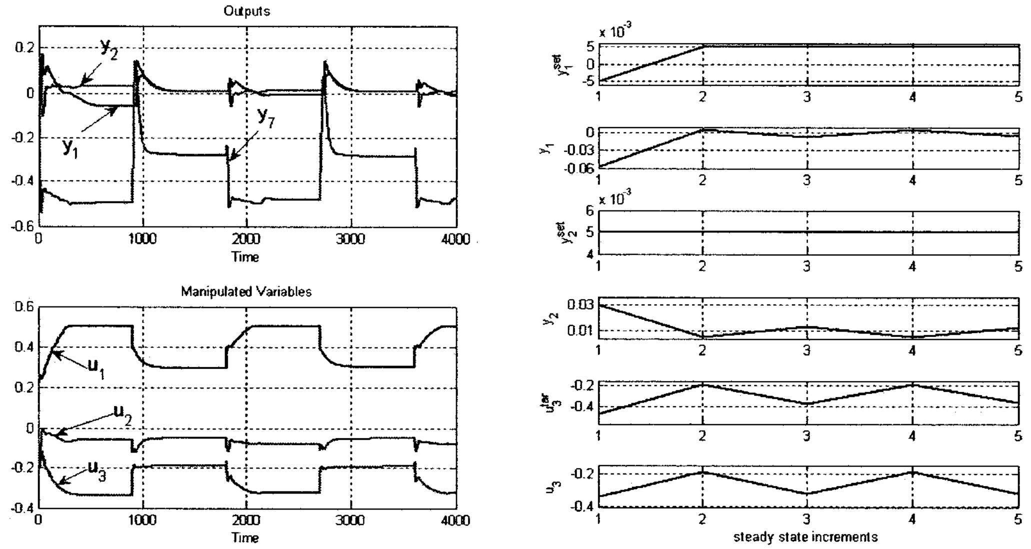
$$-0.4 \leq y_7$$

This minor change at the LP level introduced significant changes in the overall system behavior. It can be seen in Figure 3.10 that MPC is stable and brings the plant to the new steady-state. However, every reoptimization of the set points drives the plant to a new steady-state without overall stabilization at particular constant values. The possible reason for such behavior is the fact that the output variable y_7 is not controlled but is used for the bias update. This assumption was confirmed by the simulation where the mismatch for y_7 was removed (presented in Figure 3.11). Stability of the system was achieved, although the performance was poor and it took many iterations to arrive to a constant steady-state. The steady-state equation which relate y_7 to the manipulated variables cannot be omitted at the LP level, because its steady-state value must be monitor to guarantee its presence



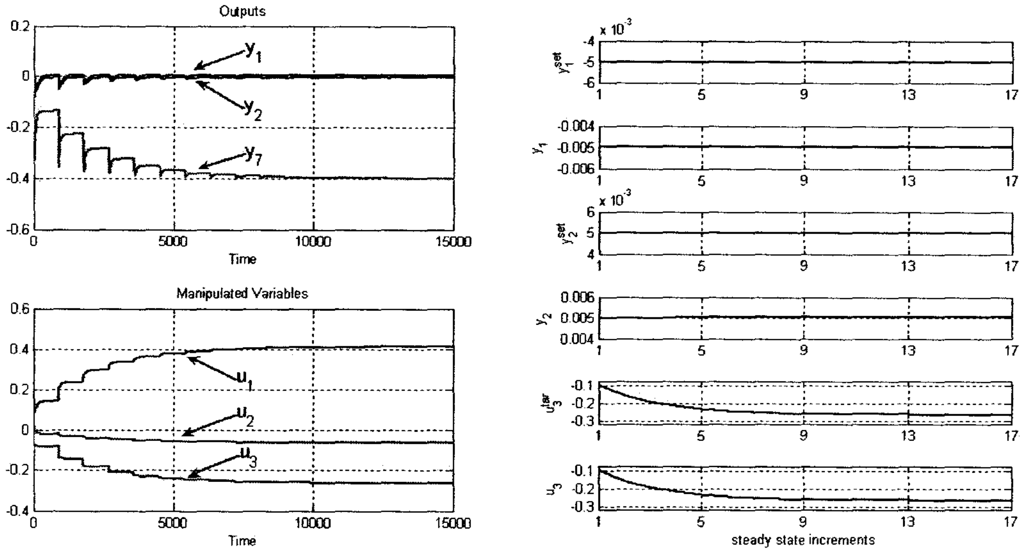
(a) Plant responses. (b) LP solutions and plant steady-states.

Figure 3.9: Steady-state optimization. MIMO case. $-0.5 \leq y_7$



(a) Plant responses. (b) LP solutions and plant steady-states.

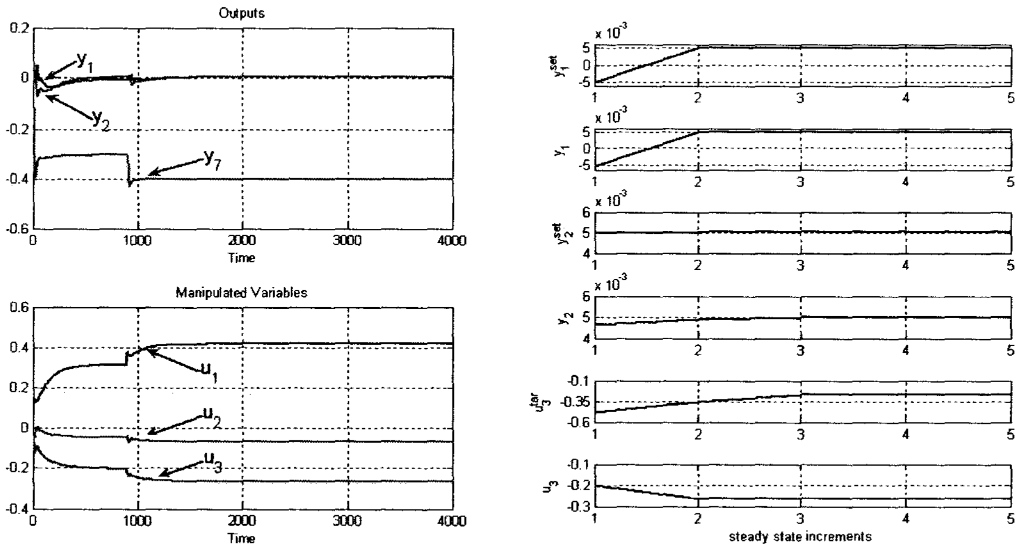
Figure 3.10: Steady-state optimization. MIMO case. $-0.4 \leq y_7$



(a) Plant responses.

(b) LP solutions and plant steady-states.

Figure 3.11: Steady-state optimization. No mismatch in the model for y_7 $(-0.4 \leq y_7)$.



(a) Plant responses.

(b) LP solutions and plant steady-states.

Figure 3.12: Steady-state optimization. All outputs are controlled $(-0.4 \leq y_7)$

inside the bounds. Therefore, in case when the range control is implemented, the issue of the bias update using non-controllable outputs must be addressed. Also, simulations where control of all outputs (y_1, y_2 and y_7 are controlled variables, u_3 is used for control) with the original mismatch was considered. The system appeared to be stable and the performance was good despite the mismatch (See Figure 3.12).

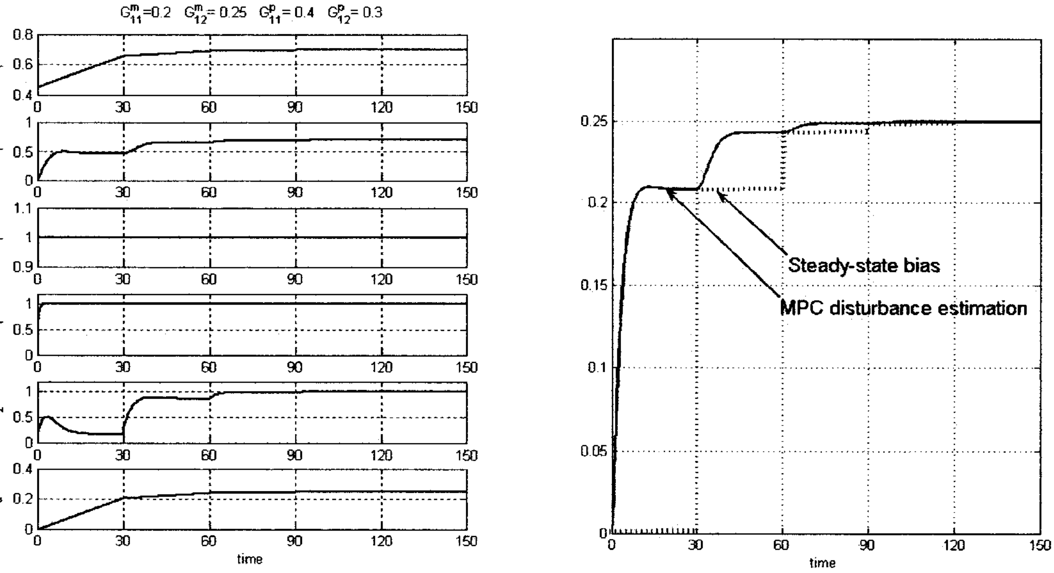
3.2 LP Optimization between the Steady-States

Often in practice the LP is executed not only at steady-state but also during the transient. Since the process is not at steady-state, the set points cannot be used for the LP update, i.e. scheme (i) cannot be used in this case. At every iteration when the LP is implemented, the difference between the output measurements and the predicted output value is considered as the bias and is updated at the LP level.

Ying and Joseph (1999) analyzed the stability of a two-level cascade control system for the perfect model case. Here, the effect of plant/model mismatch and the frequency of the LP optimization implementation is considered. Case Study 3.2 presented in Section 3.1.2 was used for simulations. The following parameters were chosen for the simulations: $G_{11}^m = 0.2$, $G_{12}^m = 0.25$ and the objective function is $\max_{y,u_1} y$.

The results of the simulations with steady-state optimization only are presented in Figure 3.13(a).

As discussed previously, since the model gains are smaller than the plant gains it takes several iterations for the system to reach a constant operating point. From iteration to iteration the bias accumulates the error between the predictions and the measurements. The bias steady-state value depends on the mismatch and the steady-state inputs and can be calculated as: $d_{ss} = G_{11}^p \cdot u_1^{ss} + G_{12}^p \cdot u_2^{ss} - G_{11}^m \cdot u_1^{ss} - G_{12}^m \cdot u_2^{ss} = 0.4 + 0.3 - 0.25 - 0.2 = 0.25$ (this is confirmed by the data in the plot). The comparison of the bias values and the



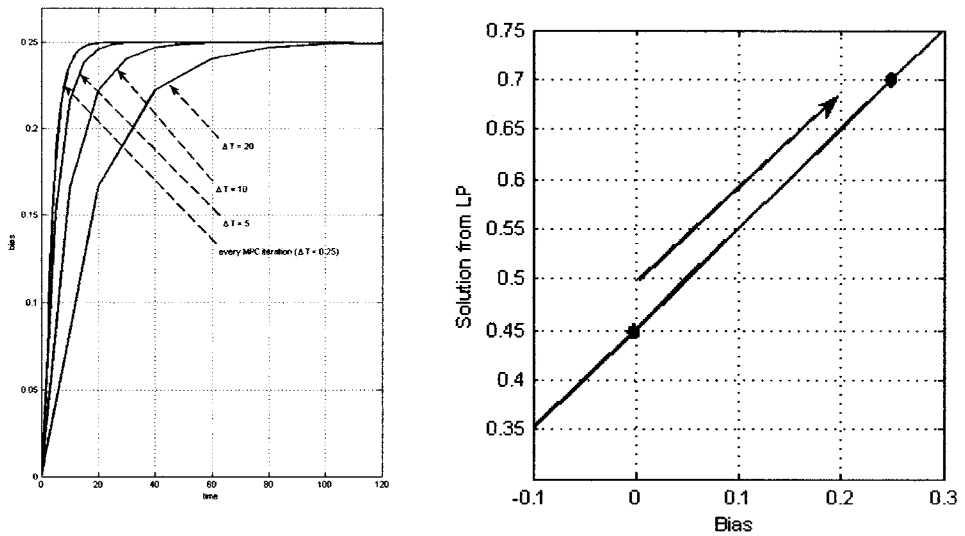
(a) Plant responses.

(b) Bias values and MPC disturbance estimation.

Figure 3.13: Steady-state optimization

MPC controller disturbance values is presented in Figure 3.13(b). With this bias update method, the bias value at steady-states is equal to the output disturbance estimated at steady-states. However, as can be seen during the transient, the bias does not approximate the disturbance well, and therefore, it requires several steady-state passes to start operating at the plant optimum. This motivates the idea to update the bias more often.

Simulations have been run with different frequencies of the LP optimizer implementation. The resulting biases are presented in Figure 3.14(a). With more frequent LP implementation, the bias approaches steady-state faster as well as the set points from the LP. The LP implementation at every MPC iteration corresponds to the fastest way to the optimum operation. Also, frequent disturbance estimation makes the bias change smoothly. Therefore, the set points from the LP do not change drastically contributing to the stability of the overall system. These can be the reasons why many researchers have proposed to use the optimizer at every iteration of LP implementation (Kassman *et al.*, 2000; Sorensen and Cutler, 1998; Ying and Joseph, 1999).



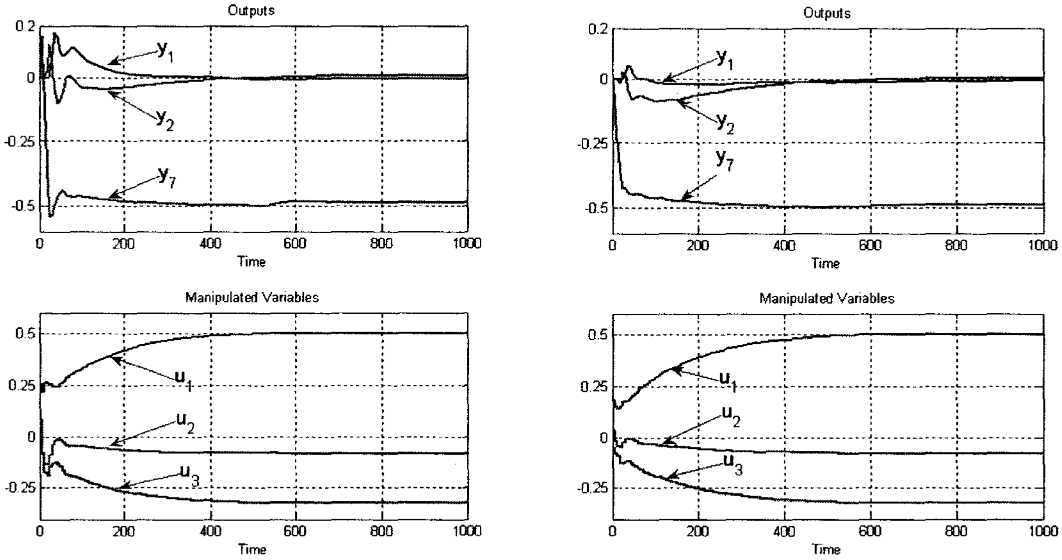
(a) LP biases for different frequencies of the optimizer implementation.

(b) LP solution depends on the bias value.

Figure 3.14: Frequent LP optimization

The effect of the bias value on the LP solution is presented in Figure 3.14(b). Here, the LP solution is presented as the function of the bias value. At the very first iteration, the bias is zero and the LP solution is 0.45 (this is consistent with the results in Figure 3.13(a)). From the Figure 3.14(a) it follows that in time the bias is increasing and reaches steady-state at 0.25. This corresponds to the proportional changes in LP solutions from 0.45 to 0.7. The rate of the solution shifting depends on the frequency of the LP implementation. Also, this figure can be used for stability analysis. From the figure it follows that for every feasible bias value there exists a unique LP solution and with changes in the bias, this solution changes proportionally. In case of LP execution at every MPC iteration, the bias term changes smoothly along this line and so does the set point from the LP. This precludes erratic set point changes thereby contributing to the overall stability. Since for a fixed bias term the solution of the LP is unique, the two-level system instability requires permanent fluctuation of the bias over time. Observations showed that the important factor to avoid such behavior is to update the LP bias frequently. Even though the instability of the two-level system with the bias update at every iteration has not been observed, it cannot be deduced that such a scenario could never happen, and the rigorous stability proof is an open issue.

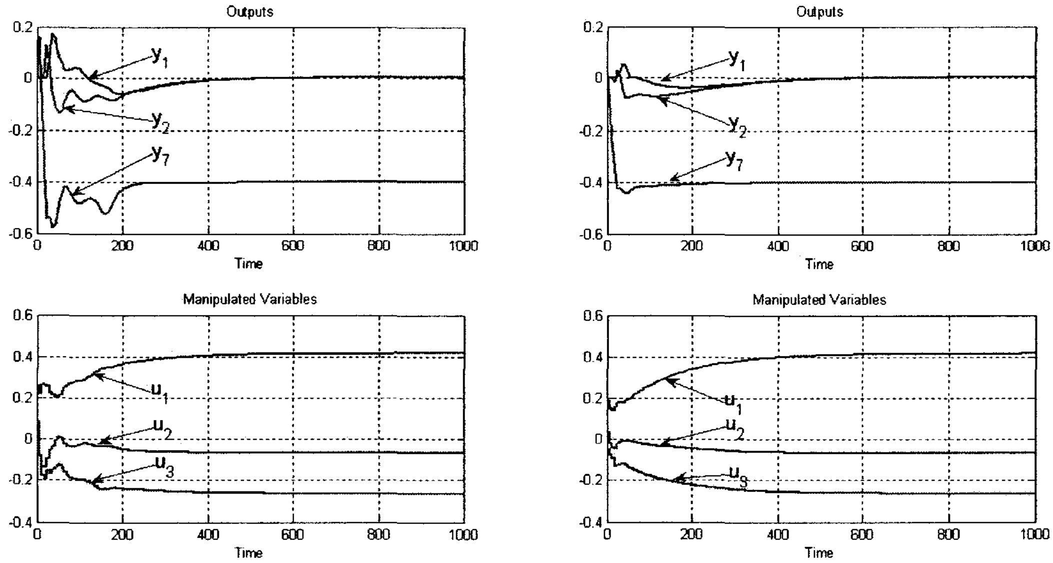
Constraint shifts at the LP level should not affect the stability and performance properties of the system except for the case when the non-controlled outputs are used for the bias update. The MIMO system from the case study presented in Section 3.1.3 was considered to check this issue. For the case when the MIMO system exhibited a stable response ($-0.5 \leq y_7$), the LP optimization at every MPC iteration was also stable, although the performance of the system was poor (See Figure 3.15).



(a) y_1, y_2 and u_3 are controlled variables. (b) y_1, y_2 and y_7 are controlled variables.

Figure 3.15: Optimization at every iteration ($-0.5 \leq y_7$)

Unlike the steady-state optimization case, the two level system was stable with tighter constraints ($-0.4 \leq y_7$) (See Figure 3.16). In the case when all outputs were controlled and the LP was implemented at every iteration, the two-level control system was stable and had better performance than in case when y_7 was not controlled (See Figure 3.16(a)). The simulations showed that optimization at every iteration can provide stability to a system



(a) y_1, y_2 and u_3 are controlled variables. (b) y_1, y_2 and y_7 are controlled variables.

Figure 3.16: Optimization at every iteration ($-0.4 \leq y_7$)

which exhibited unstable operation with steady-state optimization only.

3.3 The Problem of the Control Structure Selection

As mentioned in Section 3.1.3, the chosen control structure may have a significant impact on the stability and performance of the cascade system with steady-state to steady-state operation. Tight constraints at the LP level may cause unstable overall system steady-state operation. However, in the case when all outputs were controlled, the system stabilized, though the performance was poor. The other possible remedy to overcome the system instability is more frequent LP execution. The responses for the same case study with LP execution at every MPC iteration were stable although the control performance was different for different sets of controlled and manipulated variables. In these simulations, even though at steady-state the plant operates at the same economic operating point ($u_3^{ss} = -0.2641$) the transient response for the case when y_7 is controlled and u_3 is not, is better than in the opposite case.

All these observations indicate that although the optimal steady-state operation corresponds to minimum consumption of u_3, this variable should not necessarily be included in the MPC objective function as a controlled variable, and there may be another set of CVs and MVs which are able to provide stable responses with better performance and operate at no worse economic operating point. However, determining a proper control structure is not a trivial problem. For this case study there are two reasonable combinations which can be compared using simulations. If there are many MVs and CVs and the LP objective function comprises many variables it may be a formidable task to consider each possible control structure since the number of the possible combinations increases significantly with an increase in plant dimension. This becomes even more difficult since for any chosen control structure the MPC objective function weights can affect the two-level control system response and, therefore, they need to be determined in a proper manner.

This motivates the idea to formulate the problem of control structure selection within an optimization framework of optimal control system design. Significant amount of research devoted to optimization-based design uses PI and PID controllers (Schweiger and Floudas, 1998; Mohideen *et al.*, 1996; Kookos and Perkins, 2001). Kookos and Perkins (2002) studied the decision making control design problem on whether a centralized or decentralized controller should be used in a system. The decision making framework for a two-level control system structure selection is required for the present problem. In such a framework, the decision variables are binary variables which represent the control system structure applied to a plant. The objective function in the problem may have different forms, for example, identical to the LP objective function (which means to find such a control structure which corresponds to the operation at the point where the LP has the best possible solution), the minimum squared error form for some of the controlled variables (to provide good control performance) or any other. The overall two-level closed-loop control system response can be modelled as a multilevel optimization problem, which may be transformed into a single-level problem with complementarity constraints using the Karush-Kuhn-Tucker conditions of the sub-problem as described in Baker and Swartz (2007). However, the problem now includes binary variables associated with the control structure. The solution of the problem specifies which of the binary coefficients are unity, i.e. the corresponding variables are the controlled variables, and which are zeros, i.e. the corresponding variables are not controlled. In the case of steady-state optimization only, the plant dynamic responses between the optimizations can be substituted by steady-state simulations (for closed-loop stable plants) which would simplify the problem formulation. Two methods for such calculations are given and discussed in Chapter 5.

3.4 Chapter Summary

The performance of LP-MPC cascade control systems was explored under variety of conditions. This included investigation of the effects of plant/model mismatch, bias update scheme, constraints, LP objective function, control structure and frequency of LP execution.

The presented results have shown that in a deterministic system with step-like disturbances it is unlikely that two-level cascade system instability would occur if an appropriate bias model update is used. The bias calculated using model predictions contributes to the stability of the overall system; moreover, it guarantees system operation at the plant optimum under certain conditions (Forbes and Marlin, 1994a). Plant/model mismatch did not appear to be a cause of two-level system instability; however, it has an effect on the system performance.

More frequent LP execution than steady-state only, seemed to contribute to stabilization to a two-level control system and it also provided more rapid steady-state achievement.

Chattering in the LP set points was observed in case when uncontrolled constrained outputs were used for the bias update. More frequent LP execution was able to stabilize the system; however, the performance was poor. Changes in the control structure were able to improve the performance. These observations lead to opportunities for future research where the effect of the control structure and LP execution frequency on the stability and performance of the overall two-level control system may be studied.

Chapter 4

Calculation of Closed-Loop Steady-State of Constrained MPC System

Steady-state simulations are an important stage of the system design where the material and energy balances as well as equipment properties and machinery efficiency can be determined and checked. The control systems are usually designed and incorporated at later stages to achieve required performance. However, the process steady-state behavior depends not only on the process itself but also on the control system which maintains it at steady-state. Therefore, the effects of the control system on the steady-state plant behavior should be taken into account in steady-state simulations.

As discussed in Section 2.1, steady-state optimization requires the plant to reach steady-state prior to RTO implementation. In industry, steady-state detection criteria must be satisfied before the optimization may be executed. For research purposes the behavior of the two-level cascade system can be studied using computer simulations. In this case, the execution period for the RTO has to be long enough (this period can be estimated using knowledge about the plant dynamics) so that the plant may be assumed to have reached steady-state before the next re-optimization. Another approach is to pose some conditions on the simulation variables (for example, the increments of all manipulated (all controlled) variables between two controller executions should be smaller than some specified value) to shorten the duration of the simulations. However, even though such tests are able to reduce the duration of the simulation they may not be accurate. Also for plants with slow dynamics, the two-level RTO-MPC simulations can still take long time despite the fact that the RTO has been executed right after steady-state was achieved.

These observations motivated the idea of substituting the dynamic plant operation between the steady-states with a calculation which yields the resulting closed-loop steady-state behavior in one execution. The advantage of using such a steady-state modelling method for process plants is not only the reduction of the excessive calculations, but also the possibility to integrate process and control systems into higher order optimization schemes.

Marlin and Young (1998) proposed a method for including controllers in steady-state simulation for multiple single-loop controllers. This method represents the steady-state controller algorithm within an open-form model as a set of equations which are solved simultaneously with the process model to find the steady-state plant behavior. The method is valid for the scenarios when the input has reached a constraint at steady-state and when it has not.

This method was extended in Kassidas *et al.* (2000). Here, the proposed procedure was designed for multi-input, multi-output control systems (nonsquare, in the general case) under centralized multivariable Dynamic Matrix Control. Similarly to Marlin and Young (1998), the effect of the controller was introduced as a set of nonlinear equations representing the relationship between MVs and CVs at steady-state. At the optimum point, these equations are deemed equivalent to the optimization problem which the controller solves at every iteration at steady-state.

This chapter proposes two methods for steady-state simulation of the non-square plant under constrained DMC control in the presence of mismatch between plant and model steady-state gains. First it is shown that in general, use of a steady-state model is insufficient, as the optimal inputs and outputs as computed by the MPC controller may vary over the prediction horizon, even at steady-state. Then, two methods for finding the closed-loop steady-state of a non-square plant in the general case are proposed. Finally, the inclusion of the steady-state calculation into a two level LP - MPC cascade control system simulation is presented.

This chapter considers the set point control problem only i.e no disturbances affect the plant during the transient and at steady-state. The modification of the proposed methods for closed-loop steady-state calculation in the presence of step-like disturbances is straightforward. Zero initial conditions (in deviation form) have been assumed for all simulations.

The problem formulation involving a static process model is presented in Section 4.1. The more general formulation that includes dynamics within the MPC controller as well as plant/model mismatch is presented in Section 4.2. A description and discussion of the sequential and simultaneous solution strategies are provided in Sections 4.3 and 4.4 respectively. This is followed by the LP-MPC application study in Section 4.5, and conclusions in Section 4.6.

4.1 Equilibrium Point Calculation using Steady-State Models

Assume that the plant has reached steady-state and the controller keeps solving the optimization problem (2.21) at every iteration. Then it is expected that its solution at every iteration is not any different from the solution at the previous iteration. In this case all the variables in (2.21) must have reached some steady-state values which remain constant from iteration to iteration.

If the system is square and closed-loop stable and if the set points given to the plant are achievable i.e. there are no steady-state offsets then by the integral action property of MPC, the steady-state inputs can be found trivially by:

$$y = y^{set}$$
$$u = K_p^{-1} y$$

where u and y are plant steady-state input and output vectors respectively and K_p is the plant gain matrix. However, if one or more manipulated variables is saturated or if the number of the outputs is larger than the number of the inputs, then off-set free tracking is not possible and the calculation of the closed-loop steady-state requires inclusion of controller equations and constraints.

Consider a process controlled by MPC that is at steady-state. If (i) the plant model is perfect, (ii) the plant is open-loop stable, and (iii) the input trajectory computed by the MPC algorithm is constant over control move horizon, then optimization problem (2.13) reduces to

$$\min_{y,u} J = \sum_{i=1}^{P} ||y - y^{set}||_{Q_i}^2 + \sum_{i=0}^{N-1} ||u - u^{tar}||_{R_i}^2$$

subject to:

$$\begin{aligned} x &= Ax + Bu \\ y &= Cx + Du \\ y^{min} &\leq y \leq y^{max} \\ u^{min} &\leq u \leq u^{max} \end{aligned} \tag{4.1}$$

where u and y correspond to the steady-state inputs and outputs respectively.

According to (2.8) the model equations in (4.1) may be replaced by

$$y = Ku$$

where K is the process steady-state gain. For $N = P$ and constant weighting matrices, Q_i and R_i, the objective function reduces to

$$\min_{y,u} J = ||y - y^{set}||_Q^2 + ||u - u^{tar}||_R^2$$

The ability of optimization problem (4.1) to find a plant's closed-loop equilibrium point was tested using the following case study:

Case Study 4.1. Consider a plant described by the transfer function model:

$$y_1(s) = \frac{0.3}{\tau s + 1}u_1(s) + \frac{0.4}{\tau s + 1}u_2(s)$$
$$y_2(s) = \frac{-0.2}{\tau s + 1}u_1(s) + \frac{0.5}{\tau s + 1}u_2(s)$$

Constraints: $-1.0 \leq y_1,\, y_2 \leq 1.0; \quad -0.5 \leq u_1,\, u_2 \leq 1.0$

Weights: $Q = I_p;\; R = 0$

Simulation parameters: sampling time $T_s = 0.3$, prediction horizon $P = 50$, control horizon $M = 2$, plant time constant $\tau = 3.0$

All possible combinations of the set points for y_1 and y_2 inside their steady-state limits ($-1.0 \leq y_1^{set},\, y_2^{set} \leq 1.0$) with mesh size of 0.1 were considered. The results, obtained from optimization procedure (4.1) were compared with the results obtained through dynamic simulations for corresponding set points and no discrepancies were revealed. The results for a number of different set-points are presented in Table 4.1.

Set point	Dynamic simulation		Solution of problem (4.1)	
$y^{set\,T}$	y^T	u^T	y^T	u^T
[0.8 0.6]	[0.63077 0.34615]	[0.76923 1.0]	[0.63077 0.34615]	[0.76923 1.0]
[0.8 0.8]	[0.53846 0.40769]	[0.46154 1.0]	[0.53846 0.40769]	[0.46154 1.0]
[0.8 1.0]	[0.44615 0.46923]	[0.15385 1.0]	[0.44615 0.46923]	[0.15385 1.0]
[0.6 0.8]	[0.4 0.5]	[0.0 1.0]	[0.4 0.5]	[0.0 1.0]
[0.6 1.0]	[0.30769 0.56154]	[−0.30769 1.0]	[0.30769 0.56154]	[−0.30769 1.0]

Table 4.1: Case Study 4.1: Steady-State Results via Dynamic Simulation and Solution of Problem (4.1)

Case Study 4.2. Case Study 4.1 was modified as follows:

$$
\begin{aligned}
y_1(s) &= \frac{0.3}{\tau s+1}u_1(s) + \frac{0.4}{\tau s+1}u_2(s) \\
y_2(s) &= \frac{-0.2}{0.5\tau s+1}u_1(s) + \frac{0.5}{\tau s+1}u_2(s)
\end{aligned}
$$

The only difference is in the dynamics of the transfer function element relating u_1 and y_2. The same set of scenarios as in Case Study 4.1 were run, with identical problem parameters except for the transfer function element described above. The results of the simulations for the set points from Table 4.1 are presented in Table 4.2.

Set point	Dynamic simulation		Solution of problem (4.1)	
$y^{set\,T}$	y^T	u^T	y^T	u^T
[0.8 0.6]	[0.64419 0.33721]	[0.81396 1.0]	[0.63077 0.34615]	[0.76923 1.0]
[0.8 0.8]	[0.5592 0.39387]	[0.53066 1.0]	[0.53846 0.40769]	[0.46154 1.0]
[0.8 1.0]	[0.47421 0.45053]	[0.24736 1.0]	[0.44615 0.46923]	[0.15385 1.0]
[0.6 0.8]	[0.41586 0.48943]	[0.05286 1.0]	[0.4 0.5]	[0.0 1.0]
[0.6 1.0]	[0.33087 0.54609]	[−0.23044 1.0]	[0.30769 0.56154]	[−0.30769 1.0]

Table 4.2: Case Study 4.2: Steady-State Results via Dynamic Simulation and Solution of Problem (4.1)

Analyzing the results it was concluded that here, optimization problem (4.1) no longer generates the correct steady-state result. Case Study 4.1 is somewhat artificial in that the dynamics of all the transfer function elements in the open-loop plant model are the same. From Case Study 4.2 it can be concluded that optimization problem (4.2) would not be expected to yield the correct result for the general case of differing dynamics in the elements of the transfer function matrix.

The reason for the discrepancies observed in Case Study 4.2 is that assumption (iii) in the formulation of optimization problem (4.1) is not valid for the general case, that is, the input and output trajectories in the MPC calculation cannot be assumed constant over the respective horizons, even with the system at steady-state. The closed-loop output trajectories for the set point change $y^{set} = [0.8 \quad 0.8]^T$ in Case Study 4.2 are shown in Figure 4.1, together with the output prediction trajectories at several points during the simulation. It can be seen that while the prediction trajectories do not change from one MPC calculation to another when the system is at steady-state, the values along the trajectory are not constant.

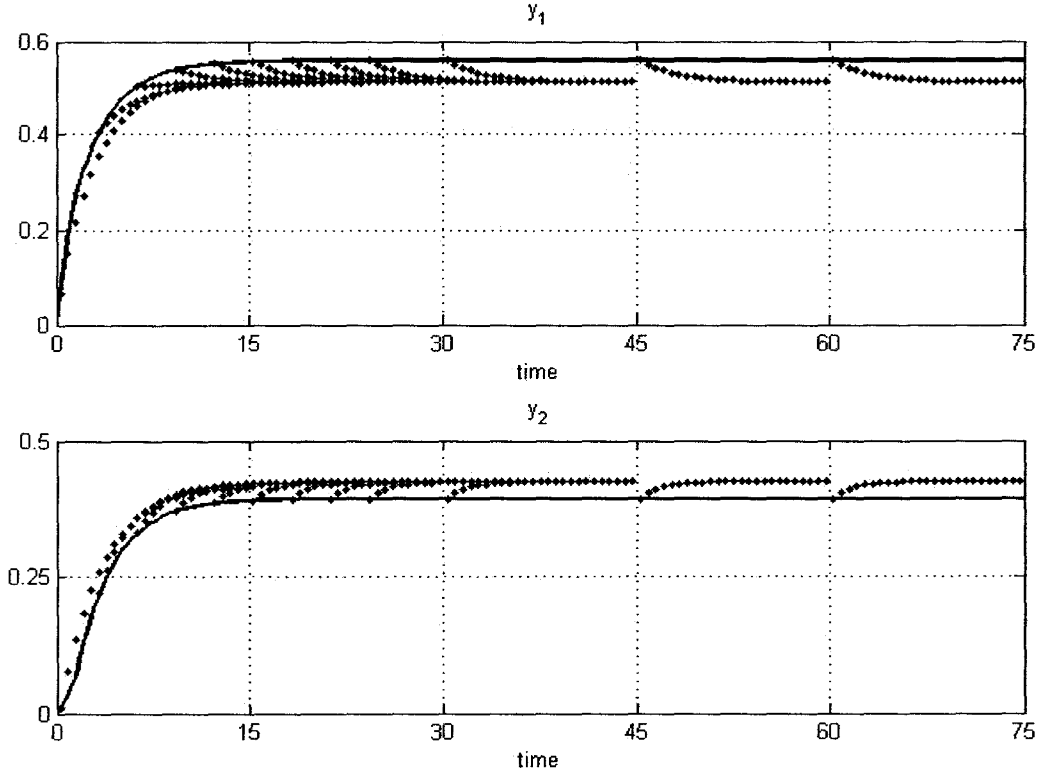


Figure 4.1: Case Study 4.2: Plant outputs and output prediction trajectories for $y^{set} = [0.8 \quad 0.8]^T$. Solid line: closed-loop output. Diamond: output prediction

The closed-loop steady-state is also dependent on the controller parameters in the general case. Table 4.3 shows the variation of steady-state for different values of the prediction horizon and sampling period for the set point $y^{set} = [0.8 \quad 0.8]^T$. All other parameters are as in Case Study 4.2.

P	10	30	50	70	90	110	130
y	0.51792	0.56478	0.55920	0.55301	0.54926	0.54697	0.54547
	0.42139	0.39015	0.39387	0.39799	0.40049	0.40202	0.40302
u	0.39306	0.54925	0.53066	0.51003	0.49754	0.48991	0.48491
	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T_s	0.05	0.1	0.15	0.2	0.25	0.3	0.35
y	0.5124	0.55424	0.56596	0.56619	0.56294	0.55920	0.55588
	0.42506	0.39718	0.38936	0.38921	0.39137	0.39387	0.39608
u	0.37468	0.51412	0.55320	0.55396	0.54314	0.53066	0.51959
	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 4.3: Case Study 4.2: Variation in Steady-State with Control Parameters for $y^{set} = [0.8 \quad 0.8]^T$

The invalidity of assumption (iii) mentioned above was also supported by the following observation. For the set points where the optimization problem (4.1) solution and dynamic simulations gave different results, the vector of future control moves $\Delta\hat{u}_k$ at steady-state was not a complete zero vector. The results for several simulations are presented in Table 4.4 (all simulations have been run with the Case Study 4.2 nominal parameters).

Set Points	y^T	u^T	$\Delta\hat{u}^T$
[0.8 0.6]	[0.64419 0.33721]	[0.81396 1.0]	[0.0 0.0 − 0.10185 0.0]
[0.8 0.8]	[0.5592 0.39387]	[0.53066 1.0]	[0.0 0.0 − 0.15741 0.0]
[0.8 1.0]	[0.47421 0.45053]	[0.24736 1.0]	[0.0 0.0 − 0.21297 0.0]
[0.6 0.8]	[0.41586 0.48943]	[0.05286 1.0]	[0.0 0.0 − 0.12037 0.0]
[0.6 1.0]	[0.33087 0.54609]	[−0.23044 1.0]	[0.0 0.0 − 0.17593 0.0]

Table 4.4: Case Study 4.2: Future Input Moves at Steady-State

Since the future input moves vector were taken at the steady-state, its first two components are zeros, as expected. However, the third element is non-zero. Simulations with larger control horizons have been considered and they showed that with an increase of the control horizon, the number of nonzero components in vector $\Delta\hat{u}_k$ at steady-state increases as well.

The steady-state also appeared to be dependent on the values of the move suppression weights $S = sI_m$. For the set-points considered in Table 4.4, the inclusion of these weights in the MPC results in reducing of the $l-\infty$ norm of the future input moves vector; however, it is not zero at steady-state even for large weights. The steady-states of Case Study 4.2 with the set point $y^{set} = [0.8 \quad 0.8]^T$ obtained through dynamic simulations for different values of s are displayed in Table 4.5. It can be seen that these equilibrium points are different from the solution of optimization problem (4.1) which is $y^T = [0.53846 \quad 0.40769]$, $u^T = [0.46154 \quad 1.0]$.

s	y^T	u^T	$\Delta\hat{u}^T$
0	[0.55920 0.39387]	[0.53066 1.0]	[0.0 0.0 − 0.15741 0.0]
2	[0.51782 0.42145]	[0.39273 1.0]	[0.0 0.0 − 0.00213 0.0]
4	[0.51740 0.42174]	[0.39132 1.0]	[0.0 0.0 − 0.00054 0.0]
6	[0.51732 0.42179]	[0.39106 1.0]	[0.0 0.0 − 0.00024 0.0]
8	[0.51729 0.42181]	[0.39097 1.0]	[0.0 0.0 − 0.00013 0.0]
10	[0.51728 0.42182]	[0.39092 1.0]	[0.0 0.0 − 0.00009 0.0]

Table 4.5: Case Study 4.2: Variation in Steady-State for Different Move Suppression Weights for $y^{set} = [0.8 \quad 0.8]^T$

All the simulations and observations presented above have clearly indicated that optimization procedure (4.1) cannot be used for the equilibrium point calculation in the general case. First, the vector of future manipulated variable moves may not be entirely zero and, therefore, it must be presented in the computational procedure in a complete form. Second, not only the steady-state equations but also the model dynamics determine the resulting steady-state point. Finally, the controller tuning parameters may affect the steady-state plant behavior. From these observations it can be deducted that for the general case, the effects of model dynamics have to be included in the calculation of the closed-loop steady-state.

4.2 Equilibrium Point Calculation using Dynamic Models

The simulations and the discussion presented in Section 4.1 have shown that the resulting equilibrium point depends on the controller's parameters and model. Therefore, the optimization problem which is solved by the controller at every iteration is taken as a starting point in computational framework design.

Let us consider optimization Problem 2.21. Assume, that the plant reached steady-state and the controller repeatedly calculates control inputs. Since the plant is at steady-state, then all the variables do not change from iteration to iteration and, therefore, (2.21) can be rewritten as follows:

$$\begin{aligned} \min_{\Delta\hat{u}} J &= ||\hat{y} - y^{set}||_Q^2 + ||\hat{u} - u^{tar}||_R^2 + ||\Delta\hat{u}||_S^2 \\ \text{subject to:} & \\ & \hat{y} = A^{**}x^m + B^{**}u + C^{**}\Delta\hat{u} + D_y^{*}d \\ & \hat{u} = D_u^{*}u + E^{*}\Delta\hat{u} \\ & y^{min} \leq \hat{y} \leq y^{max} \\ & u^{min} \leq \hat{u} \leq u^{max} \\ & \Delta u^{min} \leq \Delta\hat{u} \leq \Delta u^{max} \end{aligned} \tag{4.2}$$

where $\hat{y}$ is a vector of the steady-state output predictions, $\hat{u}$ is a vector of the steady-state future inputs, $u \in R^m$ is a vector of the manipulated variables at steady-state, $\Delta\hat{u}$ is a vector of the steady-state future manipulated variable moves, $x^m \in R^n$ is an estimation of the model state vector at steady-state, $d \in R^p$ is an estimation of the disturbance at steady-state.

Since the plant is at steady-state, the following equations must be satisfied:

$$\begin{aligned} x^m &= Ax^m + Bu \\ y^m &= Cx^m + Du \\ y &= K_p u \\ d &= y - y^m \end{aligned} \tag{4.3}$$

where all the parameters are as defined in Section 2.5.1. Controller solves problem (4.2) at every iteration, however, since the steady-state parameters are of interest, the steady-state condition must be satisfied:

$$\Delta\hat{u}(i) = 0, \qquad i = 1, ..., m \tag{4.4}$$

At steady-state the first control moves which are sent to the controller must be zero; however, as it has been observed, the other elements of the vector can be nonzero.

It is important to recognize that equations (4.3) and (4.4) cannot be merely inserted into optimization procedure (4.2), and the unknown steady-state inputs, u, computed together with $\Delta\hat{u}$. This destroys the integrity of what actually taking place in the control algorithm, resulting in erroneous solutions.

The importance of separating the steady-state model and plant equations from the optimization procedure can be illustrated using the following study. Optimization problem

(4.2) together with equations (4.3) and (4.4) results into the following optimization problem:

$$
\begin{aligned}
\min_{\hat{y},\hat{u},\Delta\hat{u},x^m,u,d} J &= ||\hat{y} - y^{set}||_Q^2 + ||\hat{u} - u^{tar}||_R^2 + ||\Delta\hat{u}||_S^2 \\
\text{subject to:} & \\
& \hat{y} = A^{**}x^m + B^{**}u + C^{**}\Delta\hat{u} + D_y^* d \\
& \hat{u} = D_u^* u + E^*\Delta\hat{u} \\
& x^m = Ax^m + Bu \\
& d = K_p u - Cx^m - Du \\
& \Delta\hat{u}(i) = 0, \qquad i = 1, ..., m \\
& y^{min} \leq \hat{y} \leq y^{max} \\
& u^{min} \leq \hat{u} \leq u^{max} \\
& \Delta u^{min} \leq \Delta\hat{u} \leq \Delta u^{max}
\end{aligned} \tag{4.5}
$$

This formulation contains controller tuning parameters as well as dynamics of the model. However, direct implementation of this calculation framework does not provide correct results for the steady-state parameters. This approach was applied to Case Study 4.2 with its nominal parameters and the results for different trials are summarized in Table 4.6.

Set point	Dynamic simulation		Solution of problem (4.5)	
$y^{set\,T}$	y^T	u^T	y^T	u^T
[0.8 0.6]	[0.64419 0.33721]	[0.81396 1.0]	[0.7 0.3]	[1.0 1.0]
[0.8 0.8]	[0.5592 0.39387]	[0.53066 1.0]	[0.7 0.3]	[1.0 1.0]
[0.8 1.0]	[0.47421 0.45053]	[0.24736 1.0]	[0.7 0.3]	[1.0 1.0]
[0.6 0.8]	[0.41586 0.48943]	[0.05286 1.0]	[0.57716 0.3819]	[0.59052 1.0]
[0.6 1.0]	[0.33087 0.54609]	[−0.23044 1.0]	[0.56661 0.38893]	[0.55537 1.0]

Table 4.6: Case Study 4.2: Steady-State Results via Dynamic Simulation and Solution of Optimization Problem (4.5)

This simulations show that integration of optimization procedure (4.2) and steady-state equations (4.3) into one optimization problem (4.5) is not the right approach of solving the problem. Indeed, even though the controller solves optimization problem (4.2) at every iteration at steady-state, the steady-state model equations in (4.5) are not part of the controller. Therefore, the entire problem must be reformulated in a way that conforms to operation in practice.

Assume that the steady-state input vector u is known. Then all the variables in steady-state equations (4.3) can be calculated and, therefore, they may also be considered as known. If all these parameters are known, then optimization problem (4.2) can be entirely expressed in terms of $\Delta\hat{u}$ only and solved. The solution of this optimization problem is vector $\Delta\hat{u}^{opt}$, and if the plant is at steady-state, its first m elements should be zeros: $\Delta\hat{u}^{opt}(i) = 0,\ i = 1, ..., m$.

The problem of equilibrium point calculation can be formulated in "reversed" order. Finding plant equilibrium point is equivalent to finding such steady-state values of the manipulated variables u that the substitution of all the variables derived from the steady-state equations (4.3) into optimization problem (4.2) results in an optimal solution $\Delta\hat{u}^{opt}$ whose first m elements are zero.

This thesis proposes two alternative methods of solving this problem. The first is to consider the problem as a system of nonlinear equations and solve it using numerical methods. The second method employs the KKT optimality conditions of the MPC optimization problem together with steady-state equations (4.3).

4.3 Equilibrium Point Calculation using Nonlinear Equation Solver

The procedure for calculation of the optimal MV moves can be presented as a nonlinear function "F" which argument and value are vectors of the same length. Indeed, if u is considered to be an argument of such a function, then all the variables in equations (4.3) are linear functions of u and, therefore, can be determined uniquely for any feasible value of u. Since only stable closed-loop systems are considered here, with proper choice of the weights in the objective function, optimization problem (4.2) has a unique solution - the optimal vector of future control moves. Then the first m elements of this vector (denoted as Δu) can be considered as the value of the function:

$$F(u) = \Delta u$$

This can also be considered as a "black box" where only input and output parameters are known. With such a representation, the problem of the equilibrium point calculation is transformed into the problem of solving a system of nonlinear equations. It is necessary to find such values of the argument that the value of the function F is equal to zero:

$$F(u^*) = 0 \tag{4.6}$$

Since the dimension of u is also m, the dependency between u and Δu can also be considered as a square system of nonlinear equations, though not expressed explicitly.

There are many methods of solving the system of nonlinear equations and most of them employ the derivatives of the equations with respect to the unknown variables. Since the equations in F are not given explicitly, these derivatives cannot be calculated analytically. However, the value of the function can be calculated for every feasible value of the argument which allows numerical calculation of the derivatives. Before proceeding to the method of solving the problem, it is worthwhile to devote some attention to studying the linearity of the problem. In case when the system is linear, or close to linear, the solution is expected to be rapid, precise and not strongly dependent on the initial guess. If the system is nonlinear, the convergence to the optimum is not guaranteed and the success of the solving procedure may be initial guess dependent.

This method will be examined using the following case study:

Case Study 4.3.

Model	*Plant*
$y_1(s) = \frac{0.25}{\tau s+1}u_1(s) + \frac{0.3}{\tau s+1}u_2(s)$	$y_1(s) = \frac{0.3}{\tau s+1}u_1(s) + \frac{0.4}{\tau s+1}u_2(s)$
$y_2(s) = \frac{-0.25}{0.5\tau s+1}u_1(s) + \frac{0.45}{\tau s+1}u_2(s)$	$y_2(s) = \frac{-0.2}{0.5\tau s+1}u_1(s) + \frac{0.5}{\tau s+1}u_2(s)$

Constraints: $-1.0 \leq y_1,\, y_2 \leq 1.0$; $-0.5 \leq u_1,\, u_2 \leq 1.0$

Weights: $Q = I_p$; $R = 0$; $S = 2.0\, I_m$

Simulation parameters: prediction horizon $P = 50$, control horizon $M = 2$, sampling time $T_s = 0.3$, plant/model time constant $\tau = 3$.

Since the equations are not known exactly, the linearity of the system will be studied using numerical methods. For this purpose, each component of the argument can be changed inside its bounds one in a time and the resulting curves can give an indication of the linearity of the system. From dynamic simulations with the set point $y^{set\,T} = [0.8\ 0.8]$, it was found that the equilibrium point is: $y^T = [0.43698 \quad 0.47534]$, $u^T = [0.12328 \quad 1]$. In the simulations, first u_1 was changed from 0 to 1 keeping $u(2)$ at the optimal level and then vice versa. Two scenarios were applied: with constraints on the input moves ($\Delta u \leq 0.2$) and with relaxed constraints on the input moves. The curves obtained are presented in Figure 4.2.

The results have shown that the constraints on the manipulated variable moves are an important issue here. If the constraints are tight, then large deviations of u around true plant steady-state u^* may result in the non-smooth function output (knotted line in Figure 4.2) degrading the linear properties of the function. However, it was found that these constraints do not affect the equilibrium point; thus they can be relaxed. In this case the saturation of the function outputs does not occur (solid line in Figure 4.2). As it can be seen in Figure 4.2, the case study considered has good linear properties in the absence of the constraints on the control moves. In the presence of such constraints the system shows piece-wise linear behavior.

Since the system has shown good linear properties, it is expected that the numerical methods for solving the system of nonlinear equations (4.6) will be successful. To solve the system, nonlinear equation solver "*fsolve*" from *Matlab*© Optimization Toolbox (The MathWorks User's Guide, 2006) was used. For medium-scale problems this solver uses the Gauss-Newton

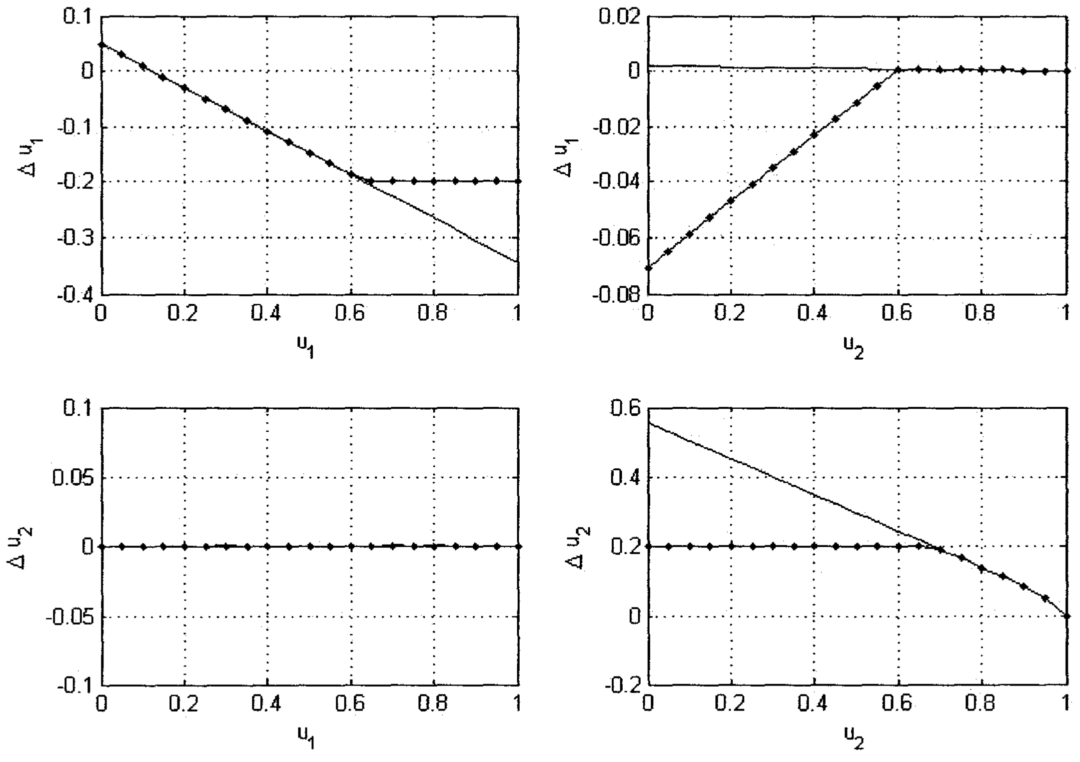


Figure 4.2: Case Study 4.3: Function F for different values of the input parameters ($y^{set\ T} =$ [0.8 0.8]). Solid line - responses without MVs moves constraints. Knot line - responses with input moves constraints

or Levenberg - Marquardt methods with line search and for large-scale optimization it uses algorithms which are based on nonlinear least-squares methods. The medium-scale algorithms were used in the case study presented.

The effect of the constraints on the manipulated variable moves was considered first. The method was executed for different values of the MV move constraints with a set of 49 initial guesses which were evenly distributed inside the manipulated variable limits. With constraints "$|\Delta\hat{u}| \leq 0.2$" only 8 initial guesses resulted in finding the correct steady-state (in all other cases the algorithm terminated at a point, which was not a root of the system). Some results for different starting points are presented in Table 4.7. When the constraints were relaxed to "$|\Delta\hat{u}| \leq 0.35$", 23 starting points were successful, for "$|\Delta\hat{u}| \leq 0.5$" 28 starting points from the set led to the algorithm convergence to the optimum. For MV move constraints "$|\Delta\hat{u}| \leq 1.0$", or more relaxed bounds the correct steady-state was found for all initial guesses from the set.

Initial Guess	y^T	u^T
[0.25 0.25]	[0.1372 0.14952]	[0.038226 0.31434]
[0.25 0.5]	[0.24961 0.25005]	[0.10776 0.54321]
[0.5 0.25]	[0.17135 0.18006]	[0.05935 0.38387]
[0.5 0.5]	[0.27643 0.27404]	[0.12434 0.59781]
[0.5 0.75]	[0.43698 0.47534]	[0.12328 1.0]
[0.75 1.0]	[0.625 0.35]	[0.75 1.0]

Table 4.7: Case Study 4.3: Steady-State Simulation using Solver "fsolve" and Different Initial Guesses for the Set Point $y^{set\,T} = [0.8\ 0.8]$ and $|\Delta\hat{u}| \leq 0.2$

This concludes that in order to achieve good performance of the method, the constraints on the manipulated variable moves in the optimization problem, which is solved inside function F, should be completely relaxed.

Next, Case Study 4.3 with the initial guess $[0.0\quad 0.0]^T$ was considered. Comparison of the equilibrium points calculated using *fsolve* solver and obtained through dynamic simulations for different set-points are presented in Table 4.8. Both the Gauss-Newton and Levenberg-Marquardt methods in the equation solver suite were used, and gave identical results.

The presented results conclude, that the proposed method of finding the equilibrium point using the nonlinear system solver is successful in case when the constraints on the MV moves are relaxed. Otherwise, its performance becomes sensitive to the initialization.

Set point	Dynamic simulation		Solution using *fsolve*	
$y^{set\,T}$	y^T	u^T	y^T	u^T
[0.6 0.6]	[0.45052 0.46632]	[0.16841 1.0]	[0.45052 0.46632]	[0.16841 1.0]
[0.6 0.8]	[0.32240 0.55173]	[−0.25867 1.0]	[0.32240 0.55173]	[−0.25867 1.0]
[0.8 0.4]	[0.69323 0.30451]	[0.97744 1.0]	[0.69323 0.30451]	[0.97744 1.0]
[0.8 0.6]	[0.56511 0.38993]	[0.55036 1.0]	[0.56511 0.38993]	[0.55036 1.0]
[0.8 0.8]	[0.43698 0.47534]	[0.12328 1.0]	[0.43698 0.47534]	[0.12328 1.0]
[0.8 1.0]	[0.30886 0.56076]	[−0.30379 1.0]	[0.30886 0.56076]	[−0.30379 1.0]

Table 4.8: Case Study 4.3: Comparison of the Dynamic Simulation and Steady-State Simulation using Nonlinear Equations Solver

4.4 Equilibrium Point Calculation using the Stationary Conditions of Optimality

The method represented in Section 4.3 considers the relationship between the steady-state values of the MV and the first MV moves as a nonlinear vector function. This function was formulated implicitly because the equations were not know exactly and the system operated as a "black box", where for each feasible argument ("input") the system generated the corresponding unique function value ("output"). Since the function contained the optimization problem inside, numerical methods for finding the function's zero had to be applied.

The method presented in this section formulates the problem of finding the equilibrium point as a solution of a system of nonlinear equations formulated explicitly. This approach replaces optimization problem (4.2) by a set of equations which are equivalent to the optimization at the optimal solution (Karush-Kuhn-Tucker (KKT) conditions of optimality).

First, let us consider optimization problem (4.2). Since the optimization variables are $\Delta\hat{u}$ and all the other variable are linearly dependent on $\Delta\hat{u}$, then (4.2) can be formulated in terms of $\Delta\hat{u}$ only:

$$\min_{\Delta\hat{u}} J = \Delta\hat{u}^T H \Delta\hat{u} + g^T \Delta\hat{u} + \alpha$$

subject to:

$$\begin{aligned} y^{min} &\leq \Omega_1 \Delta\hat{u} + \Gamma_1 \leq y^{max} \\ u^{min} &\leq \Omega_2 \Delta\hat{u} + \Gamma_2 \leq u^{max} \\ \Delta u^{min} &\leq \Delta\hat{u} \leq \Delta u^{max} \end{aligned} \tag{4.7}$$

where:

$$\begin{aligned} \Omega_1 &= C^{**} & \Gamma_1 &= A^{**} x^m + B^{**} u + D_y^* d \\ \Omega_2 &= E^{*} & \Gamma_2 &= D_u^* u \\ H &= \Omega_1^T Q \Omega_1 + \Omega_2^T R \Omega_2 + S & g &= 2\Omega_1^T Q (\Gamma_1 - y^{set}) + 2\Omega_2^T R (\Gamma_2 - u^{tar}) \\ \alpha &= (y^{set})^T Q y^{set} + (u^{tar})^T R u^{tar} \end{aligned} \tag{4.8}$$

The constant term in the objective function may be ommitted; this does not affect the solution. The KKT conditions for problem (4.7) can be derived in the following manner. The Lagrange function is first formed:

$$\begin{aligned} L = \quad & \Delta\hat{u}^T H \Delta\hat{u} + g^T \Delta\hat{u} + \\ & -(\lambda_1)^T \left(y^{max} - \Omega_1 \Delta\hat{u} - \Gamma_1\right) - (\lambda_2)^T \left(\Omega_1 \Delta\hat{u} + \Gamma_1 - y^{min}\right) - \\ & -(\lambda_3)^T \left(u^{max} - \Omega_2 \Delta\hat{u} - \Gamma_2\right) - (\lambda_4)^T \left(\Omega_2 \Delta\hat{u} + \Gamma_2 - u^{min}\right) \\ & -(\lambda_5)^T (\Delta u^{max} - \Delta\hat{u}) - (\lambda_6)^T (\Delta\hat{u} - \Delta\hat{u}^{min}) \end{aligned} \tag{4.9}$$

where $\lambda_1, ..., \lambda_6$ are Lagrange multipliers. The resulting KKT conditions for optimality are:

$$\begin{aligned}
\nabla_{\Delta\hat{u}} L = H\Delta\hat{u} + g + \Omega_1^T\lambda_1 - \Omega_1^T\lambda_2 + \Omega_2^T\lambda_3 - \Omega_2^T\lambda_4 + \lambda_5 - \lambda_6 &= 0 \\
y^{max} - \Omega_1\Delta\hat{u} - \Gamma_1 - v_1^U &= 0 \\
\Omega_1\Delta\hat{u} + \Gamma_1 - y^{min} - v_1^L &= 0 \\
u^{max} - \Omega_2\Delta\hat{u} - \Gamma_2 - v_2^U &= 0 \\
\Omega_2\Delta\hat{u} + \Gamma_2 - u^{min} - v_2^L &= 0 \\
\Delta\hat{u}^{max} - \Delta\hat{u} - v_3^U &= 0 \\
\Delta\hat{u} - \Delta\hat{u}^{min} - v_3^L &= 0 \\
(\lambda_1)_i(v_1^U)_i = 0;\ (\lambda_2)_i(v_1^L)_i = 0;\ i &= 1, ..., n_i \\
(\lambda_3)_j(v_2^U)_j = 0;\ (\lambda_4)_j(v_2^L)_j = 0;\ j &= 1, ..., n_j \\
(\lambda_5)_j(v_3^U)_j = 0;\ (\lambda_6)_j(v_3^L)_j = 0;\ j &= 1, ..., n_j \\
\lambda_1,\ \lambda_2,\ \lambda_3,\ \lambda_4,\ \lambda_5,\ \lambda_6,\ v_1^U,\ v_1^L,\ v_2^U,\ v_2^L,\ v_3^U,\ v_3^L &\geq 0
\end{aligned} \tag{4.10}$$

where $v_1^L, v_1^U, v_2^L, v_2^U, v_3^L, v_3^U$ are nonnegative slack variables, n_i is the dimension of vector λ_1 (or λ_2), n_j is the dimension of vector λ_3 (or λ_4, λ_5, λ_6) and $(\lambda_1)_i$ and $(v_1^U)_i$ are the $i-th$ elements in vectors λ_1 and v_1^U respectively with $(\lambda_2)_i$, $(\lambda_3)_j$, $(\lambda_4)_j$, $(\lambda_5)_j$, $(\lambda_6)_j$, $(v_1^L)_i$, $(v_2^U)_j$, $(v_2^L)_j$, $(v_3^U)_j$, $(v_3^L)_j$ similarly defined.

Since optimization problem (4.2) is a convex quadratic programming problem, a local optimum is a global optimum.

Finally, the problem of the equilibrium point calculation can be formulated in the following manner:

Find:

$$u,\ y$$

which together with associated problem variables (x^m, d, $\Delta\hat{u}$, *etc.*) satisfy the following system of nonlinear equations:

$$\textit{Steady-state plant/model equations:}$$

$$x^m = Ax^m + Bu$$

$$y = K_p u$$

$$d = y - Cx^m - Du$$

$$\textit{Karush-Kuhn-Tucker optimality conditions:}$$

$$H\Delta\hat{u} + g + \Omega_1^T\lambda_1 - \Omega_1^T\lambda_2 + \Omega_2^T\lambda_3 - \Omega_2^T\lambda_4 + \lambda_5 - \lambda_6 = 0$$

$$y^{max} - \Omega_1\Delta\hat{u} - \Gamma_1 - v_1^U = 0$$

$$\Omega_1\Delta\hat{u} + \Gamma_1 - y^{min} - v_1^L = 0$$

$$u^{max} - \Omega_2\Delta\hat{u} - \Gamma_2 - v_2^U = 0$$

$$\Omega_2\Delta\hat{u} + \Gamma_2 - u^{min} - v_2^L = 0$$

$$\Delta\hat{u}^{max} - \Delta\hat{u} - v_3^U = 0$$

$$\Delta\hat{u} - \Delta\hat{u}^{min} - v_3^L = 0$$

$$(\lambda_1)_i(v_1^U)_i = 0;\ (\lambda_2)_i(v_1^L)_i = 0;\ i = 1, ..., n_i$$

$$(\lambda_3)_j(v_2^U)_j = 0;\ (\lambda_4)_j(v_2^L)_j = 0;\ j = 1, ..., n_j$$

$$(\lambda_5)_j(v_3^U)_j = 0;\ (\lambda_6)_j(v_3^L)_j = 0;\ j = 1, ..., n_j$$

$$\textit{Non-negativity constraints:}$$

$$\lambda_1,\ \lambda_2,\ \lambda_3,\ \lambda_4,\ \lambda_5,\ \lambda_6,\ v_1^U,\ v_1^L,\ v_2^U,\ v_2^L,\ v_3^U,\ v_3^L \geq 0$$

$$\textit{Plant steady-state operation condition:}$$

$$\Delta\hat{u}(i) = 0,\ i = 1, ..., m \tag{4.11}$$

where A, B, C and D are as specified in (2.7); K_p as specified in (2.8); H, g, Ω_1, Ω_2, Γ_1, Γ_2 as specified in (4.8).

This framework uses the linear equation $y = K_p u$ which relates plant measured outputs with the manipulated variables at steady-state. However, instead of this equation, the non-linear plant steady-state fundamental equations (for example, obtained from material or energy balances) could be used here. This is also valid for all formulations proposed further which use steady-state plant equations and the conditions of optimality.

The steady-state inputs u may be determined by solving the system of equations (4.11). A standard nonlinear equation solver would in general not be able to accommodate the non-negativity constraints directly, and could result in convergence difficulties if one or more of the variables involved in the complementarity constraints in system (4.11) reaches zero prematurely. This has suggested the use of an interior point solution approach. *AMPL*© (Fourer *et al.*, 1993) was chosen as a modelling language with the IPOPT-CC solver (Raghunathan and Biegler (2003)) which has proven to be a reliable solver for such type of problems (Raghunathan and Biegler, 2003; Baker and Swartz, 2007).

Different case studies with different set points have been considered. All simulations have shown that the proposed method of the equilibrium point calculation gives the same parameters that were obtained from dynamic simulations with a required precision. For all simulations zero initial guesses was used. Some results are summarized in Table 4.9. Here, Case Study 4.3 with the nominal parameters were considered with the set points from Table 4.8.

Set point	Dynamic simulation		Solution of problem (4.11)	
$y^{set\ T}$	y^T	u^T	y^T	u^T
[0.6 0.6]	[0.45052 0.46632]	[0.16841 1.0]	[0.45052 0.46632]	[0.16841 1.0]
[0.6 0.8]	[0.32240 0.55173]	[−0.25867 1.0]	[0.32240 0.55173]	[−0.25867 1.0]
[0.8 0.4]	[0.69323 0.30451]	[0.97744 1.0]	[0.69323 0.30451]	[0.97744 1.0]
[0.8 0.6]	[0.56511 0.38993]	[0.55036 1.0]	[0.56511 0.38993]	[0.55036 1.0]
[0.8 0.8]	[0.43698 0.47534]	[0.12328 1.0]	[0.43698 0.47534]	[0.12328 1.0]
[0.8 1.0]	[0.30886 0.56076]	[−0.30379 1.0]	[0.30886 0.56076]	[−0.30379 1.0]

Table 4.9: Case Study 4.3: Comparison of the Dynamic Simulation and Steady-State Simulation using KKT Conditions for Optimality

4.5 Inclusion of the Steady-State Simulation into Two-Level Steady-State Optimization

As discussed in Section 2.1, the LP optimization problem can serve as an RTO level for a plant with an MPC control system. The LP objective function represents either a profit or cost of the operation, with output set points and input target values related through the steady-state gain matrix. The effects of plant/model mismatch and disturbances are compensated for through a bias term which is updated at steady-state (in the case of steady-state optimization) or more frequently.

In this section, steady-state optimization only is considered. That means that the LP is not executed until the plant has reached steady-state. At steady-state the bias term is measured in the following manner:

$$d = y - K_m u \tag{4.12}$$

Once the bias is updated, the LP recalculates the set points which are thereafter sent to the plant.

In the previous sections two methods for closed-loop steady-state calculation were presented. These methods allow calculation of the plant equilibrium point for a given control system and set points. Since the set points are sent to the plant from the LP level, it is possible to extend the steady-state calculation from the control system level to a two-level cascade system simulation.

Let us consider optimization problem (2.4) which the LP solves at steady-state. At the optimum point, this optimization problem is equivalent to the following KKT optimality conditions:

$$
\begin{aligned}
a + \lambda_1 + \lambda_2 - \lambda_3 &= 0 \\
b - K_m^T \lambda_1 + \lambda_4 - \lambda_5 &= 0 \\
y^{set} - K_m u^{tar} - d &= 0 \\
y^{max} - y^{set} - \nu_1^U &= 0 \\
y^{set} - y^{min} - \nu_1^L &= 0 \\
u^{max} - u^{tar} - \nu_2^U &= 0 \\
u^{tar} - u^{min} - \nu_2^L &= 0 \\
(\lambda_2)_i (\nu_1^U)_i = 0;\ (\lambda_3)_i (\nu_1^L)_i &= 0,\ \ i = 1, ..., n_i \\
(\lambda_4)_j (\nu_2^U)_j = 0;\ (\lambda_5)_j (\nu_2^L)_j &= 0,\ \ j = 1, ..., n_j \\
\lambda_2,\ \lambda_3,\ \lambda_4,\ \lambda_5,\ \nu_1^U,\ \nu_1^L,\ \nu_2^U,\ \nu_2^L &\geq 0
\end{aligned}
\tag{4.13}
$$

where $a^T = [a_1, ..., a_p]$ and $b^T = [b_1, ..., b_m]$ are the coefficients in the LP objective function.

Since in both systems (4.11) and (4.13) the number of the variables is equal to the number of the equations, they can be solved simultaneously. The system of nonlinear equations (4.13) is independent from system (4.11); therefore, its solution when it is solved separately is not any different from the solution when it is solved together with (4.11) as one system. However, the solution of (4.11) depends on the solution of (4.13), since it uses the set points and the target values calculated at the LP level as parameters. The simultaneous solution approach was followed in this study.

The results of several simulations for Case Study 4.3 are presented in Table 4.10, where the LP objective function coefficients are included. Similarly to the calculations in Section 4.4, these results were obtained using modelling language $AMPL^{©}$ and solver IPOPT-CC.

Because of the plant/model mismatch, it may require several iterations of the steady-state optimization before the plant starts operating at the optimal level. At every such iteration, the bias term d in the LP formulation will be updated according to (4.12) and the new

$f = [a^T \ b^T]$	$y^{set\,T}$	y^T	u^T
$[-4 \ -2 \ 2 \ 1]$	$[0.175 \ 0.575]$	$[0.21087 \ 0.55109]$	$[-0.5 \ 0.902174]$
$[-5 \ 1 \ 1 \ 1]$	$[0.55 \ 0.2]$	$[0.55 \ 0.2]$	$[0.847827 \ 73913]$
$[-1 \ -2 \ 1 \ 10]$	$[-0.275 \ -0.1]$	$[-0.275 \ -0.1]$	$[-0.423913 \ -0.369565]$

Table 4.10: Solutions of the Integrated LP - MPC System for Different LP Objective Functions

set points will be sent to the plant for implementation. The two-level LP-MPC cascade system with steady-state optimization can be successfully modelled as steady-state simulations only. The steady-state simulation of one iteration has been discussed above. The same approach can be used for steady-state simulation of several sequential iterations. The bias term d relates all the iterations with each other: at each iteration starting from the second, the bias is calculated using the variables from the previous iteration; at the first iteration it is zero. The entire two-level LP-MPC system with steady-state optimization can be formulated as the following steady-state simulation framework.

Find:

$$u_k, \ y_k, \ k = 1, ..., N$$

which satisfy the following system of nonlinear equations:

Steady-state plant/model equations:

$$x_k^m = Ax_k^m + Bu_k$$

$$y_k = K_p u_k$$

$$d_k = y_k - Cx_k^m - Du_k$$

Karush-Kuhn-Tucker conditions of optimality for the RTO level:

$$a + \lambda_{1\,k}^{L} + \lambda_{2\,k}^{L} - \lambda_{3\,k}^{L} = 0$$

$$b - K_m^T \lambda_{1\,k}^{L} + \lambda_{4\,k}^{L} - \lambda_{5\,k}^{L} = 0$$

$$y_k^{set} - K_m u_k^{tar} - d_k = 0$$

$$y^{max} - y_k^{set} - \nu_{1\,k}^{U} = 0$$

$$y_k^{set} - y^{min} - \nu_{1\,k}^{L} = 0$$

$$u^{max} - u_k^{tar} - \nu_{2\,k}^{U} = 0$$

$$u_k^{tar} - u^{min} - \nu_{2\,k}^{L} = 0$$

$$(\lambda_{2\,k}^{L})_i (\nu_{1\,k}^{U})_i = 0;\ (\lambda_{3\,k}^{L})_i (\nu_{1\,k}^{L})_i = 0,\ \ i = 1, ..., n_i$$

$$(\lambda_{4\,k}^{L})_j (\nu_{2\,k}^{U})_j = 0;\ (\lambda_{5\,k}^{L})_j (\nu_{2\,k}^{L})_j = 0,\ \ j = 1, ..., n_j$$

$$\lambda_{2\,k}^{L},\ \lambda_{3\,k}^{L},\ \lambda_{4\,k}^{L},\ \lambda_{5\,k}^{L},\ \nu_{1\,k}^{U},\ \nu_{1\,k}^{L},\ \nu_{2\,k}^{U},\ \nu_{2\,k}^{L} \geq 0$$

Control level steady-state equations:

$$\Omega_1 = C^{**};\ \Omega_2 = E^{*};\ H = \Omega_1^T Q \Omega_1 + \Omega_2^T R \Omega_2 + S$$

$$\Gamma_{1k} = A^{**} x_k^m + B^{**} u_k + D_y^* d_k;\ \Gamma_{2k} = D_u^* u_k$$

$$g_k = 2\Omega_1^T Q (\Gamma_{1k} - y_k^{set}) + 2\Omega_2^T R (\Gamma_{2k} - u_k^{tar})$$

Karush-Kuhn-Tucker conditions of optimality for the control level:

$$H\Delta\hat{u}_k + g_k + \Omega_1^T \lambda_{1k}^M - \Omega_1^T \lambda_{2k}^M + \Omega_2^T \lambda_{3k}^M - \Omega_2^T \lambda_{4k}^M + \lambda_{5k}^M - \lambda_{6k}^M = 0$$

$$y^{max} - \Omega_1 \Delta\hat{u}_k - \Gamma_{1k} - v_{1\,k}^U = 0$$

$$\Omega_1 \Delta\hat{u}_k + \Gamma_{1k} - y^{min} - v_{1\,k}^L = 0$$

$$u^{max} - \Omega_2 \Delta\hat{u}_k - \Gamma_{2k} - v_{2\,k}^U = 0$$

$$\Omega_2 \Delta\hat{u}_k + \Gamma_{2k} - u^{min} - v_{2\,k}^L = 0$$

$$\Delta\hat{u}^{max} - \Delta\hat{u}_k - v_{3\,k}^U = 0$$

$$\Delta\hat{u}_k - \Delta\hat{u}^{min} - v_{3\,k}^L = 0$$

$$(\lambda_{1k}^M)_i (v_{1\,k}^U)_i = 0;\ (\lambda_{2k}^M)_i (v_{1\,k}^L)_i = 0;\ i = 1, ..., n_i$$

$$(\lambda_{3k}^M)_j (v_{2\,k}^U)_j = 0;\ (\lambda_{4k}^M)_j (v_{2\,k}^L)_j = 0;\ j = 1, ..., n_j$$

$$(\lambda_{5k}^M)_j (v_{3\,k}^U)_j = 0;\ (\lambda_{6k}^M)_i (v_{3\,k}^L)_j = 0;\ j = 1, ..., n_j$$

Non-negativity conditions: $\lambda_{1k},\ \lambda_{2k},\ \lambda_{3k},\ \lambda_{4k},\ \lambda_{5k},\ \lambda_{6k},\ v_{1\,k}^U,\ v_{1\,k}^L,\ v_{2\,k}^U,\ v_{2\,k}^L,\ v_{3\,k}^U,\ v_{3\,k}^L \geq 0$

Plant steady-state operation condition:

$$\Delta\hat{u}_k(i) = 0,\ i = 1, ..., m$$

Equation for bias terms:

$$d_k = y_{k-1} - K_m u_{k-1},\ k \geq 2$$

Initialization:

$$d_1 = 0$$

where N is the number of iterations and the other parameters have been specified throughout the chapter. In this framework matrices Ω_1, Ω_2 and H can be calculated off-line and sent to the calculation procedure as parameters.

This framework was applied to find the steady-state parameters for a two-level LP-MPC control system. Case Study 4.3 was used as a lower level. The upper level LP had the following objective function: $f = [a^T b^T]^T = [-1.0\ \ -2.0\ 1.0\ 10.0]^T$. The results of the simulation

k	$y_k^{set\,T}$	y_k^T	u_k^T	d_k^T
1	$[-0.275 \;\; -0.1]$	$[-0.275 \;\; -0.1]$	$[-0.4239 \;\; -0.3696]$	$[0 \;\; 0]$
2	$[-0.3332 \;\; -0.1397]$	$[-0.3332 \;\; -0.1397]$	$[-0.4813 \;\; -0.4719]$	$[-0.0582 \;\; -0.0397]$
3	$[-0.3463 \;\; -0.1477]$	$[-0.3463 \;\; -0.1477]$	$[-0.4959 \;\; -0.4937]$	$[-0.0713 \;\; -0.0477]$
4	$[-0.3492 \;\; -0.1495]$	$[-0.3492 \;\; -0.1495]$	$[-0.4991 \;\; -0.4986]$	$[-0.0742 \;\; -0.0495]$
5	$[-0.3498 \;\; -0.1499]$	$[-0.3498 \;\; -0.1499]$	$[-0.4998 \;\; -0.4997]$	$[-0.0748 \;\; -0.0499]$
6	$[-0.35 \;\; -0.15]$	$[-0.35 \;\; -0.15]$	$[-0.5 \;\; -0.4999]$	$[-0.075 \;\; -0.05]$
7	$[-0.35 \;\; -0.15]$	$[-0.35 \;\; -0.15]$	$[-0.5 \;\; -0.5]$	$[-0.075 \;\; -0.05]$
8	$[-0.35 \;\; -0.15]$	$[-0.35 \;\; -0.15]$	$[-0.5 \;\; -0.5]$	$[-0.075 \;\; -0.05]$

Table 4.11: Case Study 4.3: Steady-State Evolution of LP-MPC Cascade Control System

are presented in Table 4.11. The steady-state simulation results have been confirmed by dynamic simulation of the same case study results of which are presented in Figure 4.3.

4.6 Chapter Summary

This chapter presented a computational procedure for determining the closed-loop steady-state point of a system controlled by constrained MPC. It was shown that in the general case not only the steady-state controller and plant equations determine the equilibrium point but also the controller tunings and model dynamics. Two methods for closed-loop steady-state simulation in the presence of plant/model mismatch have been presented and evaluated. The first method uses numerical solvers for finding a solution of a system of nonlinear equations expressed implicitly. It was shown that the performance of this method is strongly dependent on the posed constraints on the manipulated variable moves. The second method employs the KKT optimality conditions to formulate the problem as the solution of a system of nonlinear equations expressed explicitly. This system of equations was solved using the solver, IPOPT-C, and the proposed method was shown to be robust and initial guess insensitive. The success and convenience of its implementation allowed

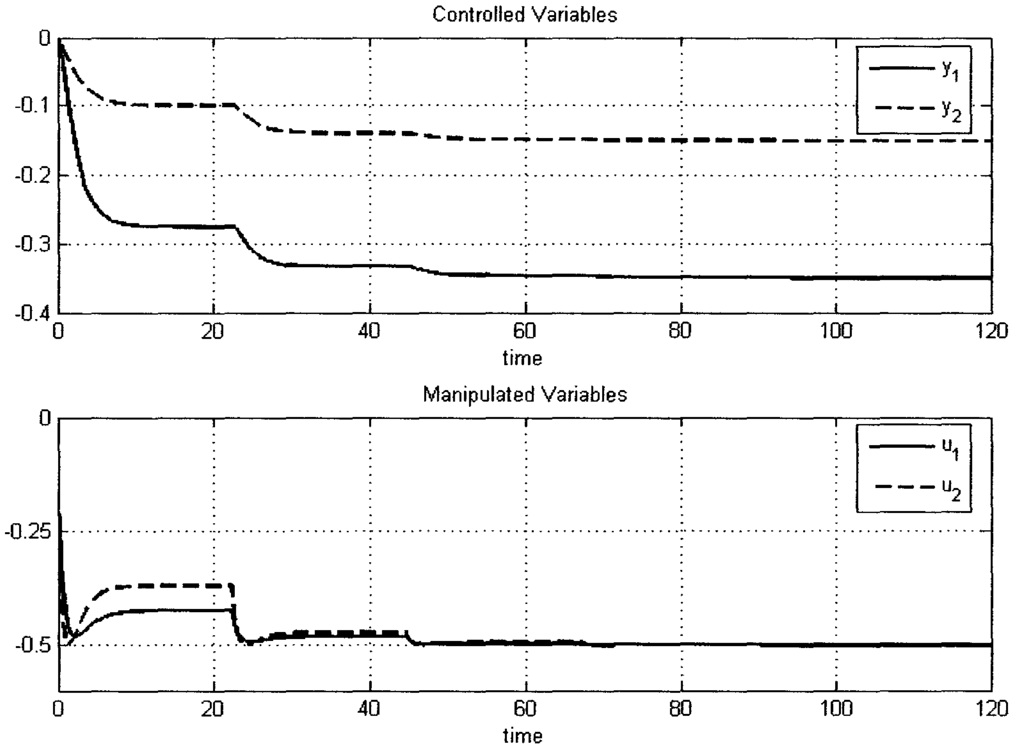


Figure 4.3: Dynamic responses for the multi-iteration steady-state simulation

the application of the method to be extended from the control system level to a two-level LP-MPC cascade steady-state simulation.

Chapter 5

LP Sensitivity Analysis

This chapter summarizes further research results obtained in the area of stability and performance of LP-MPC cascade control systems. Effects of constant disturbances and plant/model mismatch on the cascade system behavior have been considered in Chapter 3. For those cases implemented, simulations have shown that with a proper model update scheme at the LP level, the two-level control system appears to be stable and its performance depends on the combination of the size of the disturbance and the amount of plant/model mismatch. Such observations motivated the idea of considering the two-level control system behavior in the presence of nonconstant perturbations (usually, noise) instead of considering step-like disturbances. Stability of the deterministic two-level system can be achieved, if at steady-state plant/model mismatch and constant disturbances result in constant bias terms which are effectively introduced at the LP level through a proper bias update scheme. Constant bias terms result in a constant sequence of the set points from the LP level providing the stability of the two-level system. However, the presence of noise in channels implies that the resulting bias term is never constant and therefore two-level control system may result in significant variation in set points and possibly inputs and outputs as well. Before the effect of the noise on the entire two-level system is considered, it is worthwhile to study the effect of the noise on the LP solution, or LP sensitivity to the noise. This chapter summarizes the research results on this topic. First, the description of LP and MPC is given, then the LP sensitivity for SISO, MISO and MIMO cases is considered and, finally, the effect of the noise on the two-level control system is considered.

Section 5.1 gives introduction to LP sensitivity. Sections 5.2, 5.3 and 5.4 present LP sensitivity analysis and results for a SISO, MISO and MIMO system respectively. The summary of the chapter is given in Section 5.5.

5.1 Introduction to LP Sensitivity

For simplicity of analysis, an LP with one output (a MISO system) is considered first, since such a system has only one bias term in its formulation. This LP has the form:

$$\begin{aligned}
\min_{u_i,y} \quad & \alpha y^{ss} + \sum_{i=1}^{m} \beta_i u_i^{ss} \\
\text{subject to:} \quad & \\
& y^{ss} = \sum_{i=1}^{m} g_i^{ss} u_i^{ss} + d, \quad i = 1, ..., m \\
& y^{min} \le y^{ss} \le y^{max} \\
& u_i^{min} \le u_i^{ss} \le u_i^{max}, \quad i = 1, ..., m
\end{aligned} \tag{5.1}$$

where α and β_i are price coefficients and g_i^{ss} are steady-state gains of the process. According to the properties of an LP solution, the solution of problem (5.1) lies at the boundary of the feasible region, which is determined by the inequality constraints. If upper and lower bounds on the optimization variables and the coefficients in the cost function are given, then the particular placement of the solution depends only on the value of the bias d. From this viewpoint the LP problem can be considered as a mapping $\mathbb{R}^1 \Rightarrow \mathbb{R}^N$, where N is the number of optimization variables (for problem (5.1) $N = m + 1$) i.e. each feasible value of d corresponds to the solution of the LP problem which is a vector with length N.

Bias term d cannot be chosen randomly because for some values the LP is infeasible i.e. the value of d is such that it is impossible to find the solution which satisfies equality constraint $y^{ss} = \sum_{i=1}^{m} g_i^{ss} u_i^{ss} + d$ and preserve all the variables y and u_i inside their bounds at the same time. Therefore, it is first necessary to find the feasible region for d. This can be done by reformulating the optimization problem (5.1), posing d as an optimization variable. The upper bound for d can be found as:

$$\begin{cases} \max_d \quad d \\ \text{subject to:} \\ d = y^{ss} - \sum_{i=1}^{m} g_i^{ss} u_i^{ss} \\ y^{min} \leq y^{ss} \leq y^{max} \\ u_i^{min} \leq u_i^{ss} \leq u_i^{max} \end{cases} \Rightarrow \begin{cases} \max_{u_i, y} \quad y^{ss} - \sum_{i=1}^{m} g_i^{ss} u_i^{ss} \\ \text{subject to:} \\ y^{min} \leq y^{ss} \leq y^{max} \\ u_i^{min} \leq u_i^{ss} \leq u_i^{max} \end{cases} \tag{5.2}$$

The lower bound can be found analogously to (5.2) formulated as a minimization problem. If all steady-state gains g_i^{ss} are the same sign, the solution of optimization problems (5.2) can be written analytically:

$$\begin{array}{cc} g_i^{ss} > 0, \forall i & g_i^{ss} < 0, \forall i \\ d_{max} = y_{max} - \sum_{i=1}^{m} g_i^{ss} u_i^{min} & d_{max} = y_{max} - \sum_{i=1}^{m} g_i^{ss} u_i^{max} \\ d_{min} = y_{min} - \sum_{i=1}^{m} g_i^{ss} u_i^{max} & d_{min} = y_{min} - \sum_{i=1}^{m} g_i^{ss} u_i^{min} \end{array} \tag{5.3}$$

In all other cases the solution depends on the sign of each steady-state gain. Effect of the bias term on the LP solution can be illustrated using the following case study.

Case Study 5.1. MISO process with 2 inputs described by following model:

$$y(s) = \frac{0.4}{3s+1} u_1(s) + \frac{0.4}{3s+1} u_2(s)$$

with constraints:

$$-1.0 \leq y \leq 1.0; \quad -0.5 \leq u_1,\, u_2 \leq 0.5$$

Objective function:

$$\max_{y, u_1} f = y - u_1$$

The corresponding LP has the following form:

$$\max_{y,u_1} f = y - u_1$$

subject to:

$$\begin{aligned} & y = 0.4u_1 + 0.4u_2 + d \\ & -1.0 \leq y \leq 1.0 \\ & -0.5 \leq u_1 \leq 0.5 \\ & -0.5 \leq u_2 \leq 0.5 \end{aligned} \tag{5.4}$$

First, the range of the bias term d which corresponds to a feasible solution of the LP formulation (5.4) can be calculated. Since the steady-state gains are positive, formulas (5.3) can be used:

$$d_{max} = y_{max} - \sum_{i=1}^{m} g_i^{ss} u_i^{min} = 1.0 - 0.4(-0.5) - 0.4(-0.5) = 1.4$$

$$d_{min} = y_{min} - \sum_{i=1}^{m} g_i^{ss} u_i^{max} = -0.1 - 0.4(0.5) - 0.4(0.5) = -1.4$$

If the bias term is within this range, the solution of the LP exists. The dependance of this solution on the bias is presented graphically in Figure 5.1.

Each line in the figure represents an optimization variable. For any feasible value of d, at least two variables lie at their constraints. There are two particular values of bias ($d = -1.0$ and $d = 1.0$) where all three variables lie at their constraints. It is even more important to mention that at these points the constraints change their activity. When $d < -1.0$, constraint $-1 \leq y$ is active while $-0.5 \leq u_1$ is not active and when $d > -1.0$, constraint $-0.5 \leq u_1$ is active while $-1 \leq y$ is not active. This means that for any given bias value some constraints are active and some are not, and there are several special values of the bias where small deviations around these values cause such changes in the constraints. If the MPC controller's objective is to control the variables which are at their constraints and the steady-state bias value is equal to one of these values, then small changes in the bias (as an effect of white noise, for example) can cause some changes in the objective function which may result in poor control performance.

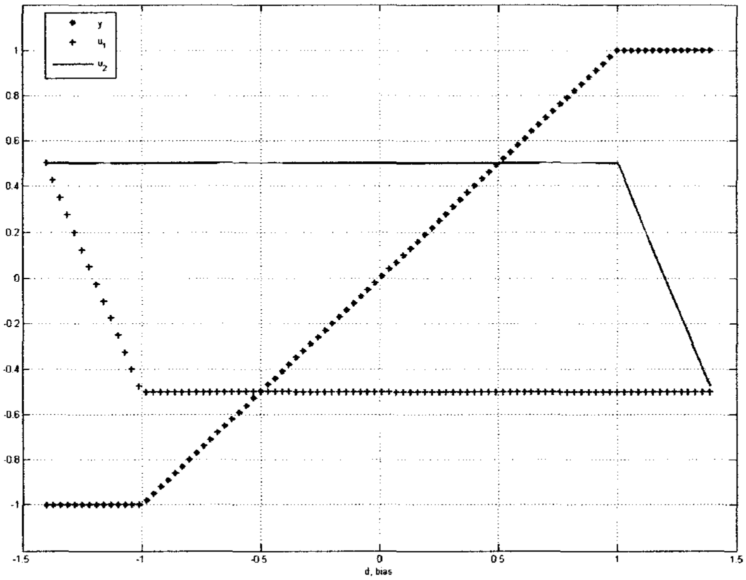


Figure 5.1: LP solution as a function of bias

When an optimization variable is not at a constraint, its value is a linear function of the bias. The slope of this function can be found using simple calculations. For $d < -1.0$, $y = y^{min} = -1.0$ and $u_2 = u^{max} = 0.5$. Then, the relation between u_1 and d can be expressed as follows:

$$\begin{aligned} & y = g_1 u_1 + g_2 u_2 + d \quad \Longrightarrow \quad u_1 = \frac{1}{g_1}(y - g_2 u_2 - d) \quad \Longrightarrow \\ & \Longrightarrow \quad u_1 = \frac{1}{g_1} y - \frac{1}{g_1} g_2 u_2 - \frac{1}{g_1} d \\ & u_1|_{d<-1.2} = \alpha d + \beta, \text{ where } \alpha = -\frac{1}{g_1},\ \beta = \frac{1}{g_1} y^{min} - \frac{1}{g_1} g_2 u_2^{max} \end{aligned} \tag{5.5}$$

From formula (5.5) it follows that this linear function has the opposite sign and a reciprocal value of the corresponding steady-state gain. Since the steady-state gains for u_1 and u_2 are the same ($g_1^{ss} = g_2^{ss} = 0.4$), their solution lines are collinear. Therefore, if a steady-state gain is small then the linear function which represents the solution for the corresponding input has a large slope and vice versa. It is important to take this observation into account because if the bias has a steady-state at the point where constraints change their activity and if the bias deviates around this value within some range, then the solution of the LP may have larger or smaller variation depending on the value of the corresponding steady-state gains.

From Figure 5.1 one can calculate that if y is not at the constraints its slope is equal to unity. This is consistent with the steady-state equation in (5.4) that if the output y is not at the constraints then it changes in exactly the same manner as the bias term d does. Analysis of the LP sensitivity will be conducted for three types of plant: SISO, MISO (1×2 plant) and MIMO (2×2 plant). This analysis aims to study the effect of the bias values on the LP solution and, therefore, on the performance and stability of the two-level control system.

5.2 Sensitivity Analysis of SISO System

Case Study 5.2. SISO process described by the following model:

$$y(s) = \frac{0.4}{3s+1} u(s)$$

with constraints:

$$-1.0 \leq y \leq 1.0; \quad -0.5 \leq u \leq 0.5$$

Objective function:

$$\max_{y} f = y$$

At the optimization level the process is described by the following LP problem:

$$\min_{y} f = -y$$

subject to:

$$y = 0.4u + d$$

$$-1.0 \leq y \leq 1.0$$

$$-0.5 \leq u \leq 0.5$$

The feasible range for the bias term d and the corresponding LP solutions are presented in Figure 5.2. From this figure it follows that if the bias is larger than 0.8 then the output is at its maximum constraint and if bias is less than 0.8, then the input is at its maximum constraint.

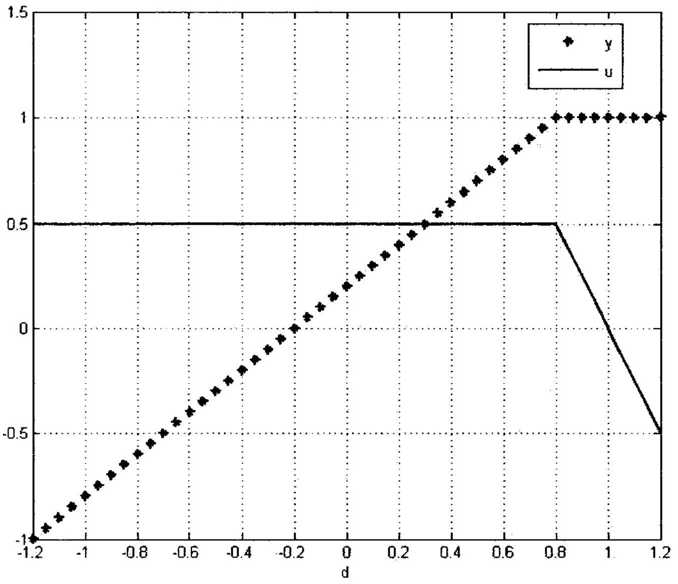


Figure 5.2: Case Study 5.2: LP solution as a function of bias

The results of the simulations are presented in Figure 5.3. As it was predicted when the bias is larger than 0.8, the output is at the maximum constraint and when the bias is smaller than 0.8, the input is at the maximum constraint. Also, in the figure one can see that when the output is not at the constraint it perfectly replicates the bias. However, when the input is not at the constraint, its deviation is larger than the bias deviation. As it was explained above, this happens because the steady-state gain is $g^{ss} = 0.4$ which corresponds to the slope of the solution for the input as $-1/g^{ss} = -2.5$ and, therefore, the input changes have the inverse direction and more than twice larger amplitude than the bias.

If the steady-state bias value is not 0.8 and it is further from 0.8 than the noise amplitude (constraints activity does not change) then only one variable deviates while another

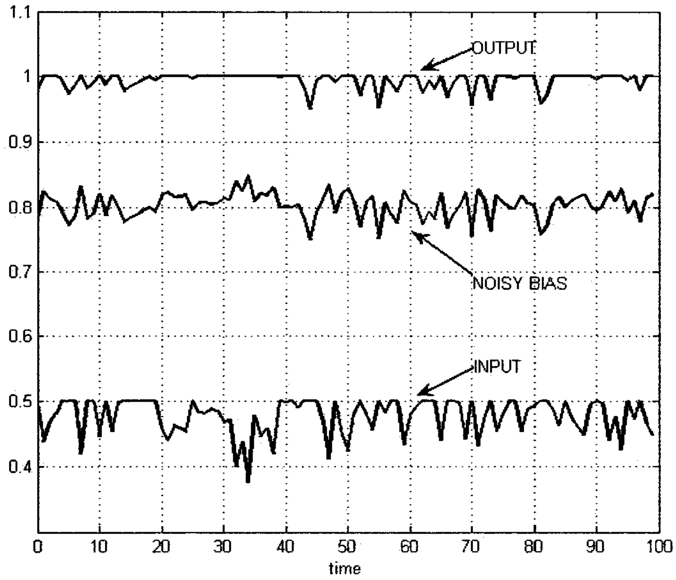


Figure 5.3: Case Study 5.2: Effect of the noisy bias on the LP solution

is at the constraint. If the input is at the constraint then the changes in the output are completely identical to the changes in the bias. If the output is at the constraint, then the input changes in the opposite direction to the bias changes and its amplitude is 2.5 time larger.

5.3 Sensitivity Analysis of MISO System. Effect of the Bias Noise on the Two-Level Cascade Control System Behavior.

Case study 5.3. For the case study here two-level LP-MPC control system with 1 output and 2 inputs and output noise was chosen. Graphically, the system is shown in Figure 5.4. In this case study, the model is perfect which means that the LP bias value is simply the steady-state bias and noise applied to the output.

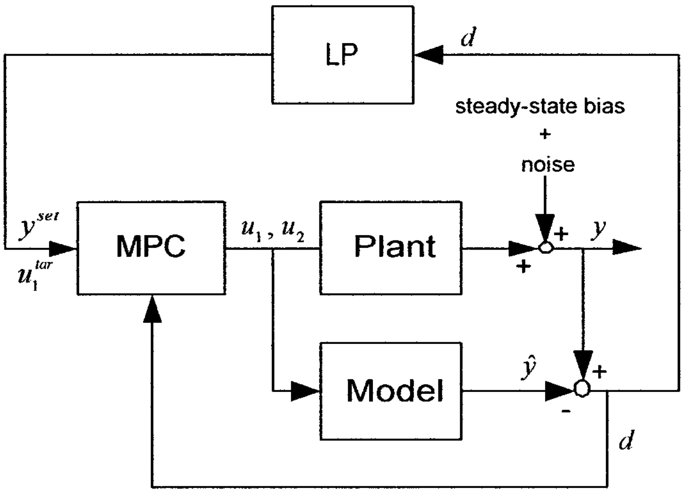


Figure 5.4: Two-level LP-MPC control system

Lower (MPC) level: The plant is described by the transfer function model:

$$y(s) = \frac{5.0}{3s+1}u_1(s) + \frac{2.0}{3s+1}u_2(s)$$

with constraints:

$$-1.0 \leq y \leq 1.0; \quad -0.5 \leq u_1,\, u_2 \leq 0.5$$

Controlled variables: y and u_1

Weights: $Q = I_p$, $R = 0$, $S = 5.0I_m$

Simulation parameters: sampling time $T_s = 0.3$, prediction horizon $P = 50$, control horizon $M = 2$

Upper (LP) level:

Objective function:

$$\max_{y,u_1} f = y - u_1$$

Constraints: identical to the lower-level constraints;

Model: identical to the lower-level plant model.

The corresponding LP has the following form:

$$\max_{y,u_1} f = y - u_1$$

subject to:

$$y = 5.0u_1 + 2.0u_2 + d$$

$$-1.0 \leq y \leq 1.0$$

$$-0.5 \leq u_1 \leq 0.5$$

$$-0.5 \leq u_2 \leq 0.5$$

The bias feasible range was calculated and the solutions of the LP for this range are presented in Figure 5.5. Now there are three regions and inside each region two optimization variables are at their constraints and one is not, and there are two points where active constraint set changes ($d = -2.5$ and $d = 2.5$).

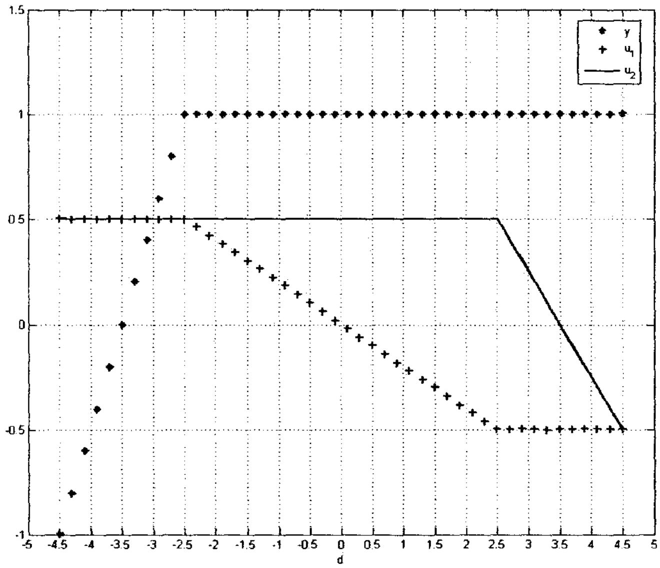


Figure 5.5: Case Study 5.3: LP solution as a function of bias

First, simulations with a perfect model and in the absence of any sustained disturbances were run. The results obtained are presented in Figure 5.6. Since the model is perfect and there are no disturbances, the bias steady-state value is zero. Zero steady-state bias

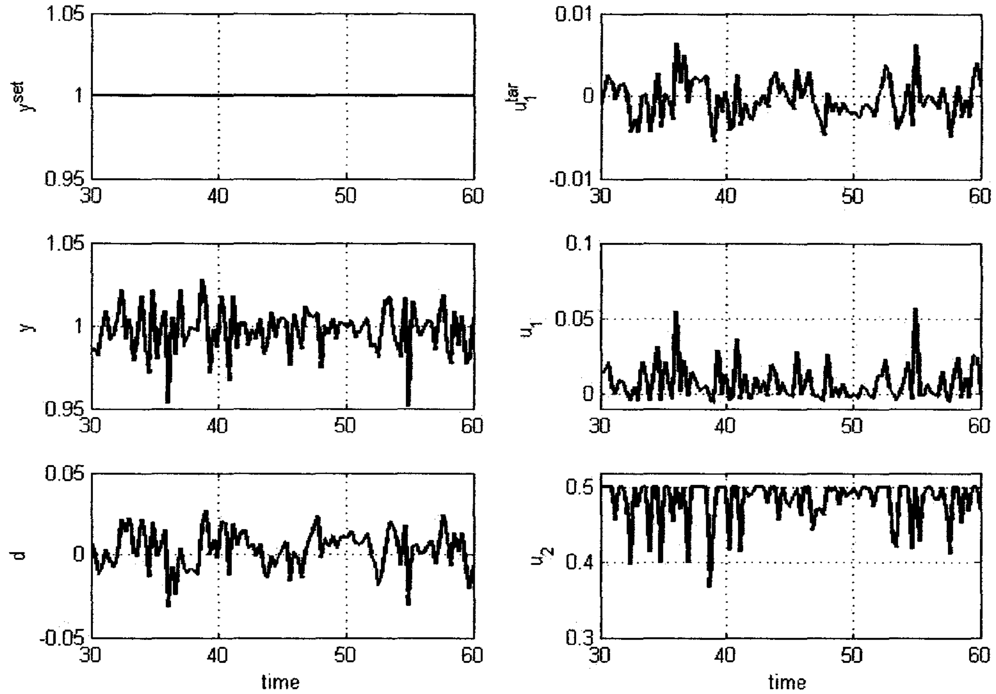


Figure 5.6: Two-level control system response in the presence of output white noise (perfect model and no step-like disturbances)

value corresponds to the following solution of the LP: $y = 1.0$, $u_1^{tar} = 0.0$; this is confirmed by the data in Figure 5.5. According to Figure 5.5, if the bias fluctuates around the zero steady-state, it should not make any changes in the set points for the output. Also, the same changes in the bias should cause changes in input u_1^{tar}. Since the steady-state gain $g_1^{ss} = 5.0$, the changes in u_1^{tar} are simply transformation of the bias values with coefficient -0.2. This is confirmed by the results presented in Figure 5.6 which shows the operation of the two-level control system where the upper level LP is executed at every iteration of MPC after the transients have died out. The performance of input u_2 which is used for the output control depends on the intensity of the white noise in the channel, since it must compensate it to keep the controlled variable at its set point.

In the next simulation, the steady-state bias value was chosen to be -2.5, since at this value the LP active constraint set changes. A steady-state bias value can be specified either by a step-like output disturbance, appropriate plant/model mismatch or a combination of both. The results of simulations are presented in Figure 5.7 (only operation after the transient is shown).

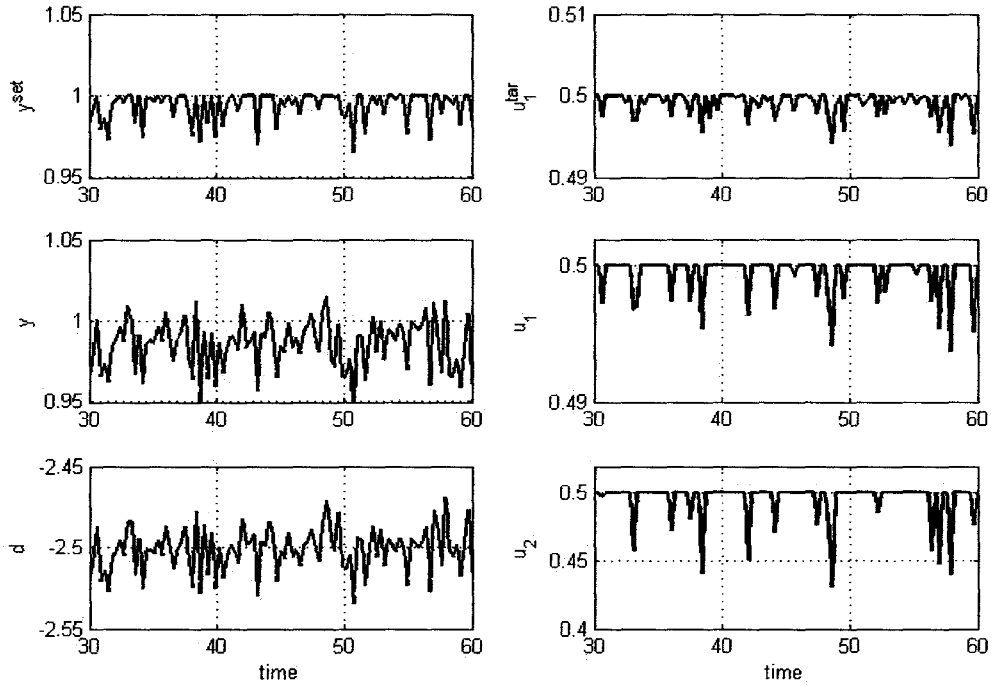


Figure 5.7: Two-level control system response in the presence of output white noise and output disturbance -2.5. Steady-state gains are larger than 1

According to Figure 5.5 when the bias is larger than -2.5, the output set point is at its maximum constraint. At the same time, the target value for input u_1 is not at the constraint and it is equal to the bias taken with coefficient -0.2. When the bias is smaller than -2.5, the output set point drifts away from the maximum constraint and, as it is expected, replicates the bias trajectory while the target value for input u_1 is at its constraint. All these observations are confirmed by Figure 5.7.

Now, for comparison, let us run the same simulation scenario using the model presented in Case Study 5.1 with the steady-state bias -1.0 (obtained from Figure 5.1). Simulation results are presented in Figure 5.8. The main difference between these two case studies is the values of the steady-state gains. If the steady-state gain is larger than unity, then the proportion between the changes in bias and changes in target values for this input is smaller than unity. This means that the bias variance is larger than the variance of the target values and vice versa.

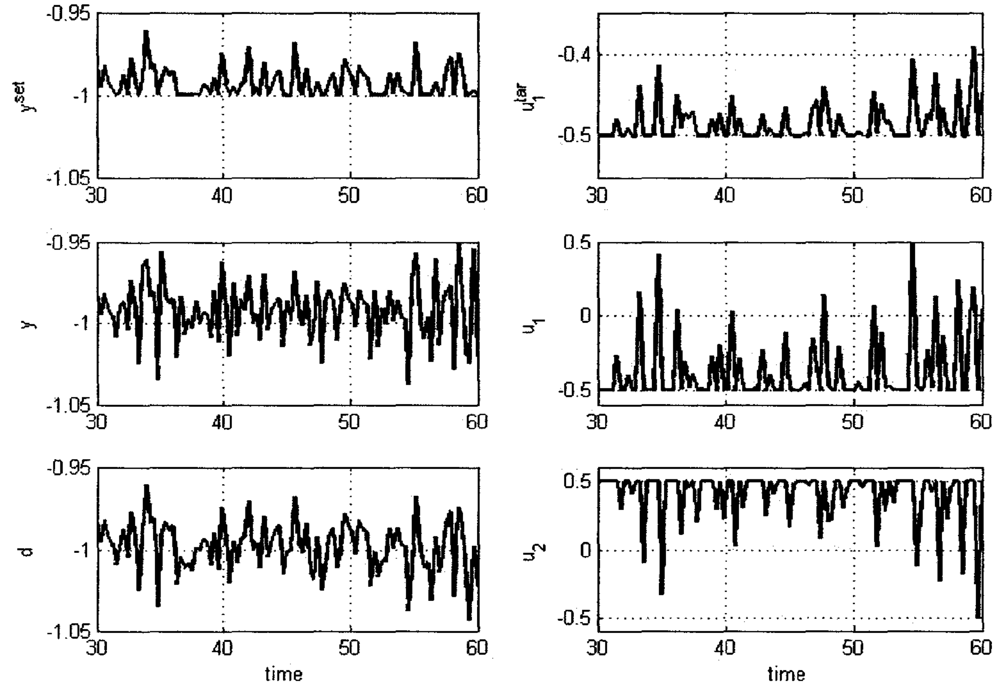


Figure 5.8: Two-level control system response in the presence of output white noise and output disturbance. Steady-state gains are smaller than 1

Figure 5.8 represents the case when the steady-state gains are smaller than unity. For bias values larger than -1.0, target value for u_1 is at the constraint and the output set point changes in the same manner as the bias. However, when bias values are smaller than -1.0 the input target values are not at the constraints and since the steady-state gain is smaller than unity, the proportion between changes in the bias values and changes in the input is larger than 1 ($|-1/0.4| = 2.5$) and, therefore, the variance of the target values for input u_1

is more than twice larger than the variance in the bias term. The fluctuating u_1^{tar} causes rapid fluctuation in u_1, which in turn affects the output. This results in poor performance in u_2 which is used to regulate y.

From these two simulations it can be concluded, that the steady-state gains and steady-state bias value may affect the two-level control system performance. If the output set point is not at the constraint, then its variation is totally determined by the variation of the noise in the channel. If the absolute value of the steady-state gain of the input which is associated with a target value is larger than unity, then its variation is smaller than the variation of the bias term. If the absolute value of the steady-state gain is smaller than unity, then the variation of the target value for this input is larger than the variation of the bias. If the variation of the input target value is sufficiently large, it can cause large variation of the inputs which are used for control. The LP formulation, together with the properties of the regulatory level which result in a steady-state bias, can cause poor performance of the manipulated variables.

Sensitivity of the two-level control system to input noise. Sometimes noise affects not only output channels but also plant inputs. For simplicity, let us consider 1×2 control system which has the following model:

$$y(s) = \frac{g_1^{ss}}{\tau_1 s + 1} u_1(s) + \frac{g_2^{ss}}{\tau_2 s + 1} u_2(s)$$

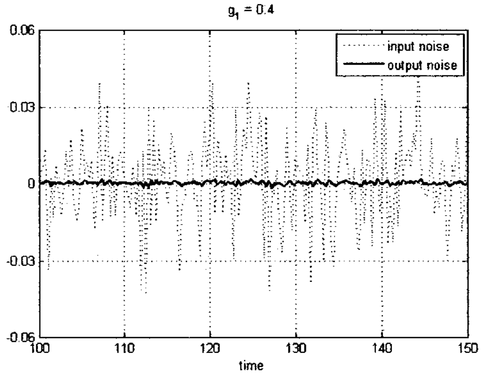
Assume, that input u_1 is subject to white noise of particular characteristics and it can be presented as a composition of deterministic and stochastic components:

$$u(s) = \bar{u}(s) + \alpha$$

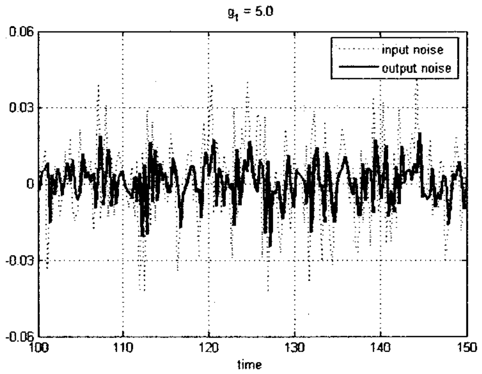
If such an input is introduced into the plant, then the resulting output has the following form:

$$\begin{aligned} y(s) &= \frac{g_1^{ss}}{\tau_1 s + 1} u_1(s) + \frac{g_2^{ss}}{\tau_2 s + 1} u_2(s) = \frac{g_1^{ss}}{\tau_1 s + 1} \left(\bar{u}(s) + \alpha \right) + \frac{g_2^{ss}}{\tau_2 s + 1} u_2(s) = \\ &= \frac{g_1^{ss}}{\tau_1 s + 1} \bar{u}(s) + \frac{g_2^{ss}}{\tau_2 s + 1} u_2(s) + \frac{g_1^{ss}}{\tau_1 s + 1} \alpha \end{aligned} \tag{5.6}$$

From formula (5.6) it follows that the steady-state effect of stochastic component α on the output depends on the absolute value of the steady-state gain g_1^{ss}. If it is larger than unity, then input noise will be modified passing through the plant and its variance may increase resulting in the output disturbance with larger variance. If the absolute value of the steady-state gain is smaller than unity then the variance of the noise decreases as it passes through the plant. An important observation here is that input noise does not shift the bias steady-state value and may affect its variance. In the simulations presented here, the noise affected the plant inputs only while the inputs which were used for the prediction calculation remained uncorrupted. This scenario is not quite realistic because in practice model predictions are calculated using the measurements of the inputs which were injected into the plant and, therefore, already contained noise.



(a) Absolute value of the steady-state gain is smaller than unity.



(b) Absolute value of the steady-state gain is larger than unity.

Figure 5.9: Effect of the input noise on the bias term

In Case Study 5.3, white noise of variance $3.4526 \cdot 10^{-4}$ was applied to input u_1. In the control system without an LP (with set points: $y_{set} = 0.1$ and $u_1^{tar} = 0.1$) at steady-state the variance of the output was $7.9694 \cdot 10^{-5}$ which is smaller than the variance of the noise, even though the steady-state gains were larger than unity. In case of the two-level control system with the same noise applied to input u_1, the variance of the output was $8.1415 \cdot 10^{-5}$ which is also smaller than the variance of the noise. The same scenario was applied to Case Study 5.1. Without LP implementation, for the same set points, the variance of the output was $5.1004 \cdot 10^{-7}$ which was significantly smaller that the noise variance. In the two-level control system, the output noise variance was $1.8080 \cdot 10^{-6}$. The comparison of the outputs for Case Study 5.1 and Case Study 5.3 for the same set points and input noise presented in Figure 5.9. Figure 5.9(b) shows that even though the steady-state gains are larger than unity, the variation of the input noise is larger than the variation in the output which it caused.

5.4 Sensitivity Analysis of MIMO System

The sensitivity of the LP level in a MIMO system is studied using a 2×2 system. Consider a 2×2 system described using the following model:

$$\begin{aligned}
y_1(s) &= \frac{g_{11}^{ss}}{\tau_{11}s+1}u_1(s) + \frac{g_{12}^{ss}}{\tau_{12}s+1}u_2(s) \\
y_2(s) &= \frac{g_{21}^{ss}}{\tau_{21}s+1}u_1(s) + \frac{g_{22}^{ss}}{\tau_{22}s+1}u_2(s) \\
y_1^{min} &\leq y_1 \leq y_1^{max} \\
y_2^{min} &\leq y_2 \leq y_2^{max} \\
u_1^{min} &\leq u_1 \leq u_1^{max} \\
u_2^{min} &\leq u_2 \leq u_2^{max}
\end{aligned}$$

The LP optimization level for such a system in the general case has the following formulation:

$$\min_{u_1,u_2,y_1,y_2} \quad \alpha_1 y_1 + \alpha_2 y_2 + \beta_1 u_1 + \beta_2 u_2$$

subject to:

$$\begin{aligned}
y_1 &= g_{11}^{ss}u_1 + g_{12}^{ss}u_2 + d_1 \\
y_2 &= g_{21}^{ss}u_1 + g_{22}^{ss}u_2 + d_2 \\
y_1^{min} &\leq y_1 \leq y_1^{max} \\
y_2^{min} &\leq y_2 \leq y_2^{max} \\
u_1^{min} &\leq u_1 \leq u_1^{max} \\
u_2^{min} &\leq u_2 \leq u_2^{max} \qquad (5.7)
\end{aligned}$$

The major difference between this formulation and the formulations considered previously is that two bias terms are now involved, and they both are updated at every iteration.

Problem 5.7 has many parameters and their effect on the LP solution should be studied first. Since optimization variables y_1 and y_2 relate to u_1 and u_2 through the system of equality constraints, the entire optimization problem can be expressed in terms of u_1 and u_2 only using direct substitution. Then, optimization problem (5.7) can be reformulated as:

$$\min_{u_1,u_2} \quad f = \alpha_1(g_{11}^{ss}u_1 + g_{12}^{ss}u_2 + d_1) + \alpha_2(g_{21}^{ss}u_1 + g_{22}^{ss}u_2 + d_2) + \beta_1 u_1 + \beta_2 u_2$$

subject to:

$$y_1^{min} \leq g_{11}^{ss}u_1 + g_{12}^{ss}u_2 + d_1 \leq y_1^{max}$$
$$y_2^{min} \leq g_{21}^{ss}u_1 + g_{22}^{ss}u_2 + d_2 \leq y_2^{max}$$
$$u_1^{min} \leq u_1 \leq u_1^{max}$$
$$u_2^{min} \leq u_2 \leq u_2^{max}$$

After simplification the optimization problem formulation is:

$$\min_{u_1,u_2} \quad f = (\alpha_1 g_{11}^{ss} + \alpha_2 g_{21}^{ss} + \beta_1)\, u_1 + (\alpha_1 g_{12}^{ss} + \alpha_2 g_{22}^{ss} + \beta_2)\, u_2 + \alpha_1 d_1 + \alpha_2 d_2$$

subject to:

$$y_1^{min} \leq g_{11}^{ss}u_1 + g_{12}^{ss}u_2 + d_1 \leq y_1^{max}$$
$$y_2^{min} \leq g_{21}^{ss}u_1 + g_{22}^{ss}u_2 + d_2 \leq y_2^{max}$$
$$u_1^{min} \leq u_1 \leq u_1^{max}$$
$$u_2^{min} \leq u_2 \leq u_2^{max}$$

This could also be written as:

$$\min_{u_1,u_2} \quad f = (\alpha_1 g_{11}^{ss} + \alpha_2 g_{21}^{ss} + \beta_1)\, u_1 + (\alpha_1 g_{12}^{ss} + \alpha_2 g_{22}^{ss} + \beta_2)\, u_2$$

subject to:

$$\begin{aligned}
&\text{pair 1} \quad \begin{cases} u_2 \le -\frac{g_{11}^{ss}}{g_{12}^{ss}} u_1 + \frac{y_1^{max} - d_1}{g_{12}^{ss}} \\ u_2 \ge -\frac{g_{11}^{ss}}{g_{12}^{ss}} u_1 + \frac{y_1^{min} - d_1}{g_{12}^{ss}} \end{cases} \\
&\text{pair 2} \quad \begin{cases} u_2 \le -\frac{g_{21}^{ss}}{g_{22}^{ss}} u_1 + \frac{y_2^{max} - d_2}{g_{22}^{ss}} \\ u_2 \ge -\frac{g_{21}^{ss}}{g_{22}^{ss}} u_1 + \frac{y_2^{min} - d_2}{g_{22}^{ss}} \end{cases} \\
&u_1^{min} \le u_1 \le u_1^{max} \\
&u_2^{min} \le u_2 \le u_2^{max}
\end{aligned} \tag{5.8}$$

Term $\alpha_1 d_1 + \alpha_2 d_2$ in the objective function is omitted because it is constant for any given biases d_1 and d_2 and, it does not affect the solution.

From formulation (5.8) several conclusions can be made regarding the feasible region of the problem and possible placement of its solution. First, on the feasible region plane the slope of the line which represents the objective function (so-called "isocost line" (Winston, 2004)) does not depend on the bias terms. The slope of the objective function depends only on the cost coefficients corresponding to every input and output variable and the steady-state gains of the process. Second, the output inequality constraints are presented in pairs where each pair consists of two collinear constraints. The slope of each pair of constraints depends on the ratio of the steady-state gains: g_{i1}^{ss}/g_{i2}^{ss}, $i = 1, 2$. If the steady-state gains g_{i1}^{ss} and g_{i2}^{ss} are the same sign then the slope of the constraint is negative and vice versa. The distance between the constraints in each pair does not depend on the bias term and is only determined by the steady-state gains and the original output constraints. The distances between the constraints in pairs are:

$$D_1 = \frac{y_1^{max} - d_1}{g_{12}^{ss}} - \frac{y_1^{min} - d_1}{g_{12}^{ss}} = \frac{y_1^{max} - y_1^{min}}{g_{12}^{ss}}$$

$$D_2 = \frac{y_2^{max} - d_2}{g_{22}^{ss}} - \frac{y_2^{min} - d_2}{g_{22}^{ss}} = \frac{y_2^{max} - y_2^{min}}{g_{22}^{ss}}$$

If d_1 is an arithmetic mean of y_1^{max} and y_1^{min} (or d_2 for the pair y_2^{max} and y_2^{min}) then this pair of constraints is symmetrical around the origin. Otherwise it is shifted up (if d_i is closer to

y_i^{min}) or down (if d_i is closer to y_i^{max}), on a u_2 versus u_1 plot (see Figure 5.10). The feasible area for each pair of constraints lies between the constraint lines. Third, the minimum and maximum constraints for the input variables form a rectangle with the feasible area inside. The resulting feasible region of the problem is the intersection of all constraints and it is schematically presented in Figure 5.10.

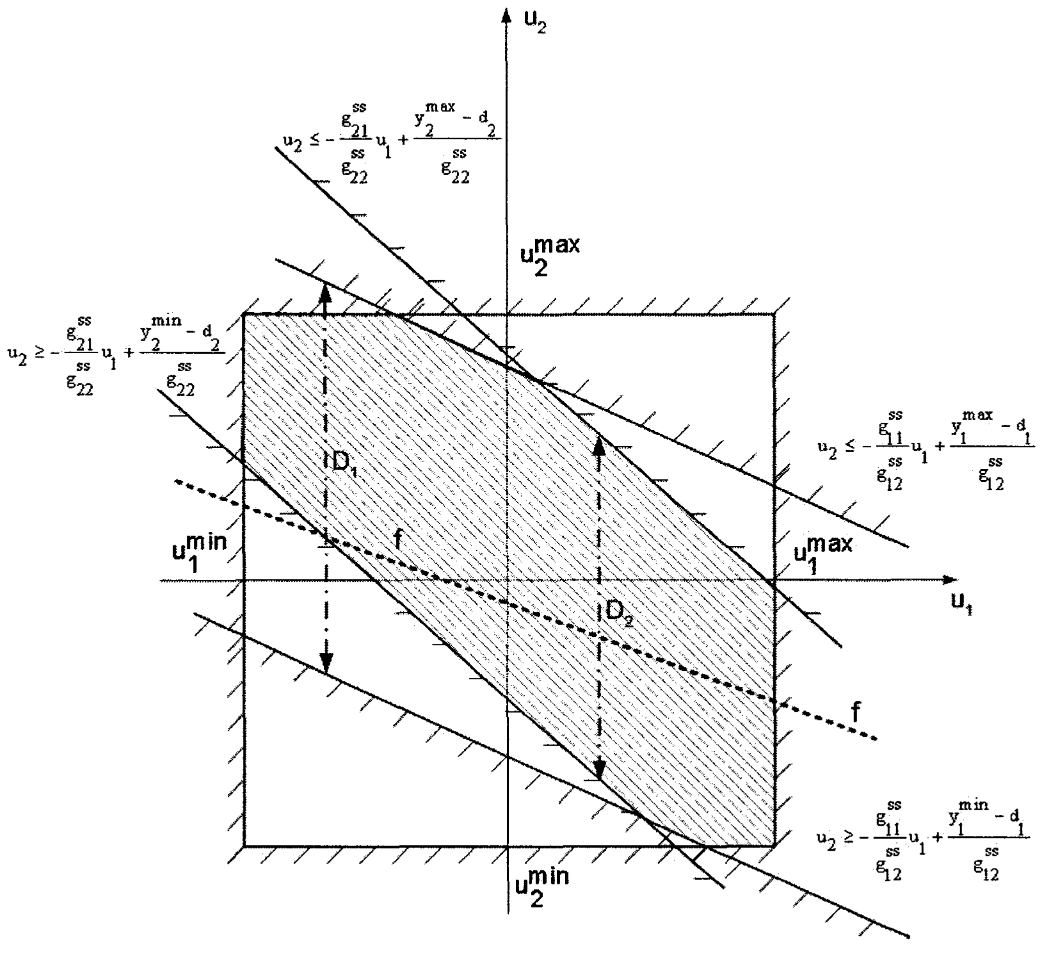


Figure 5.10: Graphical representation of the LP optimization problem for a 2×2 system

The solution of the optimization problem can be found by shifting the isocost line inside the feasible region in the direction of decreasing cost until it cannot be improved any further. The resulting position of the isocost line will determine the optimum values of the optimization variables. At every iteration the biases d_1 and d_2 are updated and therefore the shape of the feasible region changes, and so does the solution of the optimization problem. As mentioned above, none of the slopes depends on the bias terms and the feasible region changes only because the constraint lines go up and down. Therefore, the LP problem solution sensitivity can be analyzed off-line if the steady-state process gains and the variable constraints are known.

5.4.1 Sensitivity analysis of a 2×2 system. Case 1.

Assume that two output constraints intersect as shown in Figure 5.11(a) and they bound the feasible region of the problem. If the combination of the steady-state gains is such that constraints are almost collinear and the isocost line slope is such that the solution of the optimization problem lies in the point of intersection of the constraints, then the LP solution is expected to be sensitive to the small changes in the biases. If one of the biases slightly changes, the constraint lines will shift up or down. Since the constraints are almost collinear, new intersection (and, therefore, new solution) will appear in some distance from the previous solution. Small bias fluctuations can make the LP solution migrate over a large region.

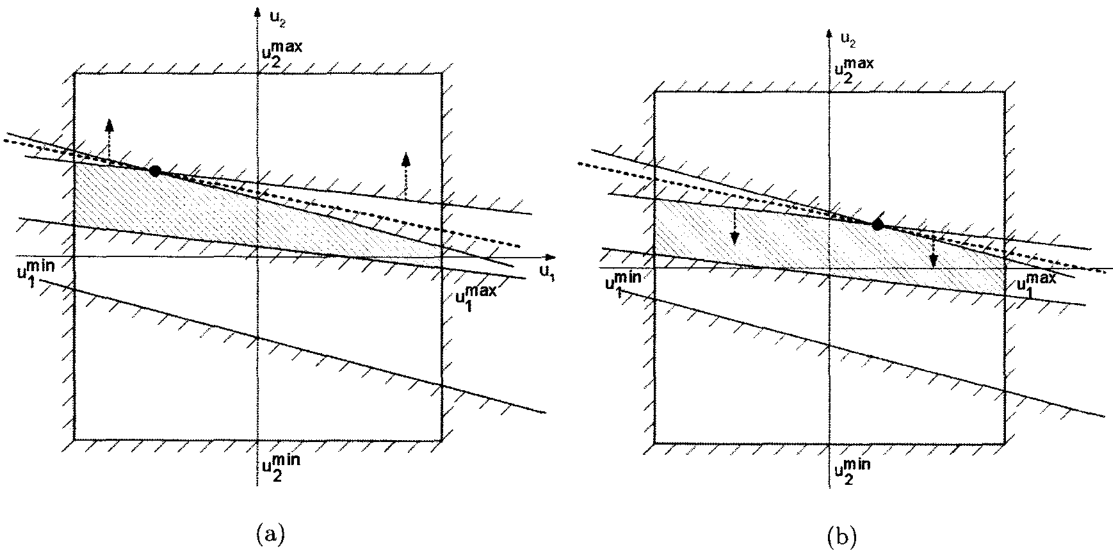


Figure 5.11: Effect of the bias on the LP solution: Case 1

Such a scenario was simulated using the following case study.

Case Study 5.4. Model:

$$y_1(s) = -\frac{1.2}{3s+1}u_1(s) - \frac{14.4}{3s+1}u_2(s)$$
$$y_2(s) = \frac{1.3}{3s+1}u_1(s) + \frac{6.5}{3s+1}u_2(s)$$

with constraints:

$$-1.0 \leq y_1 \leq 1.0$$
$$-1.0 \leq y_2 \leq 1.0$$
$$-0.5 \leq u_1 \leq 0.5$$
$$-0.5 \leq u_2 \leq 0.5$$

Objective function:

$$\min_{y_1,u_2} \quad f = 0.9y_1 + 1.0y_2 - 2.0u_1 - 8.3733u_2$$

The corresponding LP has the following form:

$$\min_{y_1,u_2} \quad f = 0.9y_1 + 1.0y_2 - 2.0u_1 - 8.3733u_2$$

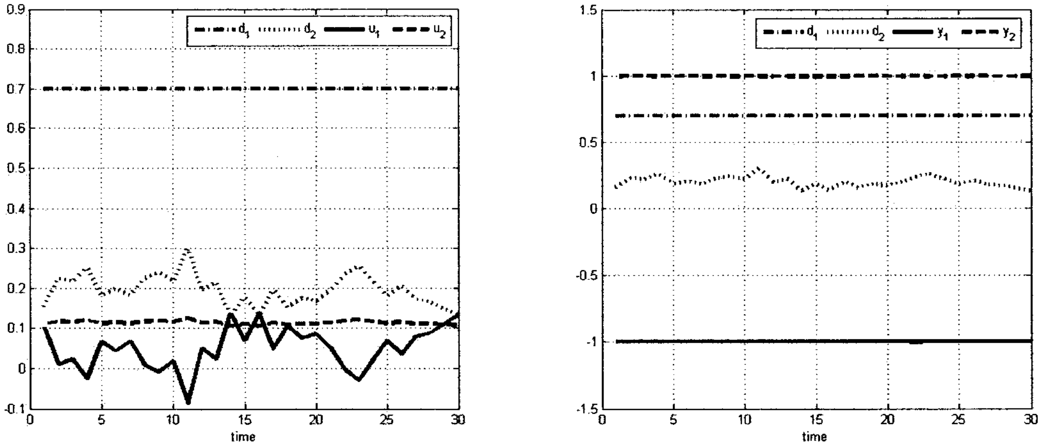
subject to:

$$y_1 = -1.2u_1 - 14.4u_2 + d_1$$
$$y_2 = 1.3u_1 + 6.5u_2 + d_2$$
$$-1.0 \leq y_1 \leq 1.0$$
$$-1.0 \leq y_2 \leq 1.0$$
$$-0.5 \leq u_1 \leq 0.5$$
$$-0.5 \leq u_2 \leq 0.5 \qquad (5.9)$$

Bias steady-state values:

$$d_1 = 0.7, \quad d_2 = 0.2$$

The solution of the LP in Case Study 5.4 is presented graphically in Figure 5.11. Its solution lies at the intersection of the upper constraints for y_1 and y_2 which are almost collinear. Assume that the bias d_1 is kept at its steady-state which means that the upper constraint for y_1 is fixed. Then decrease in bias term d_2 from its steady-state will make the upper constraint for y_2 go up (as shown in Figure 5.11(a)), and vice versa, i.e. if d_2 increases from its steady-state, the same constraint goes down (as shown in Figure 5.11(b)). Since the solution of this LP lies at the intersection of constraints y_1^{max} and y_2^{max}, and they are almost collinear, then even mild fluctuations in bias d_2 may cause the LP solution to migrate over a vast region in the manipulated variable domain. This can be shown using simulations where it was assumed that the bias terms d_1 and d_2 are at their steady-states and that d_2 is subject to white noise. The results of these simulations are presented in Figure 5.12, where Figure 5.12(a) shows the variation in the inputs and Figure 5.12(b) shows the response of the output set points. In these figures it can be seen that the noise in the bias d_2 causes variation of the optimal input target values, especially in u_1^{tar}, which appears to be much more sensitive to d_2 than u_2^{tar}. In this simulation, noise in d_2 has variation 0.0019 and the variation of the input target u_1^{tar} is 0.0032. Therefore, if input u_1 has a target value which is given by the LP, then its trajectory will be non smooth.



(a) Biases and input target values. (b) Biases and output set points.

Figure 5.12: 2 × 2 system sensitivity: Case 1

The output set points obtained in this simulation are presented in Figure 5.12(b). Although the u_1 target value has a large variation, the output set points are constant. They are constant because even though the solution of the LP changes at every iteration, it still lies at the intersection of the output constraints.

This case study concludes that even though the LP solution for input target values can appear to be sensitive to the changes in the bias terms, the output set points are not always sensitive to the same perturbations.

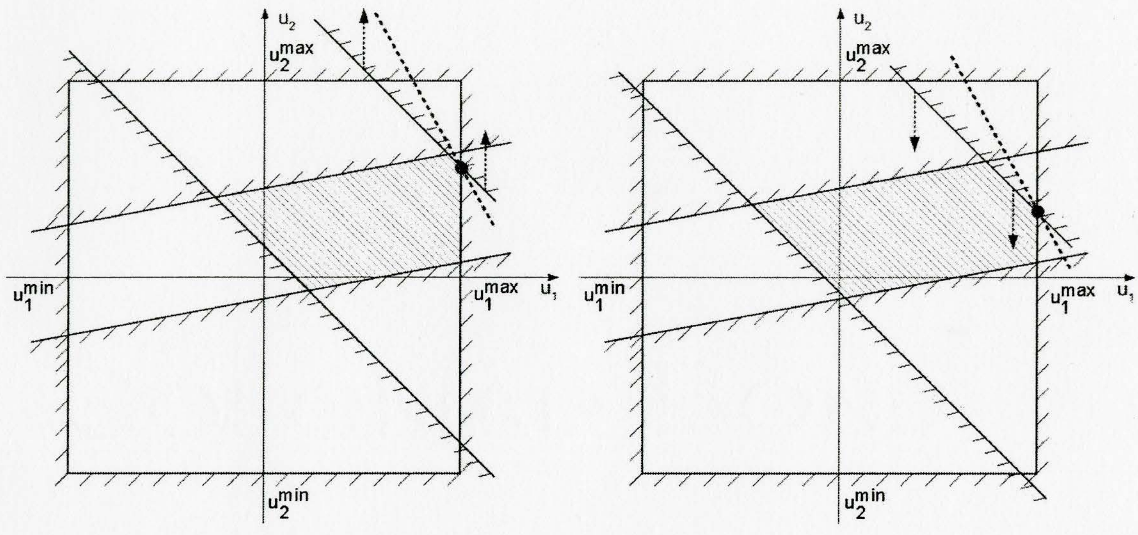
5.4.2 Sensitivity analysis of a 2×2 system. Case 2.

The case study considered above has shown that if the LP solution lies at the intersection of the output constraints only, the output set points are constant while the input target values can be sensitive to noise. Two possible situations when the solution of the LP lies at the input constraint will be considered in the next two case studies. The first simulation scenario is graphically presented in Figure 5.13. The solution of the LP lies at the intersection of constraints u_1^{max} and y_1^{max} which are close to collinearity with each other. Constraint u_1^{max} is fixed, however, decrease in bias term d_1 from its steady-state causes constraint y_1^{max} go up (see Figure 5.13(a)) and vice versa (see Figure 5.13(b)). Therefore, if the bias term d_1 is subject to white noise, constraint y_1^{max} shifts up and down, changing the LP solution at every iteration.

The following case study investigates the effect of such solution changes on the resulting input target values and output set points.

Case Study 5.5. Model:

$$
\begin{aligned}
y_1(s) &= \frac{6.2}{3s+1}u_1(s) + \frac{0.25}{3s+1}u_2(s) \\
y_2(s) &= -\frac{1.3}{3s+1}u_1(s) + \frac{2.1}{3s+1}u_2(s)
\end{aligned}
$$



(a) (b)

Figure 5.13: Effect of the bias on the LP solution: Case 2

with constraints:

$$-1.0 \leq y_1 \leq 1.0$$

$$-1.0 \leq y_2 \leq 1.0$$

$$-0.5 \leq u_1 \leq 0.5$$

$$-0.5 \leq u_2 \leq 0.5$$

Objective function:

$$\max_{y_1, y_2} \quad f = 25.6u_1 + u_2$$

The corresponding LP has the following form:

$$\max_{y_1, y_2} \quad f = 25.6u_1 + u_2$$

subject to:

$$y_1 = 6.2u_1 + 0.25u_2 + d_1$$
$$y_2 = -1.3u_1 + 2.1u_2 + d_2$$
$$-1.0 \le y_1 \le 1.0$$
$$-1.0 \le y_2 \le 1.0$$
$$-0.5 \le u_1 \le 0.5$$
$$-0.5 \le u_2 \le 0.5 \qquad (5.10)$$

Bias steady-state values:

$$d_1 = -2.1, \quad d_2 = 0.61$$

Regulatory level parameters:

Controlled variables:

$$y_1, \; y_2$$

Weights: $Q = 1.0I_p$, $R = 0.0I_m$, $S = 1.0I_m$

Simulation parameters: prediction horizon $P = 50$, control horizon $M = 2$, sampling time $T_s = 0.3$

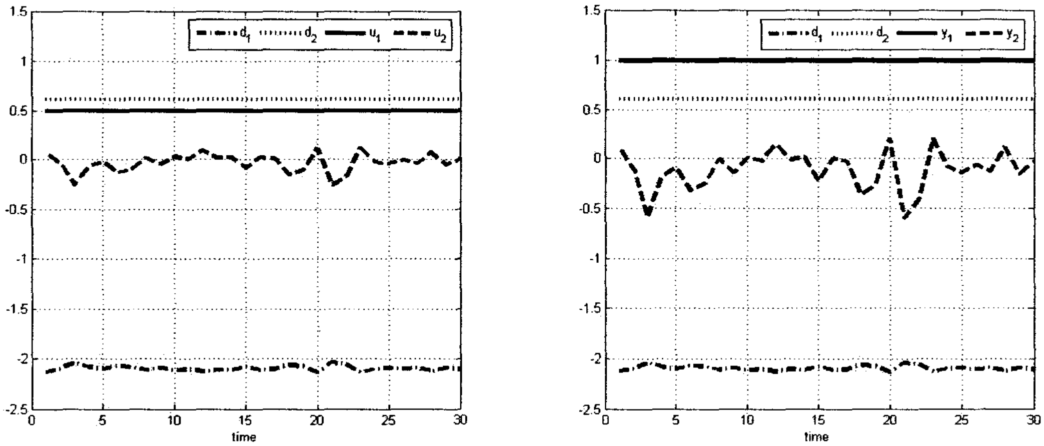
The results of the simulation are presented in Figures 5.14(a) and 5.14(b).

As expected, input u_1 is at its maximum constraint. White noise in the bias term d_2 causes some variation in the input u_2 target value and analytically this dependency can be expressed through the following equation if the y_1^{max} constraint is active:

$$originally: \quad u_2 = -\frac{g_{11}^{ss}}{g_{12}^{ss}}u_1 + \frac{y_1^{max} - d_1}{g_{12}^{ss}}$$

$$u_2 \; as \; a \; function \; of \; d_1: \quad u_2(d_1) = -\frac{1}{g_{12}^{ss}}d_1 + \left(-\frac{g_{11}^{ss}}{g_{12}^{ss}}u_1 + \frac{y_1^{max}}{g_{12}^{ss}}\right)$$

$$if \; u_1 \; is \; constant: \quad u_2(d_1) = -\frac{1}{g_{12}^{ss}}d_1 + const$$



(a) Biases and input target values.

(b) Biases and output set points.

Figure 5.14: 2×2 system sensitivity: Case 2

Therefore, the sensitivity of the solution for u_2 depends on the absolute value of steady-state gain g_{12}^{ss}. In this case study $g_{12}^{ss} = 0.25$ and therefore, fluctuations in the the target values for u_2 have about 4 times larger amplitude than the noise in d_1. Since the solution of the LP lies at the intersection of the upper bounds for y_1 and u_1 the latter are constants for the entire simulation period. However, the fluctuations in u_2 result in non constant set points for y_2. If g_{22}^{ss} is larger than unity, then any perturbations in u_2^{tar} will be amplified providing even larger oscillation in y_2^{set} and vice versa. In this case study $g_{22}^{ss} = 2.1$ and as it can be seen in Figure 5.15 the set points for y_2 have larger variation than the target values for u_2 and much larger variation than the output noise in y_2.

5.4.3 Sensitivity analysis of a 2×2 system. Case 3.

Analogously to the previous case, the solution of the LP can become sensitive to the changes in the bias term if the constraint line for u_2 in pair 1 or pair 2 in formulation (5.8) is almost collinear to the upper or lower bound for u_2. Then, if the solution of the LP lies at the intersection of these constraints, small changes in the bias term may result in a large vari-

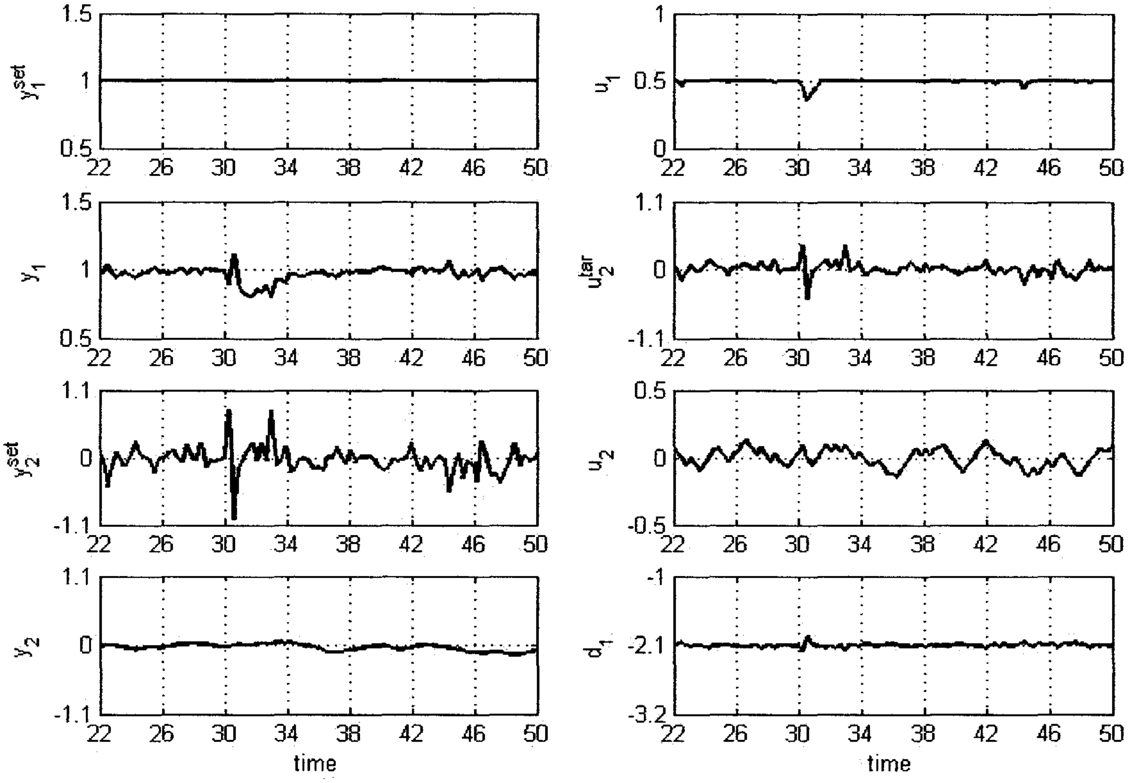


Figure 5.15: Two-level control system response: Case Study 5.5

ation in the LP solution. This scenario is graphically presented in Figure 5.16. The upper constraint in pair 2 in formulation (5.8) is almost collinear to the upper bound on u_2, and the objective function is such that the solution of the LP lies at the intersection of these constraints. A small decrease in bias term d_2 will cause the constraints in pair 2 shift up (as shown in Figure 5.16(a)) and vice versa (as shown in Figure 5.16(b)). This will result in the LP solution fluctuating along the upper bound on u_2, and since the constraints are almost collinear, these fluctuations can have much larger variation than the variation of bias d_2.

The scenario described was simulated using the following case study.

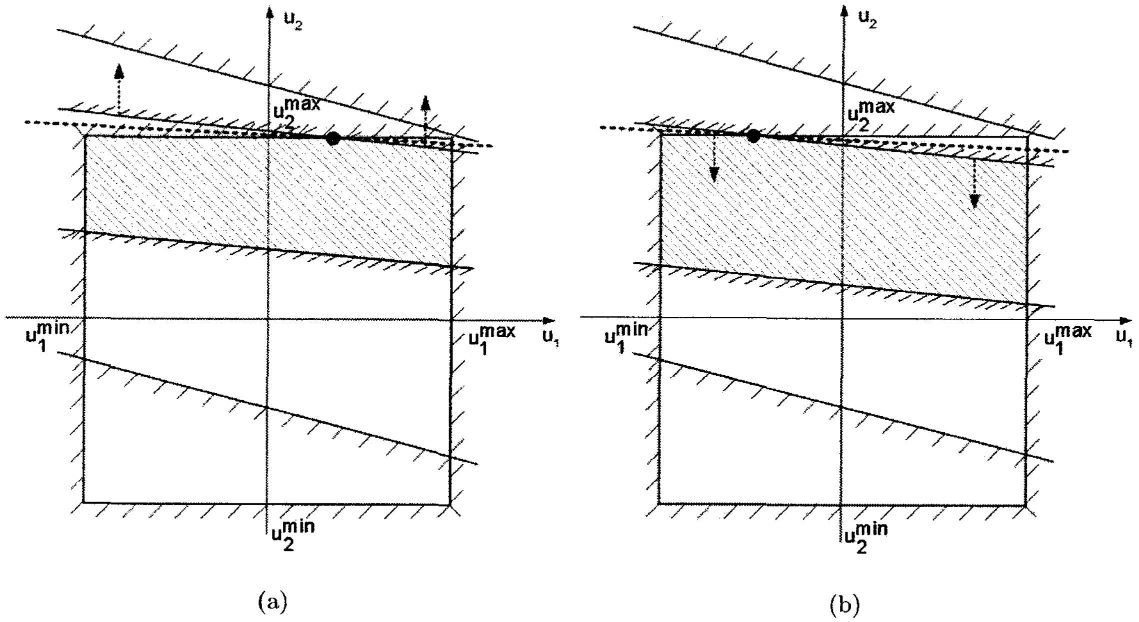


Figure 5.16: Effect of the bias on the LP solution: Case 3

Case Study 5.6. Model:

$$\begin{aligned} y_1(s) &= -\frac{1.2}{3s+1}u_1(s) - \frac{1.4}{3s+1}u_2(s) \\ y_2(s) &= \frac{0.5}{3s+1}u_1(s) + \frac{5.5}{3s+1}u_2(s) \end{aligned} \tag{5.11}$$

with constraints:

$$-1.0 \le y_1 \le 1.0$$

$$-1.0 \le y_2 \le 1.0$$

$$-0.5 \le u_1 \le 0.5$$

$$-0.5 \le u_2 \le 0.5$$

Objective function:

$$\max_{y_1,y_2,u_1,u_2} \quad f = 0.3y_1 + 0.5y_2 + 0.6u_1 + 8.3u_2$$

The corresponding LP has the following form:

$$\max_{y_1,y_2,u_1,u_2} \quad f = 0.3y_1 + 0.5y_2 + 0.6u_1 + 8.3u_2$$

subject to:

$$y_1 = -1.2u_1 - 1.4u_2 + d_1$$
$$y_2 = 0.5u_1 + 5.5u_2 + d_2$$
$$-1.0 \leq y_1 \leq 1.0$$
$$-1.0 \leq y_2 \leq 1.0$$
$$-0.5 \leq u_1 \leq 0.5$$
$$-0.5 \leq u_2 \leq 0.5 \tag{5.12}$$

Bias steady-state values:

$$d_1 = 0.4, \quad d_2 = -1.8$$

Regulatory level parameters:

Controlled variables:

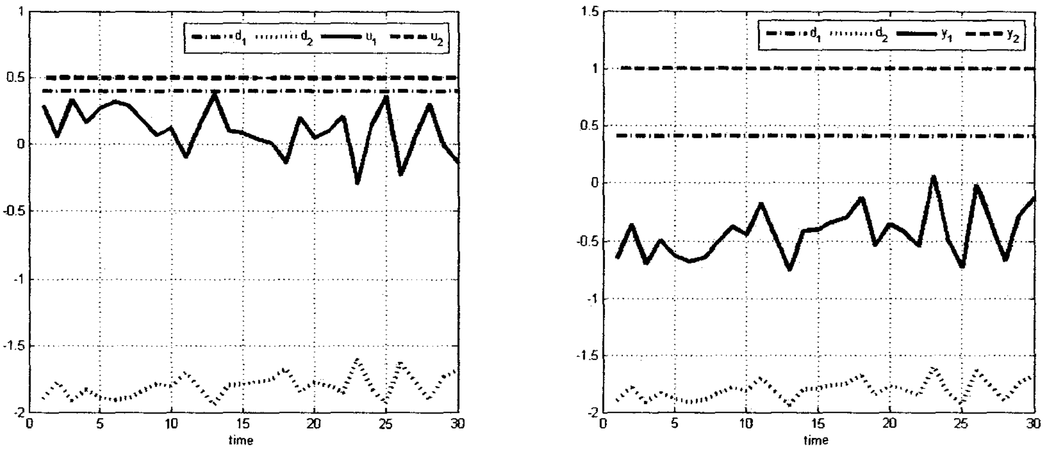
$$y_1, \; y_2$$

Weights: $Q = 1.0I_p$, $R = 0.0I_m$, $S = 1.0I_m$

Simulation parameters: prediction horizon $P = 50$, control horizon $M = 2$, sampling time $T_s = 0.3$

In the simulation, bias d_2 was subject to white noise, while bias d_1 was kept at its steady-state value. The results of the simulation for MV targets and CV set points are presented in Figure 5.16. Since the solution of the LP lies at the intersection of the constraints for y_2 and u_2, the corresponding set points and target values are constant through the entire simulation period. Figure 5.17(a) shows that noise in d_2 causes fluctuation of u_1 with larger variation. Indeed, since y_2^{set} is always equal to $y_2^{max} = 1$ and $u_2^{tar} = u_2^{max}$ then the LP solution for any value of bias d_2 must satisfy the equality:

$$y_2^{max} = g_{21}^{ss}u_1 + g_{22}^{ss}u_2^{max} + d_2 \Leftrightarrow u_1 = -\frac{1}{g_{21}^{ss}}d_2 + const$$



(a) Biases and input target values.

(b) Biases and output set points.

Figure 5.17: 2×2 system sensitivity: Case 3

$g_{21}^{ss} = 0.5$ and therefore, any changes in d_2 causes changes in u_1 that are twice as large and in the opposite direction. At the same time, since y_1 is related to u_1 through the steady state equation 5.12, and u_2 and d_1 are constants, any changes in u_1 result in changes in y_1 with coefficient g_{11}^{ss} (which is -1.2 in this case study). Summarizing both effects, it can be concluded that noise in d_2 results in changes in y_1 such that: $y_1 = -g_{11}^{ss}/g_{21}^{ss} d_2$. In this case study $-g_{11}^{ss}/g_{21}^{ss} = 2.4$, and it is confirmed by the results presented in Figure 5.17.

The effect on the two-level control system behavior has also been considered. The process and the model were presented through equations (5.11), and the steady state bias values were achieved by introducing the output step-like disturbances passing through first-order filters with unity steady-state gains. Also, white noise was applied to output y_2, while output y_1 was maintained uncorrupted. The results of the two-level control system operation after the transient effects have died out are presented in Figure 5.18. As expected, white noise with variance 0.0013 caused fluctuations in u_1^{set} with variance 0.0051 which resulted in oscillations in y_1^{set} with variance 0.0074. This confirms that in this case study, white noise in the second output may cause the fluctuation of the output set points with much larger variation. Even though the set points for first output are severely corrupted by the

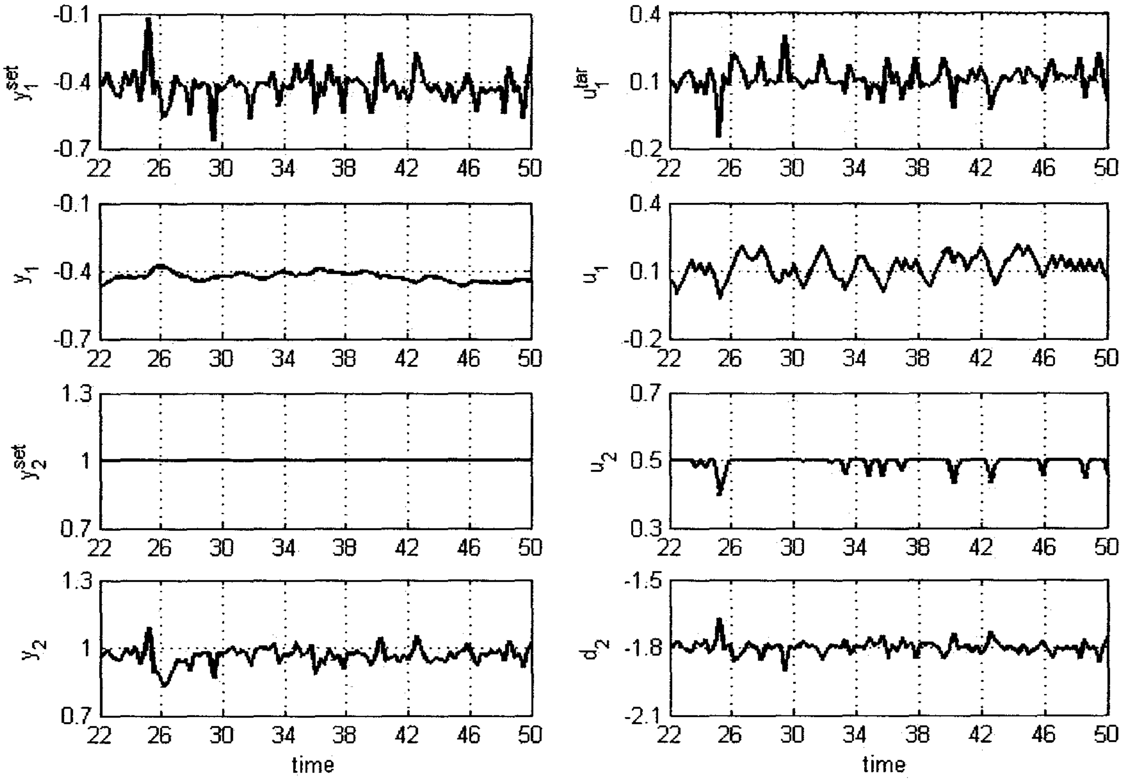


Figure 5.18: Two-level control system response: Case Study 5.6

noise, the control performance for the first output is quite good. The variance of the output (0.000493) is much smaller than the variance of the set points. However the performance of output y_2 is quite poor (its variance is 0.0018) considering that its set point is constant at steady state. This can be explained by fact that fast changes in y_1^{set} caused frequent changes in u_1 and at the same time, u_2 was operating near its upper constraint and at some points in time it saturated and could not be used for control.

5.5 Chapter Summary

The obtained results have shown that in some cases the LP solution can be sensitive to the bias values provided from the regulatory level. In the general case for a SISO and a

MISO system, the noise in the bias term causes variation in the input target values which is inversely proportional to the corresponding steady-state gain. If the output is not at a constraint, its deviation replicates the deviation of the bias without any modification.

The sensitivity of the 2×2 LP to the non-constant perturbations depends on the placement of the solution. If the solution lies at the intersection of the output constraints, then input target values can be sensitive to noise, although the output set points would stay constant. In case when the solution remains at the input constraints, the output set points can be sensitive to such variations if the plant steady- state gains are large.

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

This thesis is devoted to the analysis of the control performance of LP-MPC cascade control systems under a variety of conditions.

Initially, two methods of LP bias update, a method which uses set points and a method with model predictions, were evaluated and compared. A variety of scenarios with different plant/model mismatch, constraints and LP objective function were considered. It was observed that use of the steady-state process model directly in the bias update scheme resulted in stability of the two-level system and good dynamic performance in most scenarios considered, even with quite significant plant/model mismatch. In the case, where the bias was calculated using LP set points, cases of instability of the cascade system with steady-state optimization were observed.

Different LP execution frequencies were studied, and more frequent LP execution appeared to have a stabilizing effect on the overall system, in addition to more rapid convergence to the final steady-state.

A MIMO case study has shown that the chosen control structure may have an effect on the stability and control performance of an LP-MPC cascade control system. Chattering of the set points was observed when an auxiliary output was constrained but not controlled. The set points stabilized with an increased LP execution frequency, but poor dynamic performance was observed. Changing the control structure to include the auxiliary output as a controlled variable resulted in superior performance.

Sensitivity of the LP to noise-like perturbations was also considered. It was shown that the output noise can affect the LP solution through the bias term, and depending on the steady-state model gains, the LP set points may have larger variation than the noise itself. Large variation in target values for the manipulated variables from the LP can result in erratic fluctuations in the inputs, thereby degrading the performance of controlled outputs. With a 2×2 case study it was shown that in a MIMO system, constraints and bias steady-state values can have a significant effect on LP sensitivity. Three scenarios for a 2×2 system, where the LP solution appeared to be sensitive to the output noise were presented and discussed. Inclusion of such LP designs into a two-level cascade control system resulted in overall performance degradation.

In addition to steady-state optimization studies, the problem of plant steady-state calculation was considered. A computational framework for determining the equilibrium point of a closed-loop system under constrained MPC was provided. It was shown that use a steady-state model only is in general case insufficient for correct calculation, and that such a calculation in general cannot be generated as the solution of a standard quadratic programming problem. The problem formulation developed accounts for the effects of the controller dynamics as well as plant/model mismatch. Two alternative solution strategies were proposed; a sequential and a simultaneous solution strategy. The sequential strategy has discontinuous derivatives induced by an inner quadratic programming problem and application of standard gradient-based nonlinear equation solution techniques is consequently expected to pose difficulties. The simultaneous strategy involves replacement of the quadratic programming problem by its equivalent Karush-Kuhn-Tacker conditions.

The resulting problem includes complementarity constraints which can pose computational difficulties if not carefully dealt with. An interior point method that is able to handle such constraints in the primary problem was shown to reliably converge to the solution.

An application of the method to an LP-MPC cascade control system where the LP is executed only at steady-state was presented. The proposed method was used to compute the closed-loop steady-state of the MPC-controlled process for given set points. It was also shown how the equation system may be included within overall optimization setting which gives possibilities for its further use in applications such as design of real-time optimization systems.

6.2 Recommendations for Further Work

It was shown in Section 3.2 that control structure may have a significant effect on the performance and even stability of LP-MPC cascade systems. Therefore, an interesting issue for further study is a thorough analysis of the effect of control structure on overall LP-MPC system performance. Some recommendations on this topic have been summarized in Section 3.3.

In this thesis, studies on LP sensitivity were made for 1×2 and 2×2 systems. In all simulations presented, the solution of the LP migrated over a wide area; however, the noise considered did not cause any changes in the active constraint set. It is possible, that in an LP with larger dimension, high frequency disturbances may cause switching of the LP solution between different constraint intersections and even different active constraint sets. In this case, the set of the variables used for control and the set of the variables which operate at constraints will be changing in a highly erratic manner. This could lead to potential stability problems of LP-MPC cascade system and is worth studying in future.

List of References

ÅSTRÖM, K. J. AND WITTENMARK, B. (1990). *Computer-Controlled Systems: Theory and Design.* Prentice Hall.

BAKER, R. AND SWARTZ, C. L. E. (2007). Interior Point Solution of Multilevel QP Problems in Embedded Constrained MPC Formulations. Accepted for Publication in Industrial and Engineering Chemistry Research.

BROSILOW, C. AND ZHAO, G. Q. (1988). A Linear Programming Approach to Constrained Multivariable Process Control. *Control and Dynamic Systems*, **27**, 141–181.

CUTLER, C. R. AND RAMAKER, B. L. (1979). Dynamic Matrix Control - A Computer Algorithm. In *AIChE 86th National Meeting, Houston, TX.*

FORBES, F. AND MARLIN, T. E. (1994a). Model Accuracy for Economic Optimizing Controllers: the Bias Update Case. *Industrial and Engineering Chemistry Research*, **33**, 1919–1929.

FORBES, F. AND MARLIN, T. E. (1994b). Model Adequacy Requirements for Optimizing Plant Operations. *Computers and Chemical Engineering*, **18**, 497–510.

FOURER, R., GAY, D. M., AND KERNIGHAN, B. W. (1993). *AMPL - A Modelling Language for Mathematical Programming.* The Scientific Press.

FRIEDMAN, Y. (2000). Closed-Loop Optimization Update - a Step Closer to Fulfilling the Dream. *Hydrocarbon Processing*, **79**(1), 15–16.

GARCIA, C. AND MORSHEDI, A. (1986). Quadratic Dynamic Solution of Dynamic Matrix Control (QDMC). *Chemical Engineering Communications*, **46**, 73–87.

HALL, J. AND VERNE, T. (1993). RCU Optimization in a DCS Gives Fast Payback. *Hydrocarbon Processing*, **10**, 85–92.

JAKHETE, R., RAGER, W., AND HOFFMAN, D. (1999). Online Implementation of Composite LP Optimizers FCCU/GPU Complex. *Hydrocarbon Processing*, **78**(2), 69–76.

KASSIDAS, A., PATRY, J., AND MARLIN, T. (2000). Integrating Process and Controller Models for the Design of Self Optimizing Control. *Computers and Chemical Engineering*, **24**, 2589–2602.

KASSMAN, D. E., BADGWELL, T. A., AND HAWKINS, R. B. (2000). Robust Steady-State Target Calculation for Model Predictive Control. *AIChE Journal*, **46**(5), 1007–1024.

KOOKOS, I. K. AND PERKINS, J. D. (2001). An Algorithm for Simultaneous Process Design and Control. *Industrial and Engineering Chemistry Research*, **40**, 4079–4088.

KOOKOS, I. K. AND PERKINS, J. D. (2002). Regulatory Control Structure Selection of Linear Systems. *Computers and Chemical Engineering*, **26**, 875–887.

KOZUB, D. J. (2002). Controller Performance Monitoring and Diagnosis. Industrial Perspectives. In *Camacho, E. F., Basanez, L., and de la Puente, J. A. (Eds.)*, 15th Triennial World Congress of the International Federation of Automatic Control (IFAC). 21-26 July 2002, Barcelona, Spain.

KWAKERNAAK, H. AND SIVAN, R. (1972). *Linear Optimal Control Systems.* John Wiley and Sons, Inc.

LI, S., LIM, K., AND FISHER, D. (1989). A State-Space Formulation for Model Predictive Control. *AIChE Journal*, **35**, 241–249.

MACIEJOWSKI, J. M. (2002). *Predictive Control with Constraints.* Prentice Hall.

MARLIN, T. AND MUDT, D. (2004). Selecting Among Approaches for Model-Based Operations Optimization. Presented at AspenWorld, Orlando, FL, Oct. 11-14.

MARLIN, T. E. AND HRYMAK, A. N. (1997). Real-Time Optimization of Continuous Processes. *In Fifth International Conference on Chemical Process Control. AIChE Symposium Series*, **93**, 156–164.

MARLIN, T. E. AND YOUNG, M. (1998). Integrating the Effects of Process Controllers with Steady-State, Equation-Based Simulation. *Chemical Engineering Communications*, **165**, 67–87.

MARQUIS, P. AND BROUSTAIL, J. (1998). SMOC, A Bridge Between State-Space and Model Predictive Controllers: Application to the Automation of a Hydrotreating Unit. In *Proceedings of the 1988 IFAC Workshop on Model Based Process Control*, pp. 37–45.

MOHIDEEN, M. J., PERKINS, J. D., AND PISTIKOPOULOS, E. N. (1996). Optimal Design of Dynamic Systems under Uncertainty. *AIChE Journal*, **42**(8), 2251–2272.

MORO, L. F. L. AND ODLOAK, D. (1995). Constrained Multivariable Control of Fluid Catalytic Cracking Converters. *Journal of Process Control*, **5**, 29–39.

MUSKE, K. R. (1997). Steady-State Target Optimization in Linear Model Predictive Control. In *Proceedings of the American Control Conference, Albuquerque, New Mexico*, pp. 3597–3601.

NATH, R., ALZEIN, Z., AND POUWER, R. (1999). On-Line Dynamic Optimization of an Ethylene Plant Using Profit Optimizer. For Presentation at the 1999 NPRA Computer Conference, November 14-17, 1999.

PEDERSON, C., MUDT, D., BAILEY, J., AND AYALA, J. (1995). Closed-Loop Real-Time Optimization of Hydrocracker Complex. In *NPRA Computer Conference, Nashville, TN, Nov 1995.*

PRETT, D. AND MORARI, M. (1986). *The Shell Process Control Workshop.* Butterworths, Stoneham, MA.

QIN, S. AND BADGWELL, T. (2003). A Survey of Industrial Model Predictive Control Technology. *Control Engineering Practice*, **11**, 733–764.

RAGHUNATHAN, A. AND BIEGLER, L. T. (2003). Mathematical Programs with Equilibrium Constraints (MPECs) in Process Engineering. *Computers and Chemical Engineering*, **27**, 1381–1392.

RAMOS, C., SENENT, J., BLASCO, X., AND SANCHIS, J. (2002). LP-DMC Control of a Chemical Plant with Integral Behaviour. In *15th IFAC World Congress 2002, Barcelona, Spain, July 2002*.

RICHALET, J., RAULT, A., TESTUD, J., AND PAPON, J. (1978). Model Predictive Heuristic Control: Applications to Industrial Processes. *Automatica*, **14**, 413–428.

RICKER, N. (1990). Model Predictive Control with State Estimation. *Industrial and Engineering Chemistry Research*, **29**, 374–382.

SCHWEIGER, C. A. AND FLOUDAS, C. A. (1998). Interaction of Design and Control: Optimization with Dynamic Models. In Hager, W. and Pardalos, P. (Eds.). *Optimal Control: Theory, Algorithms, and Applications*, . pp. 388-435, Kluwer Academic Publishers.

SHAH, S., PATWARDHAN, R., AND HUANG, B. (2002). Multivariable Controller Performance Analysis: Methods, Applications and Challenges. *AIChE Symposium Series*, **98**, 190–207.

SORENSEN, R. AND CUTLER, C. (1998). LP Integrates Economics into Dynamic Matrix Control. *Hydrocarbon Processing*, **77**(9), 57–65.

SOUFIAN, M. AND SANDOZ, D. (1996). Constrained Multivariable Control and Real-Time Optimization of a Distillation Process. In *UKACC International Conference on Control, 2-5 September 1996, IEE*.

THE MATHWORKS USER'S GUIDE (2006). *Optimization Toolbox for Use with Matlab©*. URL: `http://www.mathworks.com/access/helpdesk/help/pdf_doc/optim/optim_tb.pdf`.

VERMEER, P. AND PEDERSON, C. (1996). Design and Integration Issues for Dynamic Blend Optimization. In *NPRA Computer Conference, 11-13 November 1996*.

Verne, T., Jerrit, B., and Bester, P. (1999). Unique Approach to Real-Time Optimization. *Hydrocarbon Processing*, **78**(3), 53–60.

Winston, W. L. (2004). *Operations Research: Applications and Algorithms.* Duxbury Press, 4-th edition.

Ying, C. M. and Joseph, B. (1999). Performance and Stability Analysis of LP-MPC and QP-MPC Cascade Control System. *AIChE Journal*, **45**(7), 1521–1533.

Yousfi, C. and Tournier, R. (1991). Steady-State Optimization Inside Model Predictive Control. In *Proceedings of the American Control Conference*, pp. 1866–1870.

Yu, Z., Li, W., Lee, J., and Morari, M. (1994). State Estimation Based Model Predictive Control Applied to Shell Control Problem: A Case Study. *Chemical Engineering Science*, **49**, 285–301.