Czech Technical University in Prague
Faculty of Electrical Engineering

DIPLOMA THESIS

Influence of Model Uncertainty on Constraints Handling in Predictive Control

Prague, 2009 **Ondřej Šantin**

Declaration

I hereby declare that I am the author of the following text. All the citations and references are complete and properly named.

In Prague on ______________ ____________________

signature

Acknowledgement

I would like to express my gratitude to Jaroslav Pekař for his support during realization of this work and for numerous valuable consultations that I had with him. Many thanks to Vladimír Havlena and a special thanks to my family and my friends for providing me constant encouragement.

Abstrakt

Prediktivní řízení (MPC) je populární metoda, která dokáže přirozeně brát v úvahu omezení a na základě definovaného kritéra optimality zprostředkovat optimální řídící zásah. Hlavní nevýhodou tohoto přístupu je výpočetně náročná optimalizace, která se řeší v průběhu řízení pro každý akční zásah, takže není takto možné řídit rychlé systémy. Tento nedostatek částečně odstraňuje explicitní formulace MPC. Zde je však limitujím faktorem složitost výsledného zákona řízení, která roste s počtem omezení, které jsou na systém kladeny.

Další nevýhodou MPC je potřeba dostatečně přesného modelu řízeného systému, protože ten je použit pro predikci budoucího vývoje systému. Pokud je model nepřesně určen a vyskytují-li se poruchy, kvalita řízení může být nízká a pro systémy s omezeními nemusí existovat výsledek optimalizace splňující zadaná omezení. Proto mnoho praktických aplikací MPC používá tzv. řízení "set range", kde je umožněno aby se řízené veličiny pohybovaly v určitém rozsahu a jakékoliv porušení tohoto rozsahu je penalizováno nějakou kvadratickou funkcí. Bohužel, "set range" řízení vyžaduje přidání velkého množství omezení, takže rychle vzrůstá i složitost zákona řízení explicitního MPC regulátoru.

Konflikt toho, že je třeba zajistit existenci řešení optimalizace splňující zadaná omezení i přes neurčitost modelu a vliv poruch při malé složitosti zákona řízení MPC regulátoru, může být vyřešen použitím algoritmu, který je popsán v této práci.

Abstract

Model Predictive Control (MPC) is a popular method which can naturally deal with constraints and provides optimal control actions based on a declared cost function. The main drawback of this approach is need for on-line optimization, which limits the usage of MPC only for slow process. The on-line optimization can be avoided by using the Explicit formulation of MPC, where all optimization can be performed off-line, so that it is possible to use MPC even for high-speed systems. Main drawback of Explicit formulation of MPC is complexity of the result control law, which increases with the number of constraints.

The weakness of MPC is the need for a good and accurate process model because the model is used for prediction of future system response. When disturbances or model uncertainty are present, the control performance may be poor and the solution of optimization may be infeasible in constrained case. Thus many practical applications use the range control, where the controlled variables are enabled to freely vary in specified range and violation of the range is penalized by a quadratic function. The problem is that the range control needs to add many constraints so that the complexity of resulting explicit MPC control law will extremely increase.

The conflict that it is necessary to ensure the feasible solution of the constrained MPC, despite of the small complexity of the control law, presence of model uncertainty and disturbances can be solved by using the algorithm described in this work.

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Ondřej Šantin**

Studijní program: Elektrotechnika a informatika (magisterský), strukturovaný
Obor: Kybernetika a měření, blok KM1 - Řídicí technika

Název tématu: **Vliv neurčitosti modelu na splnění omezení veličin při prediktivní regulaci**

Pokyny pro vypracování:

1. Seznamte se s metodami lineární prediktivní regulace, s metodami odhadu neměřitelné poruchy (neurčitosti modelu) pomocí techniky "Unknown Input Observer", jejich modifikacemi (výběr vhodného modelu poruchy) a jejich využitím v prediktivním řízení.
2. Analyzujte vliv neurčitosti modelu na splnění "hard" a "soft" omezení. Zaměřte se na krátké horizonty predikce. Navrhněte a implementujte algoritmy zajišťující splnění omezení při zachování kvality regulace i přes neurčitost modelu. Porovnejte efektivnost a numerickou náročnost jednotlivých řešení.

Seznam odborné literatury:

[1] Rossiter J. A., Model-Based Predictive Control, CRC Press, 2003
[2] Maciejowski J., Predictive control with constraints, Prentice Hall, 2001
[3] Short course on MPC, ETH-Zurich, 2005
[4] Vybrané časopisecké publikace

Vedoucí: Prof.Ing. Vladimír Havlena, CSc.

Platnost zadání: do konce zimního semestru 2009/10

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE Fakulta elektrotechnická L.S.

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 27. 2. 2009

Contents

List of Figures

Chapter 1

Introduction

Model predictive control (MPC) [56] is a control strategy that has been widely accepted in the industrial process control community and implemented successfully in many commercial applications [8], [1], [30]. The greatest strength of MPC is the intuitive way in which constraints can be incorporated in a multivariable control problem formulation. However, the traditional MPC strategy demands a great amount of on-line computation, limiting the use of these kinds of controllers to processes with relatively slow dynamics, since an optimization problem (often a constrained quadratic program (QP)) is solved at each sampling time step.

Note that many progress was made in this field, namely the *on-line active set strategy* for linear [19] and for nonlinear MPC [20] or interior-point method [63]. Another approach of solving QP for multivariable system, called *multiplexed MPC*, was introduced in [42]. Here, the MPC problem is solved for each subsystem sequentially, and the subsystem controls are updated as soon as the solution is available.

It has recently been shown that much of the computational effort in traditional MPC can be done off-line. In [7] and [6] the authors present algorithms for solving multiparametric quadratic programs (mpQPs) that are used to obtain explicit solutions to the MPC problem. Thus, the Explicit model predictive controller accomplishes on-line MPC functionality without solving an optimization problem at each time step. This makes the Explicit MPC usable for high-speed application, like engine control [51] or attitude control for spacecraft [32].
The main disadvantage of the Explicit MPC is its large growth of complexity of a mp-QP solution with increasing number of constraints. Although the complexity can be reduced

as show [27], [62], [33], the Explicit MPC is still limited to models with small state dimensions and few process inputs.

Stability and robustness are very important properties of each control system. For MPC technology, these two properties are the subject of intensive research [53],[5],[46].

The weakness of MPC is the need for a good and accurate process model because the model is used for prediction of future system response. Thus the conjunction of model uncertainties and disturbance with constraints of process variables may caused the infeasible solution of QP. This drawback is reduced by the range control approach [43], where the soft constraints are used. The soft constraints are usually used to system output to specify the range in which the controlled variables are enabled to freely vary. The control action is without changes as long as the predictions of the controlled variables are inside of the range. Violation of the specified range is penalized, usually by a quadratic cost and then the appropriate control sequence that ensures return back to the range is generated. This control sequence may be very aggressive in case of small control or prediction horizon and may caused unsuitable behavior of the process response.

The main aim of this work is to introduce the algorithm which will provide the offset-free tracking and the output soft constraints satisfaction in spite of model uncertainties, disturbances and small control and prediction horizon of MPC in control problem where some of the controlled variables are tracked to a given references and the rest of them are controlled in specified range. The main idea of the algorithm is that the output soft constraints are considered only in one step of the outputs prediction. Therefore when the constraint is violated the controller ignores it up to step where the constraint is considered, thus no control action is generated immediately after the violation. This leads to more careful control and the choice of the step, where the constraint is considered, is a parameter of how much the control will be careful.
This approach also avoids the dramatic increase of the control law complexity which arises in standard case, where the soft constraints are considered over the whole prediction horizon.
For elimination the steady-state offset in model uncertainty and disturbances presence the Unknown Input Observer method [50],[47],[11] was used.

The work is organized as follows: in Chapter 2 the required and motivating theoretical background of model predictive control, with focus on linear MPC, and some modifications are summarized. Afterwards, Chapter 3 reviews the parametric quadratic programming and Explicit MPC with complexity analysis. The Chapter 4 gives some insight to multivariable control and our algorithm is introduced. Practical usage of our algorithm to model of port injection spark-ignited gasoline engine is described in Chapter 5. Finally, Chapter 6 is devoted to a conclusion and some ideas for future work.

1.1 Notation

In this work the next notation will be assumed:

$\boldsymbol{A}$, $\boldsymbol{T}$	the matrices
$\tilde{\boldsymbol{A}}$	the augmented (modified) matrix $\boldsymbol{A}$
$\boldsymbol{x}$	the column vector
$\tilde{\boldsymbol{x}}$	the augmented (modified) column vector
$\boldsymbol{u}_{k,N}$	the sequence $\boldsymbol{u}_k, \ldots, \boldsymbol{u}_N$
$\boldsymbol{I}_n$	the identity matrix $\boldsymbol{I}_n \in \mathcal{R}^{n \times n}$
a, β	the constants
$\boldsymbol{A} \succ 0$	the positive-definite matrix
$\boldsymbol{A} \succeq 0$	the positive-semidefinite matrix
$\hat{(.)}$	the hat superscript denotes the estimation
$\boldsymbol{u}^*_{k,N}$	the star superscript denotes the optimal value

Chapter 2

Model Predictive Control

Model Predictive Control (MPC) is an advanced method of process control that has been in use in the process industries such as chemical plants and destilation columns since the 1980s [48].
In MPC, model of the plant is used to predict the future evolution of the system state and outputs to find proper control action. This control action is determined by optimizing the certain cost function over the sequence of future control moves subject to operating constraints. The goal of the optimization is to minimize the cost function (performance index) which penalize unwanted behavior, such as high control action of controller, tracking error and others.
One of the main reasons why MPC is so popular, is ability to do on-line constraints handling in a systematic way. But it makes this technique computationally demanding and for finding the optimum of cost function, numerical algorithm needs to be used [63], [20], [26], [4].
Next major selling point of MPC is easy tunning even for Multiple-Input Multiple-Output (MIMO) systems, where classical way of tunning Proportional–Integral–Derivative controller (PID) is too hard [34], [24].

The chapter is organized as follows: in the first section the prediction based on state-space model and model uncertainty will introduced. In the second section the basic algorithm for unconstrained and for constrained MPC will be formulated. Finally, in the last section some modification of MPC such as Δu-formulation or constraints handling will be introduced. In conclusion of the section the methods for offset-free tracking will be presented.

2.1 Prediction in Model Predictive Control

Most control laws, for example PID (proportional, integral and derivative) take into account only expected closed-loop dynamics and they do not consider the future implication of current control actions. But MPC explicitly computes the predicted behavior over some horizon.

In order to predict the future behavior of a process, we must have a model of how the process behaves. In particular, this model must show the dependence of output on the current measured variable and the current and/or future inputs.

In practice, most MPC algorithms use linear models because the dependence of the predictions on future control choices is then linear and this facilities optimization as well as off-line analysis of expected closed-loop behavior. However, nonlinear models can be used where the implied computational burden is not a problem and linear approximation is not accurate enough. A few types of models can be used for prediction, e.g. CARIMA, FIR, state-space model and other. This work will focus on linear time invariant state-space models, for more information about others, see [56].

2.1.1 State-space Model

MPC is usually implemented in discrete time, so this work always consider the linear time-invariant (LTI), discrete system

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{2.1}$$
$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k, \tag{2.2}$$

where $\boldsymbol{x}_k \in \mathcal{R}^n$ is the state vector, $\boldsymbol{u}_k \in \mathcal{R}^m$ is the control (input) vector, $\boldsymbol{y}_k \in \mathcal{R}^p$ is the output vector, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are real $n \times n$, $n \times m$, $p \times n$, and $p \times m$ matrices.

2.1.2 Prediction of the System Future Behavior

MPC algorithms make use of predictions of the system behavior. In this section it will be shown how to compute those predictions for state-space model [14],[40].

Consider the state-space model (2.1), which gives the one step ahead predictions (modified [56]):

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{2.3}$$
$$\boldsymbol{y}_{k+1} = \boldsymbol{C}\boldsymbol{x}_{k+1} + \boldsymbol{D}\boldsymbol{u}_{k+1}, \tag{2.4}$$

One can use this relationship recursively to find predictions, for instance:
Write (2.3) at $k+2$

$$\begin{aligned} \boldsymbol{x}_{k+2} &= \boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{u}_{k+1} && (2.5)\\ \boldsymbol{y}_{k+2} &= \boldsymbol{C}\boldsymbol{x}_{k+2} + \boldsymbol{D}\boldsymbol{u}_{k+2}, && (2.6)\end{aligned}$$

and substitute (2.3) into (2.5) to eliminate $\boldsymbol{x}_{k+1}$

$$\begin{aligned} \boldsymbol{x}_{k+2} &= \boldsymbol{A}^2\boldsymbol{x}_{k} + \boldsymbol{A}\boldsymbol{B}\boldsymbol{u}_{k} + \boldsymbol{B}\boldsymbol{u}_{k+1} && (2.7)\\ \boldsymbol{y}_{k+2} &= \boldsymbol{C}\boldsymbol{A}^2\boldsymbol{x}_{k} + \boldsymbol{C}\boldsymbol{A}\boldsymbol{B}\boldsymbol{u}_{k} + \boldsymbol{C}\boldsymbol{B}\boldsymbol{u}_{k+1} + \boldsymbol{D}\boldsymbol{u}_{k+2} && (2.8)\end{aligned}$$

More generally one can continue this recursion to give the T-step ahead predictions as:

$$\begin{aligned} \boldsymbol{x}_{k+T} &= \boldsymbol{A}^T\boldsymbol{x}_{k} + \boldsymbol{A}^{T-1}\boldsymbol{B}\boldsymbol{u}_{k} + \boldsymbol{A}^{T-2}\boldsymbol{B}\boldsymbol{u}_{k+1} + \cdots + \boldsymbol{B}\boldsymbol{u}_{k+T-1} && (2.9)\\ \boldsymbol{y}_{k+T} &= \boldsymbol{C}\left[\boldsymbol{A}^T\boldsymbol{x}_{k} + \boldsymbol{A}^{T-1}\boldsymbol{B}\boldsymbol{u}_{k} + \boldsymbol{A}^{T-2}\boldsymbol{B}\boldsymbol{u}_{k+1} + \cdots + \boldsymbol{B}\boldsymbol{u}_{k+T-1}\right] + \boldsymbol{D}\boldsymbol{u}_{k+T} && (2.10)\end{aligned}$$

Hence one can form the whole vector of future predictions up to a horizon T_p as follows:

$$\underbrace{\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{x}_{k+2} \\ \vdots \\ \boldsymbol{x}_{k+T_p} \end{bmatrix}}_{\boldsymbol{x}_{k+1,T_p}} = \underbrace{\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^{T_p} \end{bmatrix}}_{\boldsymbol{V_x}} \boldsymbol{x}_k + \underbrace{\begin{bmatrix} \boldsymbol{B} & & & \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & & \\ \vdots & & \ddots & \\ \boldsymbol{A}^{T_p-1}\boldsymbol{B} & & \boldsymbol{B} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{S_x}} \underbrace{\begin{bmatrix} \boldsymbol{u}_{k} \\ \boldsymbol{u}_{k+1} \\ \vdots \\ \boldsymbol{u}_{k+T_p-1} \end{bmatrix}}_{\boldsymbol{u}_{k,T_p-1}} \qquad (2.11)$$

and for outputs:

$$\underbrace{\begin{bmatrix} \boldsymbol{y}_{k+1} \\ \boldsymbol{y}_{k+2} \\ \vdots \\ \boldsymbol{y}_{k+T_p} \end{bmatrix}}_{\boldsymbol{y}_{k+1,T_p}} = \underbrace{\begin{bmatrix} \boldsymbol{C}\boldsymbol{A} \\ \boldsymbol{C}\boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{T_p} \end{bmatrix}}_{\boldsymbol{V_y}} \boldsymbol{x}_k + \underbrace{\begin{bmatrix} \boldsymbol{C}\boldsymbol{B} & \boldsymbol{D} & & \\ \boldsymbol{C}\boldsymbol{A}\boldsymbol{B} & \boldsymbol{C}\boldsymbol{B} & & \\ \vdots & & \ddots & \\ \boldsymbol{C}\boldsymbol{A}^{T_p-1}\boldsymbol{B} & & \boldsymbol{C}\boldsymbol{B} & \boldsymbol{D} \end{bmatrix}}_{\boldsymbol{S_y}} \boldsymbol{u}_{k,T_p-1} \qquad (2.12)$$

Thus the predictions of future states and outputs over prediction horizon are affine in the current state and the future control moves:

$$\begin{aligned} \boldsymbol{x}_{k+1,T_p} &= \boldsymbol{V_x}\,\boldsymbol{x}_k + \boldsymbol{S_x}\,\boldsymbol{u}_{k,T_p-1} && (2.13)\\ \boldsymbol{y}_{k+1,T_p} &= \boldsymbol{V_y}\,\boldsymbol{x}_k + \boldsymbol{S_y}\,\boldsymbol{u}_{k,T_p-1} && (2.14)\end{aligned}$$

where $\boldsymbol{V_y}\,\boldsymbol{x}_k$ is reaction of the system output to state $\boldsymbol{x}_k$, and $\boldsymbol{S_y}\,\boldsymbol{u}_{k,T_p-1}$ is reaction of the system output to control moves $\boldsymbol{u}_{k,T_p-1}$.

2.1.3 Model Uncertainty

Each system model has some uncertainty caused by noise affecting the model identification, by an inaccurate model structure, by noise presence, by surroundings or by some another influence. Therefore a fundamental question about MPC is its robustness. Any statement about robustness must make reference to a specific uncertainty range and to a specific performance criteria. While a rich theory has been developed for the robust control of linear systems, little is known about the robust control of linear systems with constraints. For handle the robustness of constrained MPC, many approaches were introduced: *Robust Invariant Sets*, *Min-Max Formulation of MPC*, etc. More information about these, reader can find in [56].
An important and probably fundamental part of robust controller design is the way of uncertainty description and modeling. There are different approaches to describe model uncertainties depending mainly on the type of technique used for designing the controllers. In the MPC context, the most important approaches are the following two:

- The true system Σ_0 belongs to a set $\mathcal{S}$, that is $\Sigma_0 \in \mathcal{S}$, where the set $\mathcal{S}$ is a given family of LTI systems.
- An unmeasured noise $\boldsymbol{w}_k$ (or measured/unmeasured disturbances) enters the system, where $\boldsymbol{w} \in \mathcal{W}$ and $\mathcal{W}$ is a given set.

The *parameter uncertainty* belongs into the first family of uncertainties description. The real model then vary within the convex hull defined by set of possible models. That is, for the *nominal model* $\boldsymbol{x}_{k+1} = \boldsymbol{A_0}\boldsymbol{x}_k + \boldsymbol{B_0}\boldsymbol{u}_k$ may for example exist set of models:

$$\boldsymbol{x}_{k+1} = \left(\boldsymbol{A_0} + \boldsymbol{\Delta A}\right)\boldsymbol{x}_k + \boldsymbol{B_0}\boldsymbol{u}_k, \tag{2.15}$$

where $\Delta\boldsymbol{A}$ represents parametric uncertainty of model dynamic and $\boldsymbol{A_0}$, $\boldsymbol{B_0}$ are the system matrices of nominal model.

The second types of uncertainties description can be for example written as

$$\boldsymbol{x}_{k+1} = \boldsymbol{A_0}\boldsymbol{x}_k + \boldsymbol{B_0}\boldsymbol{u}_k + \boldsymbol{w}_k, \qquad \boldsymbol{w} \in \mathcal{W}, \tag{2.16}$$

where $\boldsymbol{w}$ represent noise or measured/unmeasured bounded disturbances.
One can figure out that it is possible to transform parametric uncertainty (2.15) into the disturbance form of (2.16):

$$\boldsymbol{x}_{k+1} = \left(\boldsymbol{A_0} + \boldsymbol{\Delta A}\right)\boldsymbol{x}_k + \boldsymbol{B_0}\boldsymbol{u}_k = \boldsymbol{A_0}\boldsymbol{x}_k + \boldsymbol{B_0}\boldsymbol{u}_k + \underbrace{\Delta\boldsymbol{A}\boldsymbol{x}_k}_{\boldsymbol{w}_k}. \tag{2.17}$$

So it is legitimate to deal with the disturbance model uncertainty as many papers were written about it, for example [50], [47], [11]. Special case of modeling uncertainty in presence of constant disturbance is discussed later in section (2.3.6).

2.2 Basic Algorithm of Model Predictive Control

Model Predictive Control (MPC) uses an explicit process model to predict the future plant response over the chosen period, also known as the *prediction horizon* T_p. At each time step an optimization problem is solved over the sequence of future control moves, possibly subject to constraints. The future control moves are optimized over the *control horizon* T_c. Situation can be seen at figure (2.1), where n is natural number and T_S is sampling time.

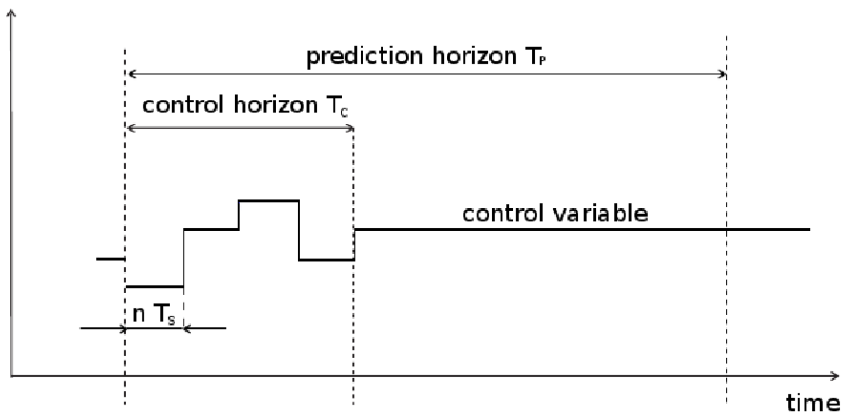

Figure 2.1: Control and prediction horizon

We also define, the *constraints horizon* T_{cs}. It is the horizon over which the constraints have to be satisfied. It is usually the same or smaller than the prediction horizon T_p.

The first of optimal control moves is the *control action* applied to the process. At the next time step, the prediction and control horizon of the optimization problem are shifted forward in time and procedure is repeated. This concept is called *receding horizon* and is described in next section.
It should be denoted, that the prediction horizon should include all significant dynamics,

otherwise performance of the control loop may be poor and important events may be unobservable.

Next, constant reference tracking problem for system (2.1) would be considered. The controlled variables are defined as:

$$\boldsymbol{z}_k = \boldsymbol{Z}\,\boldsymbol{y}_k, \tag{2.18}$$

with $\boldsymbol{z}_k \in \mathcal{R}^{p_z}$, $\boldsymbol{Z}$ full row rank and $p \geq p_z$.
Equation (2.18) means, that not all outputs track their references, but some of them are controlled only in defined range.[1].

The goal is to asymptotically eliminate the output reference tracking error for a given constant reference signal $\boldsymbol{r}_\infty$, that is

$$\boldsymbol{z}_k \longrightarrow \boldsymbol{r}_\infty, \qquad k \longrightarrow \infty. \tag{2.19}$$

Unless specified otherwise, conditions $\boldsymbol{z}_k = \boldsymbol{y}_k$, that is $\boldsymbol{Z} = \boldsymbol{I}_p$ are assumed.

2.2.0.1 The Receding Horizon Concept

Applying a computed control sequence to system, after one step of sequence, the system would be controlled in open loop. This approach is not robust and reaction to disturbance is not possible. For robust control it is necessary to measure actual output of the system in each step and recompute a control sequence. This scheme is called the *receding horizon concept*, that means horizon which is constantly moving away.
At sample k, the control law optimizes predicted performance over the time span [56]:

$$k + 1 \leq t \leq k + T_p. \tag{2.20}$$

At sample $k+1$, the control law optimizes predicted performance over the time span:

$$k + 2 \leq t \leq k + T_p + 1. \tag{2.21}$$

Thus, the time span over which the optimization takes place is always moving. At the current sample one takes into account points that were previously beyond the time span. Implementation is simple. There is optimal control sequence $\boldsymbol{u}^*_{k,T_c-1}$ found in k-th step, but only first control move is applied to system.

$$\boldsymbol{u}_k = [\boldsymbol{I}_m, \boldsymbol{0}, \cdots, \boldsymbol{0}]\, \boldsymbol{u}^*_{k,T_c-1} \tag{2.22}$$

[1] The range control is briefly described in section 2.3.4

After applying first control move of optimal sequence to system, new output $\boldsymbol{y}_{k+1}$ is measured and new optimal control sequence $\boldsymbol{u}_{k+1,T_c}$ is computed.

2.2.1 Unconstrained MPC

In MPC, the optimal control sequence $\boldsymbol{u}_k^*, \cdots, \boldsymbol{u}_{k+T_c-1}^*$ is searched. This sequence minimizes a chosen cost function with respect of dynamics of controlled system. Quadratic cost function is traditionally used.
For distinguish actual and predicted values of outputs, consider new labeling. Expression $\boldsymbol{y}_{k+n|k}$ means predicted value of output $\boldsymbol{y}$ at time $k+n$ on the basis of output at time k. It is clear that $\boldsymbol{y}_{k|k} = \boldsymbol{y}_k$.
Thus algorithm of MPC controller can be written as optimization problem over prediction and control horizon respectively:

$$\begin{aligned}
\boldsymbol{u^*}_{n,T_c-1} &= arg \min_{\boldsymbol{u}_{n,T_c-1}} J(\boldsymbol{u}_n, \cdots, \boldsymbol{u}_{n+T_c-1} | \boldsymbol{x}_n, \boldsymbol{r}_n) = \\
&= \frac{1}{2}\left[\sum_{k=n}^{n+T_p-1} (\boldsymbol{y}_k - \boldsymbol{r}_k)^T \boldsymbol{Q} (\boldsymbol{y}_k - \boldsymbol{r}_k) + \sum_{k=n}^{n+T_c-1} \boldsymbol{u}_k^T \boldsymbol{R} \boldsymbol{u}_k\right] && (2.23a)\\
\text{s.t.} \quad \boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k && (2.23b)\\
\boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k && (2.23c)\\
\boldsymbol{Q} &= \boldsymbol{Q}^T \succeq \boldsymbol{0}, \ \boldsymbol{R} = \boldsymbol{R}^T \succ \boldsymbol{0}. && (2.23d)
\end{aligned}$$

where $\boldsymbol{r}_k$ is output reference at time k, $\boldsymbol{Q} \in \mathcal{R}^{p\times p}$, $\boldsymbol{R} \in \mathcal{R}^{m\times m}$ are weighting matrices of reference tracking error and control moves. These matrices create together with prediction and control horizon tuning parameters of controller.

It is assumed that reference $\boldsymbol{r}_k$ is constant during the prediction horizon, that is:

$$\boldsymbol{r}_{k+i} = \boldsymbol{r}_k, \quad 0 \le i \le T_p \tag{2.24}$$

It is also assumed that control moves are constant beyond the control horizon, that is:

$$\boldsymbol{u}_{k+i|k} = \boldsymbol{u}_{k+T_c-1|k}, \quad T_c \le i < T_p. \tag{2.25}$$

This restriction, called *move blocking* [13], can be easily implement by modifying the predictions (2.13):

$$\boldsymbol{x}_{k+1,T_p} = \boldsymbol{V_x}\, \boldsymbol{x}_k + \boldsymbol{S_x}\, \boldsymbol{M_b}\boldsymbol{u}_{k,T_c-1} \tag{2.26}$$

$$\boldsymbol{y}_{k+1,T_p} = \boldsymbol{V_y}\, \boldsymbol{x}_k + \boldsymbol{S_y}\, \boldsymbol{M_b}\boldsymbol{u}_{k,T_c-1}, \tag{2.27}$$

where $\boldsymbol{M_b} \in \mathcal{R}^{T_p \times T_c}$. For Single-Input-Single-Output (SISO) system, $T_c = 1$ and $T_p = 4$ it is:

$$\boldsymbol{M_b} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T \tag{2.28}$$

Next, prediction matrix $\boldsymbol{\acute{S}_*}$ always represents the modified prediction matrix, that is:

$$\boldsymbol{\acute{S}_*} = \boldsymbol{S_*}\,\boldsymbol{M_b} \tag{2.29}$$

Finding optimal control sequence which minimizes quadratic cost function (2.23) represents dynamic optimization and in case when there are no constraints it can be found as analytic solution.
Using the predictions (2.26) one can formulate the cost function as:

$$\begin{aligned} J &= \left(\boldsymbol{V_y}\,\boldsymbol{x}_k + \boldsymbol{\acute{S}_y}\,\boldsymbol{u}_{k,T_c-1} - \boldsymbol{r}_{k+1,T_p}\right)^T \boldsymbol{\acute{Q}} \left(\boldsymbol{V_y}\,\boldsymbol{x}_k + \boldsymbol{\acute{S}_y}\,\boldsymbol{u}_{k,T_c-1} - \boldsymbol{r}_{k+1,T_p}\right) + \\ &+ \boldsymbol{u}_{k,T_c-1}^T \boldsymbol{\acute{R}} \boldsymbol{u}_{k,T_c-1}, \end{aligned} \tag{2.30}$$

where $\boldsymbol{\acute{Q}} \in \mathcal{R}^{(p \cdot T_p)\times(p \cdot T_p)}$, $\boldsymbol{\acute{R}} \in \mathcal{R}^{(m \cdot T_c)\times(m \cdot T_c)}$ are weighting block diagonal matrices of reference tracking error and control moves over prediction or control horizon respectively. Then the cost function can be rewritten as: [2]

$$\begin{aligned} \boldsymbol{u^*}_{k,T_c-1} &= arg \min_{\boldsymbol{u}_{k,T_c-1}} J(\boldsymbol{u}_k, \cdots, \boldsymbol{u}_{k+T_c-1} | \boldsymbol{x}_k, \boldsymbol{r}_k) = \\ &= \frac{1}{2}\boldsymbol{u}_{k,T_c-1}^T \left(\boldsymbol{\acute{S}_y}^T \boldsymbol{\acute{Q}} \boldsymbol{\acute{S}_y} + \boldsymbol{\acute{R}}\right) \boldsymbol{u}_{k,T_c-1} + \\ &+ \left[\left(\boldsymbol{V_y}\boldsymbol{x}_k - \boldsymbol{r}_{k+1,T_p}\right)^T \boldsymbol{\acute{Q}} \boldsymbol{\acute{S}_y}\right] \boldsymbol{u}_{k,T_c-1} \end{aligned} \tag{2.31a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{2.31b}$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k \tag{2.31c}$$

$$\boldsymbol{\acute{Q}} = \boldsymbol{\acute{Q}}^T \succeq \boldsymbol{0}, \ \boldsymbol{\acute{R}} = \boldsymbol{\acute{R}}^T \succ \boldsymbol{0}. \tag{2.31d}$$

Using derivate of cost function one can find optimal control sequence:

$$\boldsymbol{u}^*_{k,T_c-1} = -\boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{G}\boldsymbol{r}_k, \tag{2.32}$$

where

$$\boldsymbol{F} = \left(\boldsymbol{\acute{S}_y}^T \boldsymbol{\acute{Q}} \boldsymbol{\acute{S}_y} + \boldsymbol{\acute{R}}\right)^{-1} \boldsymbol{\acute{S}_y}^T \boldsymbol{\acute{Q}} \boldsymbol{V_y} \tag{2.33}$$

$$\boldsymbol{G} = \left(\boldsymbol{\acute{S}_y}^T \boldsymbol{\acute{Q}} \boldsymbol{\acute{S}_y} + \boldsymbol{\acute{R}}\right)^{-1} \boldsymbol{\acute{S}_y}^T \boldsymbol{\acute{Q}}, \tag{2.34}$$

[2] Additional constant, which does not effect the solution is omitted.

where matrix inversion of $\left(\boldsymbol{\acute{S}_y}^T \boldsymbol{\acute{Q}} \boldsymbol{\acute{S}_y} + \boldsymbol{\acute{R}}\right)$ is possible due assumption (2.31d).
According to equation (2.32), it can be denoted that optimal analytic MPC is state feedback controller.

2.2.2 Constrained MPC

Found control law (2.32) in previous section can not incorporate the constraints to variables in the loop, e.g. controls, states, outputs, etc.. This drawback can be removed by using the constrained MPC, which allows the constraints handling in a systematic way. Now let's define the state augmentation which will be mandatory for Explicit formulation of MPC in chapter 3 and it also simpilifies the next text.

2.2.2.1 State augmentation to reference

For reference tracking problem is advantage to augment the state space model of system (2.1) to state of reference:

$$\underbrace{\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{x_{ref}}_{k+1} \end{bmatrix}}_{\boldsymbol{\tilde{x}}_{k+1}} = \underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_p \end{bmatrix}}_{\boldsymbol{\tilde{A}}} \underbrace{\begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{x_{ref}}_k \end{bmatrix}}_{\boldsymbol{\tilde{x}}_k} + \underbrace{\begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{\tilde{B}}} \boldsymbol{u}_k \tag{2.35}$$

$$\boldsymbol{y}_k = \underbrace{\begin{bmatrix} \boldsymbol{C} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{\tilde{C}}} \boldsymbol{\tilde{x}}_k + \boldsymbol{D}\boldsymbol{u}_k \tag{2.36}$$

where $\boldsymbol{x_{ref}}_k$ are states of references and $\boldsymbol{\tilde{x}}_k$ is augmented state vector.
Main advantage of this approach is that one can easily get reference tracking error $\boldsymbol{e}_k = \boldsymbol{y}_k - \boldsymbol{r}_k$ as:

$$\boldsymbol{e}_k = \underbrace{\begin{bmatrix} \boldsymbol{C} & -\boldsymbol{I}_p \end{bmatrix}}_{\boldsymbol{\tilde{C}_e}} \boldsymbol{\tilde{x}}_k + \boldsymbol{D}\boldsymbol{u}_k, \tag{2.37}$$

and also it's prediction over prediction horizon T_p

$$\boldsymbol{e}_{k+1,T_p} = \boldsymbol{V_e}\, \boldsymbol{\tilde{x}}_k + \boldsymbol{\acute{S}_e}\, \boldsymbol{M_b}\, \boldsymbol{u}_{k,T_c-1}, \tag{2.38}$$

where $\boldsymbol{V_e}$ and $\boldsymbol{\acute{S}_e}$ are prediction matrices with reference tracking error as system output defining as (2.37). States reference values then will become a parameter by which the predictive controller is parametrized.

2.2.2.2 Cost Function with Constraints

Consider now quadratic cost function (2.23) and rewrite it using (2.35),(2.37) and let's add constraints to trajectories of states $\tilde{\boldsymbol{x}}_k \in \mathcal{X}$, outputs $\boldsymbol{y}_k \in \mathcal{Y}$ and control moves $\boldsymbol{u}_k \in \mathcal{U}$.

$$\begin{aligned}
\boldsymbol{u}^*{}_{n,T_c-1} \quad &= \quad arg \min_{\boldsymbol{u}_{n,T_c-1}} J(\boldsymbol{u}_n, \cdots, \boldsymbol{u}_{n+T_c-1}|\tilde{\boldsymbol{x}}_n) = \\
&= \quad \frac{1}{2}\left[\sum_{k=n}^{n+T_p-1} \boldsymbol{e}_k^T \boldsymbol{Q} \boldsymbol{e}_k + \sum_{k=n}^{n+T_c-1} \boldsymbol{u}_k^T \boldsymbol{R} \boldsymbol{u}_k\right] && \text{(2.39a)}\\
\text{s.t.} \quad \tilde{\boldsymbol{x}}_{k+1} &= \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}_k + \tilde{\boldsymbol{B}}\boldsymbol{u}_k, && \text{(2.39b)}\\
\boldsymbol{y}_k &= \tilde{\boldsymbol{C}}\tilde{\boldsymbol{x}}_k + \boldsymbol{D}\boldsymbol{u}_k, && \text{(2.39c)}\\
\boldsymbol{e}_k &= \tilde{\boldsymbol{C}}_{\boldsymbol{e}}\tilde{\boldsymbol{x}}_k + \boldsymbol{D}\boldsymbol{u}_k, && \text{(2.39d)}\\
\boldsymbol{Q} &= \boldsymbol{Q}^T \succeq \boldsymbol{0},\ \boldsymbol{R} = \boldsymbol{R}^T \succ \boldsymbol{0}, && \text{(2.39e)}\\
\tilde{\boldsymbol{x}}_k &\in \mathcal{X},\ \boldsymbol{y}_k \in \mathcal{Y},\ \boldsymbol{u}_k \in \mathcal{U},\ k = n, \cdots, n+T_p-1 && \text{(2.39f)}
\end{aligned}$$

Obtained control law is not linear function of the actual state and problem can not be solved analytically. In our case *Quadratic Programming* (QP) numerical algorithm is used for finding a solution. It can be shown that the solution is a piecewise affine function of system state. This conclusion is used in Explicit formulation of MPC described in chapter 3.

2.2.2.3 Quadratic Programming

Quadratic programming (QP) is a special type of mathematical optimization problem. It is the problem of optimizing (minimizing or maximizing) a quadratic function of several variables subject to linear constraints on these variables.
The quadratic programming problem can be formulated as:

$$\min_{\boldsymbol{z}} \left\{ f(\boldsymbol{z}) = \frac{1}{2}\boldsymbol{z}^T\boldsymbol{H}\boldsymbol{z} + \boldsymbol{z}^T\boldsymbol{f} + \beta | \boldsymbol{A}\boldsymbol{z} \leq \boldsymbol{b} \right\}, \tag{2.40}$$

where $\boldsymbol{H} = \boldsymbol{H}^T \in \mathcal{R}^{s\times s}$, $\boldsymbol{H} \succeq 0$, $\boldsymbol{f} \in \mathcal{R}^s$, $\boldsymbol{A} \in \mathcal{R}^{q\times s}$, $\boldsymbol{b} \in \mathcal{R}^q$, s is length of variable $\boldsymbol{z}$ and q is number of constraints.
Constraints $\boldsymbol{A}\boldsymbol{z} \leq \boldsymbol{b}$ create convex set (also polyhedral) and together with condition of positive semi-definite of $\boldsymbol{H} \succeq 0$ one can say, that even optimization problem (2.40) would be complex. For these types of optimization many numerically stable algorithms already exist. For more information about convex optimization, please see [12].

2.2.2.4 Quadratic predictive controller

Let's formulate QP for optimal control law. For purposes of the following chapters one can formulate constraints inequalities as:

$$\boldsymbol{G}_c\, \boldsymbol{u}_k \leq \boldsymbol{W}_c + \boldsymbol{E}_c\, \tilde{\boldsymbol{x}}_k, \tag{2.41}$$

with $\boldsymbol{G} \in \mathcal{R}^{1\times m}$, $\boldsymbol{W} \in \mathcal{R}$ and $\boldsymbol{E} \in \mathcal{R}^{1\times \tilde{n}}$ and $\tilde{n}$ is length of augmented system state vector.
Then cost function can be rewritten as:

$$\begin{aligned}
\boldsymbol{u}^{*}{}_{n,T_c-1} &= arg \min_{\boldsymbol{u}_{n,T_c-1}} J(\boldsymbol{u}_n, \cdots, \boldsymbol{u}_{n+T_c-1}|\tilde{\boldsymbol{x}}_n) = \\
&= \frac{1}{2}\left[\sum_{k=n}^{n+T_p-1} \boldsymbol{e}_k^T \boldsymbol{Q} \boldsymbol{e}_k + \sum_{k=n}^{n+T_c-1} \boldsymbol{u}_k^T \boldsymbol{R} \boldsymbol{u}_k\right] && (2.42a)\\
\text{s.t.} \quad \tilde{\boldsymbol{x}}_{k+1} &= \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}_k + \tilde{\boldsymbol{B}}\boldsymbol{u}_k, && (2.42b)\\
\boldsymbol{y}_k &= \tilde{\boldsymbol{C}}\tilde{\boldsymbol{x}}_k + \boldsymbol{D}\boldsymbol{u}_k, && (2.42c)\\
\boldsymbol{e}_k &= \tilde{\boldsymbol{C}}_{\boldsymbol{e}}\tilde{\boldsymbol{x}}_k + \boldsymbol{D}\boldsymbol{u}_k, && (2.42d)\\
\boldsymbol{G}_c\, \boldsymbol{u}_k &\leq \boldsymbol{W}_c + \boldsymbol{E}_c\, \tilde{\boldsymbol{x}}_k,\ k = n, \cdots, n + T_p - 1, && (2.42e)\\
\boldsymbol{Q} &= \boldsymbol{Q}^T \succeq \boldsymbol{0},\ \boldsymbol{R} = \boldsymbol{R}^T \succ \boldsymbol{0}. && (2.42f)
\end{aligned}$$

As in case of unconstrained predictive controller, using the predictions (2.26) one can get cost function in matrix form:

$$\begin{aligned}
J(\boldsymbol{u}_{k,T_c-1}|\tilde{\boldsymbol{x}}_k) &= \frac{1}{2}\boldsymbol{u}_{k,T_c-1}^T\, \boldsymbol{H}\, \boldsymbol{u}_{k,T_c-1} + \tilde{\boldsymbol{x}}_k^T\, \boldsymbol{F}\, \boldsymbol{u}_{k,T_c-1} + \frac{1}{2}\tilde{\boldsymbol{x}}_k^T\, \boldsymbol{Y}\, \tilde{\boldsymbol{x}}_k, && (2.43a)\\
\text{s.t.} \quad \boldsymbol{G}\, \boldsymbol{u}_{k,T_c-1} &\leq \boldsymbol{W} + \boldsymbol{E}\, \tilde{\boldsymbol{x}}_k, && (2.43b)
\end{aligned}$$

where matrices $\boldsymbol{H}$, $\boldsymbol{F}$ and $\boldsymbol{Y}$ are defined as follows:

$$\begin{aligned}
\boldsymbol{H} &= \acute{\boldsymbol{S}}_{\boldsymbol{e}}{}^T \acute{\boldsymbol{Q}} \acute{\boldsymbol{S}}_{\boldsymbol{e}} + \acute{\boldsymbol{R}}, && (2.44)\\
\boldsymbol{F} &= \boldsymbol{V}_{\boldsymbol{e}}{}^T \acute{\boldsymbol{Q}} \acute{\boldsymbol{S}}_{\boldsymbol{e}}, && (2.45)\\
\boldsymbol{Y} &= \boldsymbol{V}_{\boldsymbol{e}}{}^T \acute{\boldsymbol{Q}} \boldsymbol{V}_{\boldsymbol{e}}. && (2.46)
\end{aligned}$$

It should be denoted that term with matrix $\boldsymbol{Y}$ of (2.43) can be omitted because there is no influence to optimal solution $\boldsymbol{u}_{k,T_c-1}$.
Where matrices $\boldsymbol{G} \in \mathcal{R}^{q\times m\cdot T_c}$, $\boldsymbol{W} \in \mathcal{R}^{q}$ and $\boldsymbol{E} \in \mathcal{R}^{q\times \tilde{n}}$.

Cost function (2.43) is quadratic program where optimization vector is control sequence $\boldsymbol{u}_{k,T_c-1}$ and state vector $\tilde{\boldsymbol{x}}$ is parameter of this program. For $\boldsymbol{H} \succ 0$ one can conclude, that solution $\boldsymbol{u}^*_{k,T_c-1}$ is unique [3] and defined by actual state $\boldsymbol{x}_k$:

$$\boldsymbol{u}^*_{k,T_c-1}(\boldsymbol{x}_k) \quad = \quad \arg \min_{\boldsymbol{u}_{k,T_c-1}} J(\boldsymbol{u}_{k,T_c-1}|\tilde{\boldsymbol{x}}_k) \qquad \text{s.t. constraints} \tag{2.47}$$

2.3 Modifications of Model Predictive Control

In practice, the MPC is usually not implemented in such way as it is described in previous section but it is freely modified. This section gives some traditional approaches how MPC can be modified for constraints handling and for robust tracking problem.

2.3.1 Obtaining Integral Character of MPC

Equation (2.32) shows, that MPC is state feedback controller, which has only proportional character. It means, that if reference tracking is assumed, the optimal control moves can not be zero in steady state. Thus, the standard MPC is modified to obtain the integral character, which provides the zero control moves, so-called *control increments* in steady state as mentioned bellow. It should be denoted that in model uncertainty or disturbances presence, the tracking error will be still non-zero. To remove this drawback, more sophisticated methods have to be used. These methods are described in section 2.3.6.

In the MPC, obtaining integral character can be done by modifying the cost function, sometimes called Δu-formulation or control increment formulation [17]:

$$\begin{aligned}
\Delta \boldsymbol{u^*}_{n,T_c-1} \quad &= \quad arg \min_{\Delta \boldsymbol{u}_{n,T_c-1}} J(\tilde{\boldsymbol{x}}_n, \Delta \boldsymbol{u}_n, \cdots, \Delta \boldsymbol{u}_{n+T_c-1}) = \\
&= \quad \frac{1}{2}\left[\sum_{k=n}^{n+T_p-1} \boldsymbol{e}_k^T \boldsymbol{Q} \boldsymbol{e}_k + \sum_{k=n}^{n+T_c-1} \Delta \boldsymbol{u}_k^T \boldsymbol{R} \Delta \boldsymbol{u}_k\right], \qquad (2.48a) \\
\text{s.t.} \qquad & \boldsymbol{G}\, \Delta \boldsymbol{u}_{k,T_c-1} \leq \boldsymbol{W} + \boldsymbol{E}\, \tilde{\boldsymbol{x}}_k, \qquad (2.48b)
\end{aligned}$$

where $\Delta \boldsymbol{u}_k = \boldsymbol{u}_k - \boldsymbol{u}_{k-1}$ and $\Delta \boldsymbol{u}_{k,T_c-1}$ is vector of new optimizing variable - control increments. It is also necessary to augment state vector. Thus one can rewrite the

[3]This conclusion can be found in [12].

state-space equation (2.35) as

$$\underbrace{\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{u}_k \\ \boldsymbol{x_{ref}}_{k+1} \end{bmatrix}}_{\tilde{\boldsymbol{x}}_{k+1}} = \underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_m & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_p \end{bmatrix}}_{\tilde{\boldsymbol{A}}} \underbrace{\begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_{k-1} \\ \boldsymbol{x_{ref}}_k \end{bmatrix}}_{\tilde{\boldsymbol{x}}_k} + \underbrace{\begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{I}_m \\ \boldsymbol{0} \end{bmatrix}}_{\tilde{\boldsymbol{B}}} \Delta\boldsymbol{u}_k \tag{2.49}$$

$$\boldsymbol{y}_k = \underbrace{\begin{bmatrix} \boldsymbol{C} & \boldsymbol{D} & \boldsymbol{0} \end{bmatrix}}_{\tilde{\boldsymbol{C}}} \tilde{\boldsymbol{x}}_k + \boldsymbol{D}\Delta\boldsymbol{u}_k, \tag{2.50}$$

$$\boldsymbol{e}_k = \underbrace{\begin{bmatrix} \boldsymbol{C} & \boldsymbol{D} & -\boldsymbol{I}_p \end{bmatrix}}_{\tilde{\boldsymbol{C}}_{\boldsymbol{e}}} \tilde{\boldsymbol{x}}_k, \tag{2.51}$$

where states $\boldsymbol{u}_{k-1}$ represent last control move.
Output of controller is then sequence of control increments $\Delta\boldsymbol{u}_k$. To obtain the control move to process it is necessary to sum the actual control increment $\Delta\boldsymbol{u}_k$ with last control move $\boldsymbol{u}_{k-1}$, that is

$$\boldsymbol{u}_k = \Delta\boldsymbol{u}_k + \boldsymbol{u}_{k-1}. \tag{2.52}$$

Last control move $\boldsymbol{u}_{k-1}$ can be internally computed by the controller or it is necessary to include this as next input to controller.
In case of controller usage without weighting the control increments (2.42a), the cost function is not set up [4] so that the minimum (in steady state) corresponds to zero tracking error, that is, the optimum control will necessarily cause offset. The Δu-formulation causes the problem transformation to one, where in steady state (for nominal model) for the reference tracking error and *control increment* is next valid

$$\boldsymbol{e}_{ss} \to \boldsymbol{0}, \qquad \Delta\boldsymbol{u}_{ss} \to \boldsymbol{0}, \tag{2.53}$$

Then value of cost function (2.48) is also minimum, that is:

$$J(\tilde{\boldsymbol{x}}_n, \Delta\boldsymbol{u}_n, \cdots, \Delta\boldsymbol{u}_{n+T_c-1})_{ss} = 0, \tag{2.54}$$

In case of cost function usage (2.48), it is necessary to change assumption (2.25) of constant control moves $\boldsymbol{u}_k$ beyond control horizon and use new blocking strategy [13]:

$$\Delta\boldsymbol{u}_{k+i|k} = \boldsymbol{0}, \quad T_c \leq i < T_p. \tag{2.55}$$

[4]For instance, the weighting matrix of control $\boldsymbol{R}$ is too high.

This can be easily implemented by picking out first $T_c \cdot m$ columns of prediction matrix $\boldsymbol{S}_*$. Predictions of states and outputs are then:

$$\boldsymbol{x}_{k+1,T_p} = \boldsymbol{V}_{\boldsymbol{x}}\,\tilde{\boldsymbol{x}}_k + \breve{\boldsymbol{S}}_{\boldsymbol{x}}\,\Delta\boldsymbol{u}_{k,T_c-1} \tag{2.56}$$

$$\boldsymbol{y}_{k+1,T_p} = \boldsymbol{V}_{\boldsymbol{y}}\,\tilde{\boldsymbol{x}}_k + \breve{\boldsymbol{S}}_{\boldsymbol{y}}\,\Delta\boldsymbol{u}_{k,T_c-1}, \tag{2.57}$$

where $\breve{\boldsymbol{S}}_*$ is first $T_c \cdot m$ columns of prediction matrix $\boldsymbol{S}_*$.

2.3.2 Constraints Handling

As it was already said, great advantage of MPC is ability to take constraints of any system variables into account. But on the other hand, constraints presence is main reason why the optimization problem (2.43) or (2.48) is so computationally demanding. When sampling times become so short that computation times for QP solution can no longer be neglected, specialized algorithms that exploit the structure of the QPs arising in MPC problems become necessary. One of these algorithms is *Explicit formulation of QP* described in chapter 3.

The constraints may occur on any variable in the loop. The MPC controller has to ensure that none of the constraints are violated by the optimal predictions over the prediction horizon in the cost function J.

In practical problems it is common to find that the desirable constraints (2.43b) are infeasible. It means they can not all be satisfied simultaneously. In this case, if constraints do not admit a solution, then MPC optimisation is ill posed and has no solution. This case may be dangerous in real process as the resulting control would be arbitrary. To avoid this situation, constraints should be placed into two categories - *Hard constraints* and *Soft constraints* [56].
For purposes of the following chapters the constraints inequalities will be formulated in following form:

$$\boldsymbol{G}\,\Delta\boldsymbol{u}_{k,T_c-1} \leq \boldsymbol{W} + \boldsymbol{E}\,\tilde{\boldsymbol{x}}_k, \tag{2.58}$$

where n_S is number of *slack variables* (soft constraints) defined later, $\boldsymbol{G} \in \mathcal{R}^{q\times(m\cdot T_c+n_S)}$, $\boldsymbol{W} \in \mathcal{R}^q$ and $\boldsymbol{E} \in \mathcal{R}^{q\times(\tilde{n}+n_S)}$.
Next, use of cost function (2.48) weighing control increments is assumed.

2.3.3 The Hard Constraints

The hard constraints are constraints which must be satisfied. The hard constraint are naturally used for limit the control variable on actuators or on valves (which must lie between 0% and 100%) open.

2.3.3.1 Control Rate Constraints

Sometimes it is necessary to limit the control rate especially in chemical industry, where time changes of temperature has to be finite.
Take upper and lower limits on the control rate to be

$$\underline{\Delta \boldsymbol{u}} \leq \Delta \boldsymbol{u} \leq \overline{\Delta \boldsymbol{u}} \tag{2.59}$$

Using assumption (2.55), control increment are predicted to be zero beyond the control horizon T_c, one can write constraints over control horizon as:

$$\underbrace{\begin{bmatrix} \underline{\Delta \boldsymbol{u}} \\ \underline{\Delta \boldsymbol{u}} \\ \vdots \\ \underline{\Delta \boldsymbol{u}} \end{bmatrix}}_{\underline{\Delta \boldsymbol{U}}} \leq \Delta \boldsymbol{u}_{k,T_c-1} \leq \underbrace{\begin{bmatrix} \overline{\Delta \boldsymbol{u}} \\ \overline{\Delta \boldsymbol{u}} \\ \vdots \\ \overline{\Delta \boldsymbol{u}} \end{bmatrix}}_{\overline{\Delta \boldsymbol{U}}} \tag{2.60}$$

This can be rewritten in form of linear inequalities (2.48b),

$$\begin{bmatrix} \boldsymbol{I}_{m \cdot T_c} \\ -\boldsymbol{I}_{m \cdot T_c} \end{bmatrix} \Delta \boldsymbol{u}_{k,T_c-1} \leq \begin{bmatrix} \overline{\Delta \boldsymbol{U}} \\ -\underline{\Delta \boldsymbol{U}} \end{bmatrix} + \boldsymbol{0}\, \tilde{\boldsymbol{x}}_k \tag{2.61}$$

2.3.3.2 Control Constraints

More often one wants to limit control variables because of finite response of actuators. Consider upper and lower limit on the control variable

$$\underline{\boldsymbol{u}} \leq \boldsymbol{u} \leq \overline{\boldsymbol{u}}. \tag{2.62}$$

This has to be satisfied over prediction horizon T_p with assumption (2.25) that control moves are constant beyond the control horizon. Thus, one can write

$$\underbrace{\begin{bmatrix} \underline{\boldsymbol{u}} \\ \underline{\boldsymbol{u}} \\ \vdots \\ \underline{\boldsymbol{u}} \end{bmatrix}}_{\underline{\boldsymbol{U}}} \leq \boldsymbol{u}_{k,T_c-1} \leq \underbrace{\begin{bmatrix} \overline{\boldsymbol{u}} \\ \overline{\boldsymbol{u}} \\ \vdots \\ \overline{\boldsymbol{u}} \end{bmatrix}}_{\overline{\boldsymbol{U}}}. \tag{2.63}$$

For this, one must express the future control over prediction horizon. This can be done by creating new output of the augmented system (2.49)

$$\boldsymbol{u}_k = \underbrace{\begin{bmatrix} \mathbf{0} & \boldsymbol{I_m} & \mathbf{0} \end{bmatrix}}_{\tilde{\boldsymbol{C}}_{\boldsymbol{u}}} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_{k-1} \\ \boldsymbol{x_{ref}}_k \end{bmatrix} + \Delta \boldsymbol{u}_k, \tag{2.64}$$

and computing it's prediction over prediction horizon T_c

$$\boldsymbol{u}_{k,T_c-1} = \boldsymbol{V_u}\, \tilde{\boldsymbol{x}}_k + \breve{\boldsymbol{S}}_{\boldsymbol{u}}\, \Delta \boldsymbol{u}_{k,T_c-1}, \tag{2.65}$$

where $\boldsymbol{V_u}$ and $\breve{\boldsymbol{S}}_{\boldsymbol{u}}$ are prediction matrices for augmented system (2.49) with output (2.64). Then one can rewrite (2.63) in form of linear inequalities (2.48b),

$$\begin{bmatrix} \breve{\boldsymbol{S}}_{\boldsymbol{u}} \\ -\breve{\boldsymbol{S}}_{\boldsymbol{u}} \end{bmatrix} \Delta \boldsymbol{u}_{k,T_c-1} \leq \begin{bmatrix} \overline{\boldsymbol{U}} \\ -\underline{\boldsymbol{U}} \end{bmatrix} + \begin{bmatrix} -\boldsymbol{V_u} \\ \boldsymbol{V_u} \end{bmatrix} \tilde{\boldsymbol{x}}_k. \tag{2.66}$$

2.3.3.3 Output Constraints

The output constraints can be set up analogously to control constraints in section 2.3.3.2. It should be noted that the output constraints are taken up to the prediction horizon T_p. From output equation (2.50) and it's prediction (2.57) one can write

$$\begin{bmatrix} \breve{\boldsymbol{S}}_{\boldsymbol{y}} \\ -\breve{\boldsymbol{S}}_{\boldsymbol{y}} \end{bmatrix} \Delta \boldsymbol{u}_{k,T_c-1} \leq \begin{bmatrix} \overline{\boldsymbol{Y}} \\ -\underline{\boldsymbol{Y}} \end{bmatrix} + \begin{bmatrix} -\boldsymbol{V_y} \\ \boldsymbol{V_y} \end{bmatrix} \tilde{\boldsymbol{x}}_k, \tag{2.67}$$

where $\underline{\boldsymbol{Y}}$, $\overline{\boldsymbol{Y}}$ are lower and upper limits of output over prediction horizon.

2.3.4 The Soft Constraints

Soft constraints are the ones, which should be satisfied if possible. It is assumed that if necessary, they can be violated at some penalty, for example a loss of product quality. Usually soft constraints are on output/states although they could be applied to control variable too.

The soft constraints are formulated as next additive terms to cost function with appropriate weighting coefficients. The soft constraints are the base for the *Range Control Algorithm* [43],[52], where the controlled variables are enabled to freely vary in specified range. The control action is without changes as long as the predictions of the quantities are inside of the range. Violation of the specified range is penalized, usually by a quadratic cost and then the appropriate control sequence that ensures return back to range is generated. Ranges can be time-varying. The limits of the time-varying range create so-called funnels in the time and they specify the limits for the time trajectory of the system quantities. Using the soft constraints one can also find a feasible solution of MPC optimization.

2.3.4.1 Output Constraints

Consider a lower and upper soft constraints on outputs, that is:

$$\underline{\boldsymbol{y}} - \underline{\boldsymbol{\epsilon}} \leq \boldsymbol{y} \leq \overline{\boldsymbol{y}} + \overline{\boldsymbol{\epsilon}} \tag{2.68}$$

where $\underline{\boldsymbol{y}}$, $\overline{\boldsymbol{y}}$ are lower and upper limits of outputs and $\underline{\boldsymbol{\epsilon}}$, $\overline{\boldsymbol{\epsilon}}$ are the *slack variables*, the values how much can be lower and upper limit violated.
Let's create substitutes

$$\boldsymbol{\epsilon} = \begin{bmatrix} \overline{\boldsymbol{\epsilon}} & \underline{\boldsymbol{\epsilon}} \end{bmatrix}^T \qquad \underline{\overline{\boldsymbol{y}}} = \begin{bmatrix} \overline{\boldsymbol{y}} & \underline{\boldsymbol{y}} \end{bmatrix}^T \qquad \boldsymbol{\rho} = \begin{bmatrix} \overline{\boldsymbol{\rho}} & \mathbf{0} \\ \mathbf{0} & \underline{\boldsymbol{\rho}} \end{bmatrix}, \tag{2.69}$$

where $\underline{\boldsymbol{\rho}}$ and $\overline{\boldsymbol{\rho}}$ are penalty matrices of violations of the lower and upper constraints.

The goal of MPC is then to minimize the violations $\underline{\boldsymbol{\epsilon}}$ and $\overline{\boldsymbol{\epsilon}}$.
Let's modify the cost function (2.48)

$$\begin{aligned} J(\Delta \boldsymbol{u}_{k,T_c-1}, \boldsymbol{\epsilon}|\tilde{\boldsymbol{x}}_k) &= \\ &= \frac{1}{2}\Delta \boldsymbol{u}_{k,T_c-1}^T \boldsymbol{H}\, \Delta \boldsymbol{u}_{k,T_c-1} + \frac{1}{2}\boldsymbol{\epsilon}^T \boldsymbol{\rho} \boldsymbol{\epsilon} + \tilde{\boldsymbol{x}}_k^T \boldsymbol{F}\, \Delta \boldsymbol{u}_{k,T_c-1} + \frac{1}{2}\tilde{\boldsymbol{x}}_k^T \boldsymbol{Y}\, \tilde{\boldsymbol{x}}_k, \end{aligned} \tag{2.70}$$

Thus, one can augment the minimized control sequence to violations, that is

$$\boldsymbol{h} = \begin{bmatrix} \Delta \boldsymbol{u}_{k,T_c-1} & \boldsymbol{\epsilon} \end{bmatrix}^T. \tag{2.71}$$

Then cost function can be rewritten to matrix form as (2.43):

$$\boldsymbol{h}^* = \arg\min_{\boldsymbol{h}} J(\boldsymbol{h}|\tilde{\boldsymbol{x}}_k) = \frac{1}{2}\boldsymbol{h}^T \acute{\boldsymbol{H}} \boldsymbol{h} + \begin{bmatrix} \tilde{\boldsymbol{x}}_k \\ \underline{\overline{\boldsymbol{y}}} \end{bmatrix}^T \acute{\boldsymbol{F}} \boldsymbol{h} + \begin{bmatrix} \tilde{\boldsymbol{x}}_k \\ \underline{\overline{\boldsymbol{y}}} \end{bmatrix}^T \acute{\boldsymbol{Y}} \begin{bmatrix} \tilde{\boldsymbol{x}}_k \\ \underline{\overline{\boldsymbol{y}}} \end{bmatrix}, \tag{2.72}$$

where

$$\acute{\boldsymbol{H}} = \begin{bmatrix} \boldsymbol{H} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\rho} \end{bmatrix} \qquad \acute{\boldsymbol{F}} = \begin{bmatrix} \boldsymbol{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \qquad \acute{\boldsymbol{Y}} = \begin{bmatrix} \boldsymbol{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{2.73}$$

The soft constraints then can be written as linear inequalities

$$\begin{bmatrix} \breve{\boldsymbol{S}}_{\boldsymbol{y}} & -\mathbf{1} & \mathbf{0} \\ -\breve{\boldsymbol{S}}_{\boldsymbol{y}} & \mathbf{0} & -\mathbf{1} \end{bmatrix} \boldsymbol{h} \leq \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -\boldsymbol{V}_{\boldsymbol{y}} & \mathbf{1} & \mathbf{0} \\ \boldsymbol{V}_{\boldsymbol{y}} & \mathbf{0} & -\mathbf{1} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}}_k \\ \underline{\overline{\boldsymbol{y}}} \end{bmatrix}. \tag{2.74}$$

2.3.5 State Estimation

From (2.32), it is clear that MPC is state feedback controller. For it's operation it is necessary to know a state values of a controlled system [55]. But in many cases in practice, it may be either impossible or simply impractical to obtain measurements for all states. In particular, some states may not be available for measurement at all. There are also cases where it may be impractical to obtain state measurements from otherwise available states because economic reasons (e.g. some sensors may be expensive) or because of technical reasons (e.g. too noisy environment for any useful measurements). At figure 2.2, such state observer is shown.

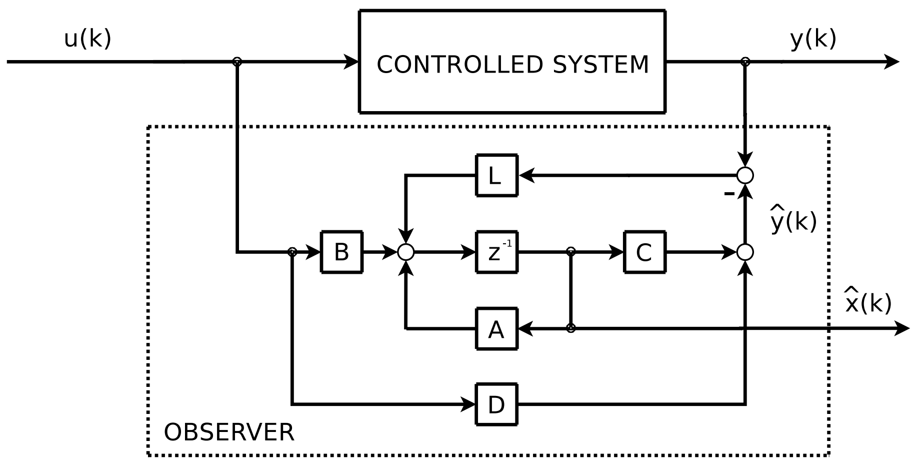

Figure 2.2: State observer

Thus, it is necessary to estimate the state of the system from available measurements, typically outputs and controls. Given system parameters $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ and the values of the controls and outputs over a time interval, it is possible to estimate the state when the system is observable [5].

For distinguish actual, last or predicted values of variables and on what their predictions are based, consider new labeling. Expression $\boldsymbol{s}_{k+1|k}$ means predicted value of variable $\boldsymbol{s}$ based on data up to time k. Expression $\boldsymbol{s}_{k|k-1}$ means estimation value of variable $\boldsymbol{s}$ based on data up to time $k-1$.

[5] That means, matrix of observability $\boldsymbol{\mathcal{O}}$ must be of full row rank.

2.3.5.1 The Predictor Estimator

The full-order identical estimator [2] of the full state $\boldsymbol{x}_k$ for system (2.1), can be constructed as a system

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{A}\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{L}\boldsymbol{\varepsilon}_{k|k-1}, \tag{2.75}$$

where $\hat{\boldsymbol{y}}_{k|k-1} = \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{D}\boldsymbol{u}_k$, $(\hat{\cdot})$ labels estimation and $\boldsymbol{L} \in \mathcal{R}^{n\times p}$ is state injection gain. We also define the *output prediction error* as

$$\boldsymbol{\varepsilon}_{k|k-1} = \boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1} \tag{2.76}$$

Dynamic of state estimation error is then

$$\boldsymbol{e}_{k+1|k} = \boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k} = (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\,\boldsymbol{e}_{k|k}. \tag{2.77}$$

If the eigenvalues of $\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}$ are inside the open unit disk of the complex plane, then $\boldsymbol{e}_k \to \boldsymbol{0}$ as $k \to \infty$, independently of the initial condition $\boldsymbol{e}_0 = \boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0$. This asymptotic state estimator is called *Luenberger observer* [44].

If pair $(\boldsymbol{A}, \boldsymbol{C})$ is not observable but the unobservable eigenvalues are stable, i.e., $(\boldsymbol{A}, \boldsymbol{C})$ is detectable, then error $\boldsymbol{e}_k$ will still tend to zero asymptotically. Hower, the unobservable eigenvalues will appear in this case as eigenvalues of $\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C}$, and they may affect the speed of the response of the estimator in the undesirable way.

2.3.5.2 The Filter

From (2.75) one can deduce that the state estimate $\hat{\boldsymbol{x}}_k$ is based on measurements up to and including $\boldsymbol{y}_{k-1}$. It is often in interest in applications to determine the state estimate $\hat{\boldsymbol{x}}_k$ based on measurement up to and including $\boldsymbol{y}_k$. If the computation time required to calculate $\hat{\boldsymbol{x}}_k$ is short compared with the sample period in a sampled-data system, then it is certainly practically possible to determine the estimate $\hat{\boldsymbol{x}}_k$ before $\boldsymbol{x}_{k+1}$ and $\boldsymbol{y}_{k+1}$ are generated by the observed system. If the state estimate, which is based on current measurements of $\boldsymbol{y}_k$, is to be used to control the system, then the unavoidable computational delays should be taken into consideration.
Denote $\bar{\boldsymbol{x}}_k$ the current state estimate based on measurements up through $\boldsymbol{y}_k$. Thus the *filter* or also the *current estimator* [2] is system described as

$$\bar{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{L}_c\left(\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1}\right), \tag{2.78}$$

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{A}\bar{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1}, \tag{2.79}$$

It can be shown, that if

$$\boldsymbol{L} = \boldsymbol{A}\boldsymbol{L}_c, \tag{2.80}$$

then the prediction estimator and filter are equivalent. For current estimator one can get dynamic of estimation error $\bar{\boldsymbol{e}}_k = \boldsymbol{x}_k - \bar{\boldsymbol{x}}_k$

$$\bar{\boldsymbol{e}}_{k+1|k} = (\boldsymbol{A} - \boldsymbol{L}_c\boldsymbol{C}\boldsymbol{A})\,\bar{\boldsymbol{e}}_{k|k}. \tag{2.81}$$

2.3.5.3 The Kalman Filter

In 1960, R.E. Kalman published his famous paper [67] describing a recursive solution to the discrete-data linear filtering problem. Since that time, due in large part to advances in digital computing, the Kalman filter has been the subject of extensive research [58] and application, particularly in the area of autonomous or assisted navigation.
The Kalman filter is the estimator for stochastic process. Consider the linear time invariant stochastic, discrete system

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{v}_k & (2.82)\\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k + \boldsymbol{e}_k, & (2.83) \end{aligned}$$

where $\boldsymbol{v}_k$, $\boldsymbol{e}_k$ are non correlated [6] discrete white noises with known covariances.

$$cov\left\{\begin{bmatrix}\boldsymbol{v}_k\\ \boldsymbol{e}_k\end{bmatrix}\cdot\begin{bmatrix}\boldsymbol{v}_k\\ \boldsymbol{e}_k\end{bmatrix}^T\right\} = \begin{bmatrix}\boldsymbol{Q} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{R}\end{bmatrix}. \tag{2.84}$$

Note that matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ create the tunning parameters of the filter. They say, how the designer trusts to the system model and to measurements of the process outputs.

Because of stochastic formulation, Kalman filter is looking for the state values of the system in linear means squares sense [7]. It implements *Linear Mean Squares* (LMS) algorithm, thus mean value of the states is

$$E\left\{\boldsymbol{x}_{k|k-1}\right\} = \hat{\boldsymbol{x}}_{k|k-1}, \tag{2.85}$$

and covariance of the state is

$$cov\left\{\boldsymbol{x}_{k|k-1}\right\} = \boldsymbol{P}_{k|k-1} \tag{2.86}$$

[6] If the noises are correlated, there exists a transform, which converts the system into one with non correlated noises.

[7] Gaussian noise are assumed.

We define the *update step* of the filter (according to [31]):

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \overline{\boldsymbol{L}}_k \left(\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{D}\boldsymbol{u}_k\right), && (2.87) \\ \boldsymbol{P}_{k|k} &= \boldsymbol{P}_{k|k-1} - \overline{\boldsymbol{L}}_k \left(\boldsymbol{C}\boldsymbol{P}_{k|k-1}\boldsymbol{C}^T + \boldsymbol{R}\right) \overline{\boldsymbol{L}}_k^T && (2.88) \end{aligned}$$

where $\overline{\boldsymbol{L}}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{C}^T \left(\boldsymbol{C}\boldsymbol{P}_{k|k-1}\boldsymbol{C}^T + \boldsymbol{R}\right)^{-1}$ is the *Kalman filter gain of update step* or also *innovation gain*.
And *predict step* of the filter:

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1|k} &= \boldsymbol{A}\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{B}\boldsymbol{u}_k && (2.89) \\ \boldsymbol{P}_{k+1|k} &= \boldsymbol{A}\boldsymbol{P}_{k|k}\boldsymbol{A}^T + \boldsymbol{Q} && (2.90) \end{aligned}$$

Both steps can be joined together into one step algorithm:

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1|k} &= \left(\boldsymbol{A} - \boldsymbol{A}\overline{\boldsymbol{L}}_k\boldsymbol{C}\right)\hat{\boldsymbol{x}}_{k|k-1} + \left(\boldsymbol{B} - \boldsymbol{A}\overline{\boldsymbol{L}}_k\boldsymbol{D}\right)\boldsymbol{u}_k + \boldsymbol{A}\overline{\boldsymbol{L}}_k\boldsymbol{y}_k && (2.91) \\ \boldsymbol{P}_{k+1|k} &= \boldsymbol{A}\boldsymbol{P}_{k|k-1}\boldsymbol{A}^T - \boldsymbol{A}\overline{\boldsymbol{L}}_k \left(\boldsymbol{C}\boldsymbol{P}_{k|k-1}\boldsymbol{C}^T + \boldsymbol{R}\right) \overline{\boldsymbol{L}}_k^T \boldsymbol{A}^T + \boldsymbol{Q} && (2.92) \end{aligned}$$

It is also possible to use the *Kalman filter gain* defined as:

$$\boldsymbol{L}_k = \boldsymbol{A}\overline{\boldsymbol{L}}_k = \boldsymbol{A}\boldsymbol{P}_{k|k-1}\boldsymbol{C}^T \left(\boldsymbol{C}\boldsymbol{P}_{k|k-1}\boldsymbol{C}^T + \boldsymbol{R}\right)^{-1} \tag{2.93}$$

And get the one step Kalman algorithm:

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1|k} &= \left(\boldsymbol{A} - \boldsymbol{L}_k\boldsymbol{C}\right)\hat{\boldsymbol{x}}_{k|k-1} + \left(\boldsymbol{B} - \boldsymbol{L}_{\boldsymbol{k}}\boldsymbol{D}\right)\boldsymbol{u}_k + \boldsymbol{L}_k\boldsymbol{y}_k && (2.94) \\ \boldsymbol{P}_{k+1|k} &= \boldsymbol{A}\boldsymbol{P}_{k|k-1}\boldsymbol{A}^T - \boldsymbol{L}_k \left(\boldsymbol{C}\boldsymbol{P}_{k|k-1}\boldsymbol{C}^T + \boldsymbol{R}\right) \boldsymbol{L}_k^T \boldsymbol{A}^T + \boldsymbol{Q} && (2.95) \end{aligned}$$

Equation (2.95) together with Kalman filter gain definition (2.93) is well known *Discrete Riccati Equation* (DRE) [39]. In many cases its solution converges [8] to limit value $\boldsymbol{P}$

$$\boldsymbol{P} = \lim_{k\to\infty} \boldsymbol{P}_{k+1|k}. \tag{2.96}$$

If such solution exists, then it is also valid for *Discrete Algebraic Riccati Equation* (DARE) defined as

$$\boldsymbol{P} = \boldsymbol{A}\boldsymbol{P}\boldsymbol{A}^T - \boldsymbol{L}\left(\boldsymbol{C}^T\boldsymbol{P}\boldsymbol{C} + \boldsymbol{R}\right)\boldsymbol{L}^T + \boldsymbol{Q}, \tag{2.97}$$

where $\boldsymbol{L}$ is limit value of Kalman filter gain

$$\boldsymbol{L} = \boldsymbol{A}\boldsymbol{P}\boldsymbol{C}^T \left(\boldsymbol{C}\boldsymbol{P}\boldsymbol{C}^T + \boldsymbol{R}\right)^{-1} \tag{2.98}$$

There are two variants of discrete-time Kalman estimators [61]:

[8] Conditions of the existence of the limit solution can be found for example in [58].

2.3.5.3.1 Current estimator The *current estimator* generates output estimates $\hat{\boldsymbol{y}}_{k|k}$ and state estimates $\hat{\boldsymbol{x}}_{k|k}$ using all available measurements up to $\boldsymbol{y}_k$. Output equations of this filter is then

$$\hat{\boldsymbol{y}}_{k|k} = \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{D}\boldsymbol{u}_k + \boldsymbol{C}\overline{\boldsymbol{L}}\left(\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{D}\boldsymbol{u}_k\right) \tag{2.99}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \overline{\boldsymbol{L}}\underbrace{\left(\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1} - \boldsymbol{D}\boldsymbol{u}_k\right)}_{\text{innovation}}, \tag{2.100}$$

where the innovation gain $\overline{\boldsymbol{L}} = \boldsymbol{P}\boldsymbol{C}^T\left(\boldsymbol{C}\boldsymbol{P}\boldsymbol{C}^T + \boldsymbol{R}\right)^{-1}$ updates the prediction $\hat{\boldsymbol{x}}_{k|k-1}$ using the new measurements $\boldsymbol{y}_k$ with *innovation.*
Note that the current estimator should be used every time where the designer needs precise estimation and computation time of the control action of the controller are assumed to be very small in comparison with the sampling time.

2.3.5.3.2 Delayed estimator The delayed estimator generates output estimates $\hat{\boldsymbol{y}}_{k|k}$ and state estimates $\hat{\boldsymbol{x}}_{k|k}$ using measurements only up to $\boldsymbol{y}_{k-1}$. This estimator is easier to implement inside control loops and has the output equation

$$\hat{\boldsymbol{y}}_{k|k-1} = \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{D}\boldsymbol{u}_k \tag{2.101}$$

$$\hat{\boldsymbol{x}}_{k|k-1} = \hat{\boldsymbol{x}}_{k|k-1} \tag{2.102}$$

2.3.6 Unknown Input Observer

Closed-loop performance of model-based control (for example MPC) algorithm is directly related to model accuracy [50]. In practice, modeling error and unmeasured disturbances can lead to steady-state offset unless precautions are taken in the control design. Elimination of steady-state offset is accomplished in two basic ways. The first method involves modifying the control objective to include integration of the tracking error. This method, employed by the PID control algorithm, can also be used in MPC framework [38]. In the MPC framework, the integral term is incorporated by augmenting the process model with tracking error states [50]. For large-scale systems, this augmentation can significantly increase the computational time of the dynamic optimization. This effect is particularly bothersome when explicit *Model Predictive Control* is used, where the complexity increases quickly with the number of state variables [9]. This approach also requires an anti-windup algorithm for the integral term to prevent an unnecessary, and sometimes

[9]The Explicit Model Predictive Control is briefly described in chapter 3.

costly performance penalty.

A second general approach to eliminating steady-state offset involves augmenting the process model [10] with a disturbance model which is used to estimate and predict the mismatch between measured and predicted outputs. The mismatch can be caused by the unknown inputs, as unmeasured or measured disturbances, unknown control action, or unmodeled system dynamics or its uncertainty.

The method which allows an estimation of the unknown input to controlled system is called *Unknown Input Observer* (UIO) [18],[23]. Situation is shown in figure 2.3. Here, an unknown inputs v and e and disturbance d enter to the controlled system and the Unknown input observer estimate them to eliminate the steady-state offset of output y.

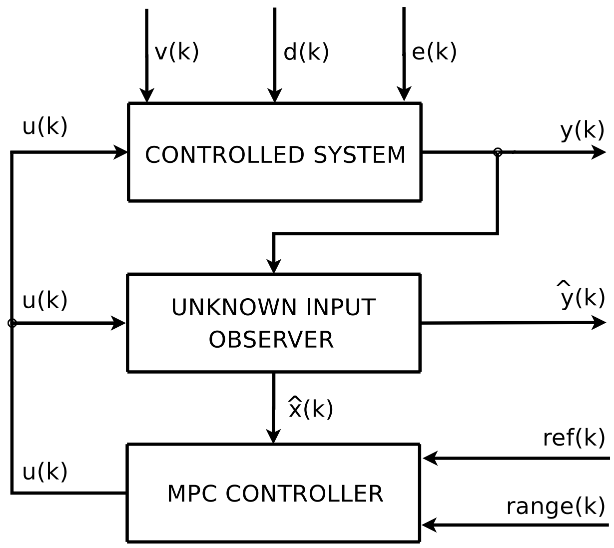

Figure 2.3: MPC controller with unknown input observer

In MPC framework, the augmentation of the process model to include a constant step disturbances to eliminate steady-state offset is widely used when tracking a constant reference. This disturbance, which is estimated from the measured process variables, is generally to remain constant in the future and its effect on the controlled variables is removed by shifting the steady-state target for the controller.

Closed loop controller performance is directly related to how accurately the disturbance model represents the actual disturbances entering the process. This subject has become known as the *internal model principle* [21].

[10] It should be denoted that augmented process model must be also used for prediction of the MPC controller.

In industrial model predictive control implementations, offset-free control is commonly achieved through the use of a step output disturbance model. A constant output disturbance model can be constructed using the following augmented state-space model [50]

$$\begin{aligned} \begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{p}_{k+1} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} & 0 \\ 0 & \boldsymbol{I}_{s_p} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{p}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ 0 \end{bmatrix} \boldsymbol{u}_k, && (2.103) \\ \boldsymbol{y}_k &= \begin{bmatrix} \boldsymbol{C} & \boldsymbol{G}_p \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{p}_k \end{bmatrix} + \boldsymbol{D}\boldsymbol{u}_k, && (2.104) \end{aligned}$$

in which $\boldsymbol{p} \in \mathcal{R}^{s_p}$, s_p is the number of augmented output disturbance states, and $\boldsymbol{G}_p$ determines the effect of these states on the output. In the standard industrial MPC implementation, $\boldsymbol{G}_p = \boldsymbol{I}_p$ and the output disturbance is estimated as $\boldsymbol{p}_k = \boldsymbol{y}_k - \boldsymbol{C}\boldsymbol{x}_k - \boldsymbol{D}\boldsymbol{u}_k$. The result is deadbeat observer for the output disturbance states and an open-loop observer for the model states.

The output disturbance model (2.103) is simple to implement but may lead to poor performance when a disturbance enters elsewhere in the loop. The solution is to augment the system model with an input or state disturbance. The state disturbance can be represented as

$$\begin{aligned} \begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{d}_{k+1} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} & \boldsymbol{G}_d \\ 0 & \boldsymbol{I}_{s_d} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{d}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ 0 \end{bmatrix} \boldsymbol{u}_k, && (2.105) \\ \boldsymbol{y}_k &= \begin{bmatrix} \boldsymbol{C} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{d}_k \end{bmatrix} + \boldsymbol{D}\boldsymbol{u}_k, && (2.106) \end{aligned}$$

where $\boldsymbol{d} \in \mathcal{R}^{s_d}$, s_d is the number of augmented disturbance states, and $\boldsymbol{G}_d$ determines the effect of the disturbance. Input disturbances can be represented by choosing $\boldsymbol{G}_d = \boldsymbol{B}$.

For linear systems, a general block diagonal disturbance model that includes both state/input and output disturbances is

$$\begin{aligned} \tilde{\boldsymbol{x}}_{k+1} &= \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}_k + \tilde{\boldsymbol{B}}\boldsymbol{u}_k \\ \boldsymbol{y}_k &= \tilde{\boldsymbol{C}}\tilde{\boldsymbol{x}}_k + \boldsymbol{D}\boldsymbol{u}_k, && (2.107) \end{aligned}$$

in which the augmented state vector and system matrices are defined as follows:

$$\tilde{\boldsymbol{x}}_k = \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{d}_k \\ \boldsymbol{p}_k \end{bmatrix}, \quad \tilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{G}_d & 0 \\ 0 & \boldsymbol{I}_{s_d} & 0 \\ 0 & 0 & \boldsymbol{I}_{s_p} \end{bmatrix}, \quad \tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{B} \\ 0 \\ 0 \end{bmatrix}, \quad \tilde{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{C} & 0 & \boldsymbol{G}_p \end{bmatrix} \tag{2.108}$$

On two next examples, the output and the state disturbance rejection will be shown on general SISO system with the requirements of offset-free control using the unknown input observer method.

Example 2.1 (Output disturbance rejection): Let's also consider presence of the output disturbance affected to the system process output. At figure 2.4, closed-loop response of such case is shown. At time 8 s and 14 s disturbances change it's value. Controller without disturbance model (left pictures) doesn't react to entered disturbance and steady state tracking error is present. On other hand, controller augmented to disturbance model (right pictures) leads to offset-free control. The observer estimates the entered disturbance and its influence to system process is eliminated

Example 2.2 (State disturbance rejection): Now consider that state disturbance enters to the system and states of the system can not be measured. At figure 2.5, closed-loop response for the case without and with process model augmentation is shown. At time 8 s, an unmeasured constant disturbance enters the process. The left pictures shows response without augmentation. Due to the disturbance, output of the system grows up out of desired reference, but controller does not react. At the left pictures, controller reacts to the disturbance and thanks to disturbance observer estimation at the steady-state there is an offset-free control.

The most basic goal for disturbance model design is to ensure that the augmented disturbance states are observable. Because these states are not asymptotically stable, they will be observable if and only if the augmented process model (2.107) is detectable [11].
In [50], authors proved the existence of detectable system (2.107) when the total number of disturbance states is equal to the number of outputs ($s_p + s_d = p$). They also proved that augmented system (2.107) is not detectable if the total number of augmented disturbance states exceeds the number of outputs.
In [47], authors note that in the case where only subset of measured variables is actually to be controlled with zero offset, previous method yields too complex models. By internal model principle, it should be possible to add as many disturbance states as there are outputs to control with zero offset. Adding more variables then necessary will lead to unnecessarily complex optimization problem in MPC.

The best achievable closed-loop system performance will be obtained when the unmeasured disturbance model matches the disturbances entering the process.

[11] Detectability means, that all unobservable modes are stable.

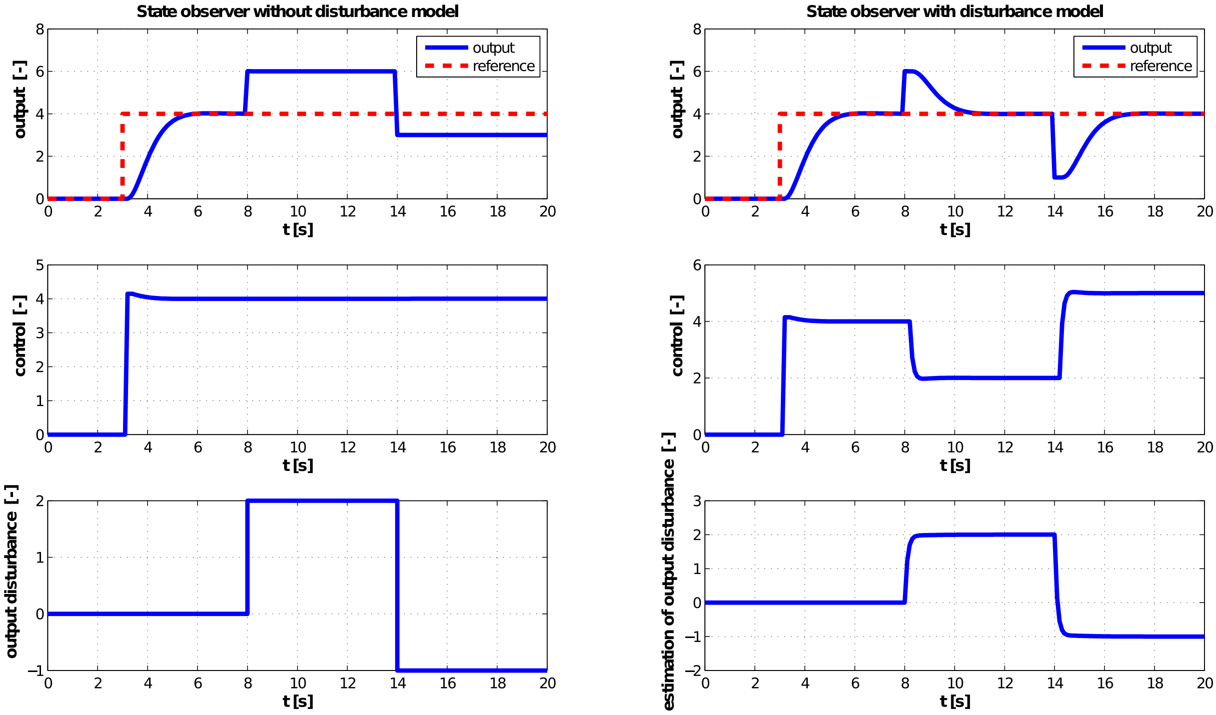

Figure 2.4: Closed-loop response of SISO system to output disturbance.

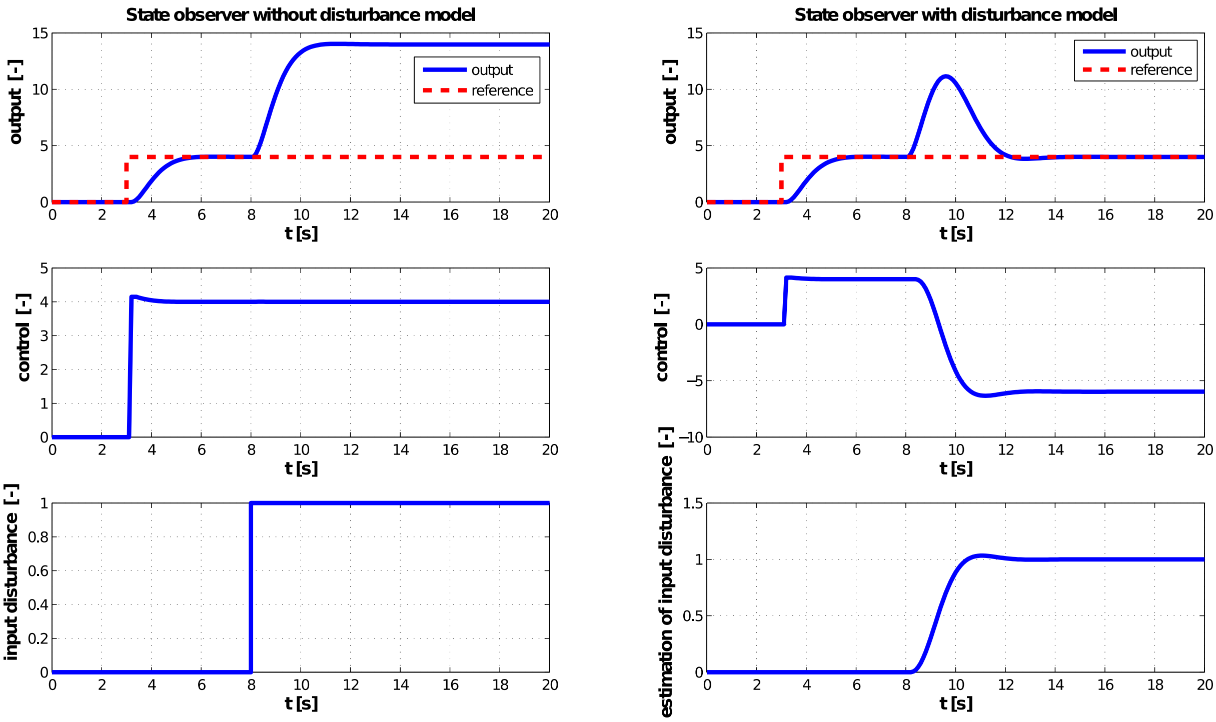

Figure 2.5: Closed-loop response of SISO system to state disturbance.

Chapter 3

Explicit Formulation of Model Predictive Control

The MPC has become the accepted standard for complex constrained multi-variable control problems in the process industries. In MPC, at each sampling time, starting at the current state, an open-loop optimal control problem (2.42a) is solved over a finite horizon. At the next time step, the computation is repeated starting from the new state and over a shifted horizon leading to a moving policy, as describes section 2.2.0.1.
When sampling times become so short that computation times of QP solution can no longer be neglected, specialized algorithms must be used.

One of these are active set methods, which come in two variants, namely primal [26] and dual [4] active set methods. Furthermore, (primal-dual) interior point methods [54] have become a strong competitor to active set methods. They posses relatively constant computational demands and polynomial runtime guarantee can be given for them. However, interior point methods suffer from the drawback that so far no efficient warm starting techniques exist.
In recent work [63] authors found approximate primal barrier method which is used to greatly speed up the computation of the online optimization.

In 2002, Bemporad, Morari, et al published paper [7] in which they found another way to solve the MPC problem. They formulated problem (2.42a) using the *Multi-Parametric Quadratic Programming* and they also found an explicit control law as function of the actual process state, which minimizes the quadratic cost function (2.42a). The greatest benefit of this approach is possibility to move all the computations necessary for implementation of MPC off-line, while preserving all its other characteristics. This largely increases the range of applicability of MPC to problems where standard MPC could not be used. Moreover, such an explicit form of the controller provides additional insight for better understanding of the control policy of MPC (e.g. stability, feasibility, etc.).

A toolbox for Matlab known as Multi Parametric Toolbox [37] is an efficient tool for solving the multi-parametric programmes in connection with model predictive control. The toolbox is freely downloadable from http://control.ee.ethz.ch/∼mpt.

The chapter is organized as follows: in the first section, the multi-parametric quadratic programming will be introduced. Based on it, the Explicit MPC will be formulated in the second section. Finally, in the last section the complexity of explicit control law will be discussed.

3.1 Multi-parametric Quadratic Programming

In this section, multi-parametric quadratic programming (mp-QP) will be briefly described. The goal is to derive an algorithm to express the solution $\boldsymbol{u}^*$ and the minimum value of the quadratic cost function $J(\boldsymbol{u}^*|\boldsymbol{x})$ as an explicit function of the parameters $\boldsymbol{x}$. In particular, can be proven that the solution $\boldsymbol{u}^*$ is a continuous piecewise affine function of $\boldsymbol{x}$ [7],[6].

3.1.1 Optimization Problem

Consider the quadratic cost function with convex constraints:

$$\begin{aligned} \min_{\boldsymbol{u}} J(\boldsymbol{u}|\boldsymbol{x}) &= \min_{\boldsymbol{u}} \left\{ \frac{1}{2} \boldsymbol{u}^T \boldsymbol{H} \, \boldsymbol{u} + \boldsymbol{x}^T \boldsymbol{F} \, \boldsymbol{u} + \frac{1}{2} \boldsymbol{x}^T \boldsymbol{Y} \, \boldsymbol{x} \right\}, \\ \text{s.t.} \qquad \boldsymbol{G} \, \boldsymbol{u} &\leq \boldsymbol{W} + \boldsymbol{E} \, \boldsymbol{x}, \end{aligned} \tag{3.1}$$

where $\boldsymbol{H} = \boldsymbol{H}^T \succ 0$, $\boldsymbol{H} \in \mathcal{R}^{s \times s}$, $\boldsymbol{G} \in \mathcal{R}^{q \times s}$, $\boldsymbol{w} \in \mathcal{R}^q$ and $\boldsymbol{E} \in \mathcal{R}^{q \times n}$.
Before proceeding further, it is useful to transform coordinate:

$$\boldsymbol{z} = \boldsymbol{u} + \boldsymbol{H}^{-1} \boldsymbol{F}^T \boldsymbol{x}, \tag{3.2}$$

which leads to the optimization problem

$$\min_{\boldsymbol{z}} \left\{ \frac{1}{2} \boldsymbol{z}^T \boldsymbol{H} \boldsymbol{z} + \beta \right\} \qquad \text{s.t.} \quad \boldsymbol{G}\boldsymbol{z} \leq \boldsymbol{W} + \boldsymbol{S}\boldsymbol{x}, \tag{3.3}$$

where

$$\boldsymbol{S} = \boldsymbol{E} + \boldsymbol{G}\boldsymbol{H}^{-1}\boldsymbol{F}^T \qquad \beta = \frac{1}{2} \boldsymbol{x}^T \left(\boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{F}^T + \boldsymbol{Y} \right) \boldsymbol{x}, \tag{3.4}$$

and $\boldsymbol{z}$ is the variable to be optimized, $\boldsymbol{x}$ is the parameter, and β is constant which does not influence the optimum of the cost function.

3.1.2 Finding the Cost Function Optimum

Now we can write the Lagrange function for problem (3.3) to minimize it:

$$L(\boldsymbol{z}, \boldsymbol{\lambda}) = \frac{1}{2} \boldsymbol{z}^T \boldsymbol{H} \boldsymbol{z} + \beta + \lambda \left(\boldsymbol{G}\boldsymbol{z} - \boldsymbol{W} - \boldsymbol{S}\boldsymbol{x} \right). \tag{3.5}$$

One can then write the first-order Karush-Kuhn-Tucker (KKT) optimality conditions [12]:

$$\begin{aligned} \frac{\partial L(\boldsymbol{z}, \boldsymbol{\lambda})}{\partial \boldsymbol{z}} = \boldsymbol{H}\boldsymbol{z} + \boldsymbol{G}^T \boldsymbol{\lambda} &= 0 && (3.6a) \\ \frac{\partial L(\boldsymbol{z}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \boldsymbol{G}\boldsymbol{z} - \boldsymbol{W} - \boldsymbol{S}\boldsymbol{x} &= 0 && (3.6b) \\ \lambda_i \left(\boldsymbol{G}^i \boldsymbol{z} - \boldsymbol{W}^i - \boldsymbol{S}^i \boldsymbol{x} \right) &= 0 && (3.6c) \\ \lambda^i &\geq 0 && (3.6d) \end{aligned}$$

with $\lambda \in \mathcal{R}^q$ and superscript $i \in \mathcal{I} = \{1, \cdots, q\}$ denotes the i-th constraint (row of matrices $\boldsymbol{G}$, $\boldsymbol{W}$ and $\boldsymbol{S}$). The constraints can be separated into two disjunctive sets:

$$\begin{aligned} \mathcal{A}(\boldsymbol{x}) &= \left\{ j \in \mathcal{I} \mid \boldsymbol{G}^j \boldsymbol{z}(\boldsymbol{x}) - \boldsymbol{S}^j \boldsymbol{x} = W_j \right\}, && (3.7) \\ \overline{\mathcal{A}}(\boldsymbol{x}) &= \left\{ j \in \mathcal{I} \mid \boldsymbol{G}^j \boldsymbol{z}(\boldsymbol{x}) - \boldsymbol{S}^j \boldsymbol{x} < W_j \right\}, && (3.8) \end{aligned}$$

where $\mathcal{A}(\boldsymbol{x})$ is active constraint ($\lambda_j > 0$) and $\overline{\mathcal{A}}(\boldsymbol{x})$ in inactive constraint ($\lambda_j = 0$).
Now, from (3.6a) one can write: ($\boldsymbol{H} \succ 0$)

$$\boldsymbol{z} = -\boldsymbol{H}^{-1}\boldsymbol{G}^T\boldsymbol{\lambda}, \tag{3.9}$$

and substitute the result into (3.6c) to obtain the complementary slackness condition

$$-\boldsymbol{G}_{\mathcal{A}}\, \boldsymbol{H}^{-1}\boldsymbol{G}_{\mathcal{A}}^T\, \boldsymbol{\lambda}_{\mathcal{A}} - \boldsymbol{W}_{\mathcal{A}} - \boldsymbol{S}_{\mathcal{A}}\, \boldsymbol{x} = 0 \tag{3.10}$$

where $(.)_{\mathcal{A}}$ represents rows of $(.)$ where constraints are active.
Therefore,

$$\boldsymbol{\lambda}^*_{\mathcal{A}} = -\left(\boldsymbol{G}_{\mathcal{A}}\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\right)^{-1}(\boldsymbol{W}_{\mathcal{A}} + \boldsymbol{S}_{\mathcal{A}}\,\boldsymbol{x}), \tag{3.11}$$

where $\left(\boldsymbol{G}_{\mathcal{A}}\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\right)^{-1}$ exists because the rows of $\boldsymbol{G}_{\mathcal{A}}$ are linearly independent. Thus, $\boldsymbol{\lambda}^*_{\mathcal{A}}$ is an affine function of $\boldsymbol{x}$.
Finally, substitute result into (3.9):

$$\boldsymbol{z}^* = \boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\left(\boldsymbol{G}_{\mathcal{A}}\,\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\right)^{-1}(\boldsymbol{W}_{\mathcal{A}} + \boldsymbol{S}_{\mathcal{A}}\,\boldsymbol{x}), \tag{3.12}$$

thus, $\boldsymbol{z}^*$ is also an affine function of $\boldsymbol{x}$. And thanks to $\boldsymbol{H} \succ 0$, the solution is also unique. From substitution (3.2) one can get optimal value of the original optimized variable $\boldsymbol{u}$

$$\boldsymbol{u}^* = \boldsymbol{z}^* - \boldsymbol{H}^{-1}\boldsymbol{F}^T\boldsymbol{x} = \boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\left(\boldsymbol{G}_{\mathcal{A}}\,\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\right)^{-1}(\boldsymbol{W}_{\mathcal{A}} + \boldsymbol{S}_{\mathcal{A}}\,\boldsymbol{x}) - \boldsymbol{H}^{-1}\boldsymbol{F}^T\boldsymbol{x}. \tag{3.13}$$

Equation (3.12) characterizes the solution only locally in the neighborhood of a specific $\boldsymbol{x}_0$, where the same constraints are active and optimal value of optimized variable $\boldsymbol{z}^*$ is also the same. Sets of the active constraints defines so called *critical polyhedral regions* (CR) of the state space. The critical region can be found easily, because variable $\boldsymbol{z}$ from (3.9) must satisfy the constraints in (3.3):

$$\boldsymbol{G}\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\left(\boldsymbol{G}_{\mathcal{A}}\,\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\right)^{-1}(\boldsymbol{W}_{\mathcal{A}} + \boldsymbol{S}_{\mathcal{A}}\,\boldsymbol{x}) \leq \boldsymbol{W} + \boldsymbol{S}\boldsymbol{x}. \tag{3.14}$$

and by (3.6d), the Lagrange multipliers in (3.11) must remain non-negative:

$$\left(\boldsymbol{G}_{\mathcal{A}}\,\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\right)^{-1}(\boldsymbol{W}_{\mathcal{A}} + \boldsymbol{S}_{\mathcal{A}}\,\boldsymbol{x}) \leq 0 \tag{3.15}$$

as we vary $\boldsymbol{x}$. Thus equations (3.14) and (3.15) can be written as polyhedron set:

$$CR : \{\boldsymbol{x} \in \mathcal{X} \mid \boldsymbol{P}\boldsymbol{x} \leq \boldsymbol{k}\,\boldsymbol{x}\} \tag{3.16}$$

with

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{G}\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\left(\boldsymbol{G}_{\mathcal{A}}\,\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\right)^{-1}\boldsymbol{S}_{\mathcal{A}} - \boldsymbol{S} \\ \left(\boldsymbol{G}_{\mathcal{A}}\,\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\right)^{-1}\boldsymbol{S}_{\mathcal{A}} \end{bmatrix} \tag{3.17}$$

$$\boldsymbol{k} = \begin{bmatrix} \boldsymbol{G}\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\left(\boldsymbol{G}_{\mathcal{A}}\,\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\right)^{-1}\boldsymbol{W}_{\mathcal{A}} + \boldsymbol{W} \\ -\left(\boldsymbol{G}_{\mathcal{A}}\,\boldsymbol{H}^{-1}\boldsymbol{G}^T_{\mathcal{A}}\right)^{-1}\boldsymbol{W}_{\mathcal{A}} \end{bmatrix} \tag{3.18}$$

After removing the redundant inequalities from (3.14) and (3.15), we obtain a compact representation of the critical region CR in (3.16) form. Obviously, CR is a polyhedron in

the $\boldsymbol{x}$-space, and represents the largest set of $\boldsymbol{x} \in \mathcal{X}$ such that the combination of active constraints at the minimizer remains unchanged.
Once the critical region CR corresponds to $\boldsymbol{x}_0 \in \mathcal{X}$ has been defined, the rest of the space $CR_{rest} = \mathcal{X} - CR$ has to be explored and new critical regions generated. An effective approach for partitioning the rest of the space was proposed in [16].

From equation (3.13) one can note that the optimal solution of the quadratic program (3.1) is continuous and piece-wise affine function of parameter $\boldsymbol{x}$ in form

$$\boldsymbol{u}^* = f(\boldsymbol{x}) = \boldsymbol{F}^i\boldsymbol{x} + \boldsymbol{g}^i \qquad \text{if} \quad \boldsymbol{x} \in CR_i, \quad i = 1, \dots N_{mpc} \tag{3.19}$$

where the polyhedral set CR_i: $\left\{\boldsymbol{x} \in \mathcal{X} \mid \boldsymbol{P}^i\boldsymbol{x} \leq \boldsymbol{k}^i\right\}$, $i = 1, \dots N_{mpc}$ are a partition of the given set of states $\mathcal{X}$.

3.1.3 Algorithm of the mp-QP

Algorithm of mp-QP can be clearly written as

1. let $\mathcal{X}$ is space of the parameter $\boldsymbol{x}$ and $\mathcal{P}$ is unexplored space of it, $\mathcal{P} \subseteq \mathcal{X}$
2. set $\mathcal{P} = \mathcal{X}$
3. for $\boldsymbol{x}_0 \in \mathcal{P}$ solve the QP problem (3.3) to find $\boldsymbol{z}_0^*$ and $\boldsymbol{\lambda}_0^*$
4. find active constraints $\mathcal{A}$ defining region CR_i, $\mathcal{P} = \mathcal{P} \setminus CR_i$
5. create matrices $\boldsymbol{G}_\mathcal{A}$, $\boldsymbol{W}_\mathcal{A}$ and $\boldsymbol{S}_\mathcal{A}$ by collecting the active constraints
6. generate the new regions of the unexplored space $\mathcal{P}$ and choose some $\boldsymbol{x}_0$ from it
7. repeat from point (3) for all found regions, thus there exist $\boldsymbol{G}_\mathcal{A}$, $\boldsymbol{W}_\mathcal{A}$ and $\boldsymbol{S}_\mathcal{A}$ for each region

3.2 Explicit Model Predictive Control

Consider quadratic cost function of MPC in matrix form (2.43) [1]

$$\begin{aligned} J(\boldsymbol{u}_{k,T_c-1}|\tilde{\boldsymbol{x}}_k) &= \frac{1}{2}\boldsymbol{u}_{k,T_c-1}^T \boldsymbol{H}\, \boldsymbol{u}_{k,T_c-1} + \tilde{\boldsymbol{x}}_k^T \boldsymbol{F}\, \boldsymbol{u}_{k,T_c-1} + \frac{1}{2}\tilde{\boldsymbol{x}}_k^T \boldsymbol{Y}\, \tilde{\boldsymbol{x}}_k, \quad (3.20a) \\ \text{s.t.} \quad \boldsymbol{G}\, \boldsymbol{u}_{k,T_c-1} &\leq \boldsymbol{W} + \boldsymbol{E}\, \tilde{\boldsymbol{x}}_k \quad (3.20b) \end{aligned}$$

Finding the optimal control sequence which minimize this cost function can be represented as mp-QP program (3.1), where $\boldsymbol{u}$ - the vector variable to be optimized corresponds to the wanted optimal control moves $\boldsymbol{u}_{k,T_c-1}$ and $\boldsymbol{x}$ - parameter of the program agrees with the process state vector $\tilde{\boldsymbol{x}}_k$. Thus, we get the optimal control law $\boldsymbol{u}^*_{k,T_c-1}$ as a continuous and piece-wise affine function of the process state vector $\tilde{\boldsymbol{x}}_k$

$$\tilde{\boldsymbol{x}}_k \in CR_i \Rightarrow \boldsymbol{u}^*_{k,T_c-1}(\tilde{\boldsymbol{x}}_k) = \boldsymbol{F}^i \tilde{\boldsymbol{x}}_k + \boldsymbol{q}^i, \quad (3.21)$$

where CR_i is a region which contains the actual process state $\tilde{\boldsymbol{x}}_k$.

Note: We define CR_1 as critical region, where no constrains are active, i.e. strong inequality is valid in (3.20b). From KKT conditions one can write

$$CR_1 : \left\{\tilde{\boldsymbol{x}}_k \in \mathcal{X} \mid -\left(\boldsymbol{E} + \boldsymbol{G}\boldsymbol{H}^{-1}\boldsymbol{F}^T\right)\tilde{\boldsymbol{x}}_k \leq \boldsymbol{W}\right\}. \quad (3.22)$$

For each state $\tilde{\boldsymbol{x}}_k \in CR_1$ the optimal control law is linear function of state value

$$\boldsymbol{u}^*_k = -\boldsymbol{K}_{MPC}\tilde{\boldsymbol{x}}_k = \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{0} & \dots & \boldsymbol{0} \end{bmatrix} \boldsymbol{H}^{-1}\boldsymbol{F}^T. \quad (3.23)$$

□

The solution of mp-QP is control law $\boldsymbol{u}^*_{k,T_c-1}$ over the control horizon T_c. Because we use the receding horizon concept, only the first control move $\boldsymbol{u}^*_k$ is taken into account. The procedure can be shown on the next example.

Example 3.1 (MPC for Double Integrator): Let's consider the discrete time invariant double integrator with state space matrices:

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, & \boldsymbol{B} &= \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \\ \boldsymbol{C} &= \begin{bmatrix} 0 & 1 \end{bmatrix}, & \boldsymbol{D} &= \begin{bmatrix} 0 \end{bmatrix}. \end{aligned} \quad (3.24)$$

[1] Where $\tilde{\boldsymbol{x}}_k$ represents augmented state space vector for example to references, last control moves or/and disturbance states as describes section 2.3.

The goal is to find the explicit MPC control law which minimizes the cost function:

$$\begin{aligned}
\boldsymbol{u^*}_{n,T_c-1} &= arg \min_{\boldsymbol{u}_{n,T_c-1}} J(u_n, \cdots, \boldsymbol{u}_{n+T_c-1}|\boldsymbol{x}_n) = \\
&= \frac{1}{2}\left[\sum_{k=n}^{n+T_p-1} \boldsymbol{x}_k^T \boldsymbol{Q}_x \boldsymbol{x}_k + \sum_{k=n}^{n+T_c-1} u_k^T \boldsymbol{R} u_k\right] && (3.25)\\
\text{s.t.} \quad \boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}u_k, && (3.26)\\
y_k &= \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}u_k && (3.27)
\end{aligned}$$

with additional constraints to output and control:

$$-15 \le y_k \le 15 \qquad -1 \le u_k \le 1 \tag{3.28}$$

The cost function (3.25) is a simpler version of (2.42a). It does not describe the reference tracking problem, but the resulting MPC controller which optimizes it is the state feedback controller which drives all system states towards origin. The reference tracking problem in explicit MPC needs the state augmentation to reference, described in section 2.2.2.1, which in this example [2] precludes the possibility to show easily the resulting control law and critical regions in plain.

For weightening matrices $\boldsymbol{Q}_x = \boldsymbol{I}_2$ and $\boldsymbol{R} = 1$, initial state condition $\boldsymbol{x}_0 = \begin{bmatrix} 4 & -10 \end{bmatrix}^T$ and length of prediction and control horizon $T_p = T_c = 2$ we are looking for optimal control law which drives all system states to origin.

Solution: Using the prediction of the states (2.13) over the prediction horizon one can get the cost function in matrices form:

$$J(\boldsymbol{u}_{k,T_c-1}|\boldsymbol{x}_k) = \frac{1}{2}\boldsymbol{u}_{k,T_c-1}^T \boldsymbol{H}_x \boldsymbol{u}_{k,T_c-1} + \boldsymbol{x}_k^T \boldsymbol{F}_x \boldsymbol{u}_{k,T_c-1} + \frac{1}{2}\boldsymbol{x}_k^T \boldsymbol{Y}_x \boldsymbol{x}_k, \text{s.t. constraints}$$

where $\boldsymbol{H}_x = \acute{\boldsymbol{S}}_{\boldsymbol{x}}{}^T \acute{\boldsymbol{Q}}_x \acute{\boldsymbol{S}}_{\boldsymbol{x}} + \acute{\boldsymbol{R}}$, $\boldsymbol{F}_x = \boldsymbol{V}_{\boldsymbol{x}}{}^T \acute{\boldsymbol{Q}}_x \acute{\boldsymbol{S}}_{\boldsymbol{x}}$ and $\boldsymbol{Y}_x = \boldsymbol{V}_{\boldsymbol{x}}{}^T \acute{\boldsymbol{Q}}_x \boldsymbol{V}_{\boldsymbol{x}}$.
According to section 2.3.2 we construct the constraints matrices in matrix form

$$\boldsymbol{G}\,\boldsymbol{u}_{k,T_c-1} \le \boldsymbol{W} + \boldsymbol{E}\,\boldsymbol{x}, \tag{3.29}$$

✓

The resulting explicit control law is shown on figure 3.1. As one can see, the found control law is defined in 17 state space regions (left image) and value of the control action is a continuous affine function of the state (middle image). Two right images show closed

[2]The double integrator has 2 states.

loop response to a given initial condition. In time 2, the controller respects the control constraints (bottom image). All system states are successfully driven to the original.

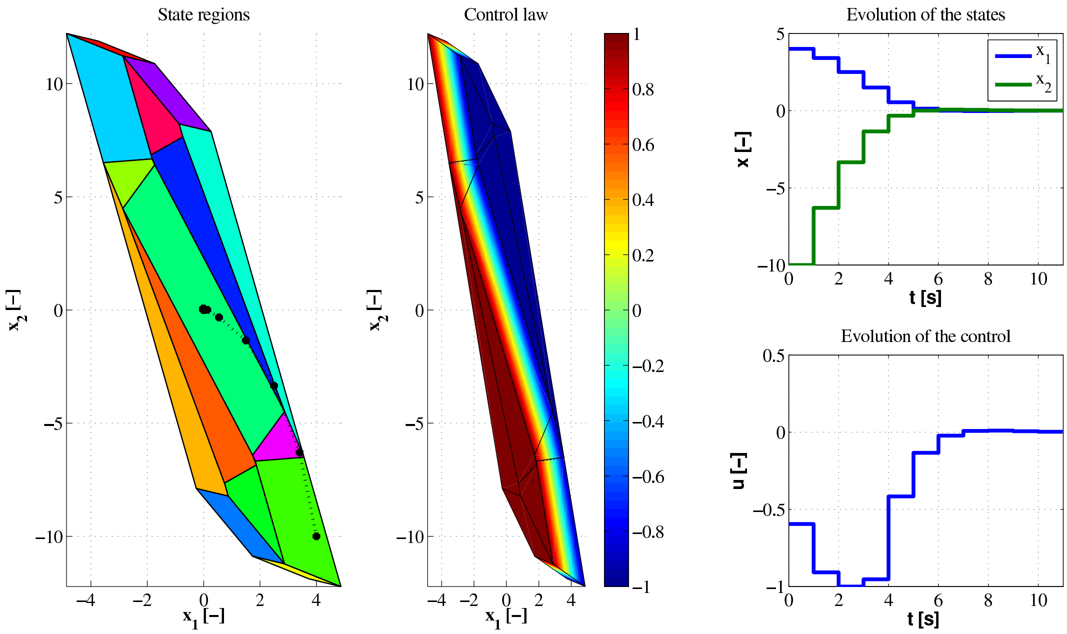

Figure 3.1: Explicit control law for double integrator and closed loop response to initial condition $\boldsymbol{x}_0$.

3.2.1 Complexity Analysis of Explicit Control Law

The main disadvantage of the explicit MPC is its large growth of complexity of a mp-QP solution with increasing number of q constraints [27]. Complexity of the mp-QP solution is reflected by the number N_{mpc} of regions in the mp-QP solution, which influences the computation time of mp-QP (*off-line* complexity) and search time (*on-line* complexity) of the correct region, which includes the actual state $\boldsymbol{x}_k$.

The number N_{mpc} of regions in the mp-QP solution depends on the dimension n of the state vector, and on the number of degrees of freedom $f = mT_c$ and constraints q in the optimization problem (3.3). Note that in complexity analysis, the number of possible combinations of active constraints at the solution of a QP is important. That is at most

$$N_{mpc_{MAX}} = \sum_{k=0}^{q} \binom{q}{k} = 2^q. \tag{3.30}$$

This number represents an upper bound on the number of different linear feedback gains which describes the controller. In practice, far fewer combinations are usually generated. For example lower and upper bound of control variable can not be satisfied simultaneously. Furthermore, the gains for the future control moves $\boldsymbol{u}_{k+1,T_c-1}$ are not relevant for the control law, because of receding horizon concept. Thus, several different combinations of active constraints may lead to the same first control move $\boldsymbol{u}_k^*(\boldsymbol{x})$. It means that several regions can be merged together without any influence to the control quality.
In [7], authors proved that number N_{mpc} of regions in the mp-QP solution remains constant with increasing the number of states n if

$$q_s = \operatorname{rank} \boldsymbol{S}, \quad q_s \leq q, \qquad n > q_s. \tag{3.31}$$

Therefore, the number of partitions N_{mpc} is insensitive to the dimension n of the state $\boldsymbol{x}$ for all $n > q_s$. In particular, the additional parameters that were introduced in section 2.3 to extend MPC to reference tracking, disturbance rejection, soft constraints, do not affect significantly, the complexity of the mp-QP, and hence, the number N_{mpc} of regions in the MPC controller.

The number q of constraints increases with constraints horizon T_{cs} [3] and control horizon T_c, for instance, $q = 2\ f = 2m\ T_c$ in case of control constraints only. Thus, the larger m, p, T_{cs}, T_{cs}, the larger number of constraints q, and therefore N_{mpc}.

Detailed description of complexity analysis of explicit control law can be found in [27]. A note on the complexity of explicit solution versus active set QP algorithm can be found in recent papers [9] and [10].

[3]The constraints horizon T_{cs}, is horizon over which the constraints have to be satisfied. It is usually the same or smaller than the prediction horizon T_p.

Chapter 4

Influence of Model Uncertainty on Constraints Handling in MPC

In the standard formulation of the MPC, the optimization problem is usually formulated as a reference tracking problem, i.e. the system output should follow the given reference signal or stay at some set-point value. This request can be too conservative for some industrial processes, namely in presence of disturbances and model uncertainties therefore the standard MPC algorithm is modified. This extension is called the *Range control* [43], where the controlled variables are enabled to freely vary in specified range, which is defined by the soft constraints. In the last few years many theoretical results were obtained in this field, namely the stability and robustness for linear time-invariant systems [43], and robustness for nonlinear systems [52].

The big drawback of the Range control is that it is computational demanding and therefore it can not be used for on-line control of high-speed systems. The Explicit MPC formulation [7], where the control law is computed off-line is the way how to make on-line control possible even for high-speed systems. But, there is a limitation of resulting control law complexity which depends on the number of constraints. Note that in standard Range control the number of constraints is very high, therefore the complexity of control law is also high. This leads to worse on-line complexity of the control law (increase of the searching time of the appropriate region) and thus prevents the possibility to control the high-speed systems. Therefore, the reduction of the constraints number is needed. Often this is done by choosing the control horizon $T_c = 1$ and small prediction horizon T_p. But this choice leads to poor closed loop performance mainly if disturbances, model uncertainties or nonlinearity are present.

In this chapter the algorithm which reduces the number of soft output constraints with preservation of closed loop performance will be introduced. It considers the soft output constraints only in one step of the prediction of the MPC. It will be shown that this consideration has positive effect on quality of reference tracking problem and it will also improve the controller robustness. This will be shown on example of controlling the linear time invariant discrete MIMO system. The ability of this algorithm will be also shown on controlling the nonlinear model of port injection spark-ignited gasoline engine in chapter 5.

The chapter is organized as follows: in the first section, the main objectives and principles of MIMO feedback control are reviewed. In the following section the new algorithm of output soft constraint handling is introduced. Finally, in last section the influences of the new constraint handling algorithm to controlled system in frequency and time domain are illustrated on example.

4.1 Basic Principles of MIMO Feedback Control

In MIMO case there is a problem to define the system gains, because there exist interactions between inputs or outputs. Generally the response of the MIMO system does not depend only on input frequency but also on input direction.
To describe the input-output behavior of the system we will introduce the transfer function of the system. To state the frequency response of the multi-variable systems we will remind the generalization of Bode plot [15] by introduction the singular values [49]. To express the interactions and ill-conditionality of the system we will introduce the conditional number [28], [59]. We will also show the dependency of model uncertainty to condition number. Then we will introduce the basic objectives of feedback for multi-variable systems.
Note that the last few years brought major development in the mathematical theory of multi-variable linear time invariant feedback systems. The developments include certain generalizations of frequency-domain concept which offers analysis and synthesis tools in the classical SISO tradition [45], such as the classical Bode gain/phase [15], gain margin [65] and the phase margin [64].

4.1.1 Transfer Function

In section 2.1.1 the state-space model was introduced. It describes internal behavior of the system perfectly but it is not suitable for input-output description and for survey a frequency dependency of the system. Rather, the transfer function, which is the mathematical representation of the relation between the inputs and outputs will be introduced. Consider the linear time-invariant (LTI), discrete system in state space form (2.1)

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{4.1a}$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k, \tag{4.1b}$$

Taking the Z-transform [22] of (4.1) with zero initial conditions we find

$$\boldsymbol{y}(z) = \boldsymbol{G}(z)\,\boldsymbol{u}(z) \tag{4.2}$$

where

$$\boldsymbol{G}(z) = \boldsymbol{C}\left(z\boldsymbol{I}_n - \boldsymbol{A}\right)^{-1}\boldsymbol{B} + \boldsymbol{D} \tag{4.3}$$

is referred to as the *system transfer matrix*. The elements $\{g_{ij}(z)\}$ of $\boldsymbol{G}(z)$ are transfer functions expressing the relationship between specific inputs $u_j(z)$ and outputs $y_i(z)$.

4.1.2 Singular Values and Condition Number

The *singular values* [49] of a complex $n \times m$ matrix $\boldsymbol{A}$, denoted $\sigma_i(\boldsymbol{A})$, are the k largest nonnegative square roots of the eigenvalues of $\boldsymbol{A}^H\boldsymbol{A}$ [1] where $k = \min\{n, m\}$, that is

$$\sigma_i(\boldsymbol{A}) = \sqrt{\lambda_i\left(\boldsymbol{A}^H\boldsymbol{A}\right)} \qquad i = 1, 2, \ldots, k, \tag{4.4}$$

where we assume that the σ_i are ordered such that $\sigma_i \geq \sigma_{i+1}$.
We define the maximum $\overline{\sigma}$ and minimum $\underline{\sigma}$ singular values alternatively by

$$\overline{\sigma} = \max_{\boldsymbol{x}\neq 0} \frac{|\boldsymbol{A}\boldsymbol{x}|_2}{|\boldsymbol{x}|_2} = ||\boldsymbol{A}||_2, \tag{4.5}$$

$$\begin{aligned}\underline{\sigma} &= \left[\max_{\boldsymbol{x}\neq 0} \frac{|\boldsymbol{A}^{-1}\boldsymbol{x}|_2}{|\boldsymbol{x}|_2}\right]^{-1} = ||\boldsymbol{A}^{-1}||_2^{-1} \quad \text{if} \quad \boldsymbol{A}^{-1}\text{exists} \\ &= \min_{\boldsymbol{x}\neq 0}\left[\frac{|\boldsymbol{A}^{-1}\boldsymbol{x}|_2}{|\boldsymbol{x}|_2}\right]^{-1} = \min_{\boldsymbol{x}\neq 0} \frac{|\boldsymbol{x}|_2}{|\boldsymbol{A}^{-1}\boldsymbol{x}|_2} = \min_{\boldsymbol{x}\neq 0} \frac{|\boldsymbol{A}\boldsymbol{x}|_2}{|\boldsymbol{x}|_2}\end{aligned} \tag{4.6}$$

[1] Superscript $(.)^H$ is used to denote complex conjugate transpose.

where $|.|$ notes vector 2-norm and $||.||$ notes 2-norm of a function or matrix.
Thus $\overline{\sigma}$ and $\underline{\sigma}$ can be interpreted geometrically as the *least upper bound* and the *greatest lower bound* on the *magnification* of a vector by the matrix function $\boldsymbol{A}$.

A convenient way of representing a matrix that exposes its internal structure is known as the *Singular Value Decomposition* (SVD) [25]. For an $n \times m$ matrix $\boldsymbol{A}$, the SVD of $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H = \sum_{i=1}^{k} \sigma_i(\boldsymbol{A})\boldsymbol{u}_i\, \boldsymbol{v}_i^H, \tag{4.7}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices with column vectors defined by

$$\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n) \qquad \boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m) \tag{4.8}$$

and $\boldsymbol{\Sigma}$ contains a diagonal nonnegative definite matrix $\boldsymbol{\Sigma}_1$ of singular values arranged in descending order as in

$$\boldsymbol{\Sigma} = \begin{bmatrix} \Sigma_1 \\ \boldsymbol{0} \end{bmatrix} \quad n \geq m \tag{4.9}$$

or

$$\boldsymbol{\Sigma} = \begin{bmatrix} \Sigma_1 & \boldsymbol{0} \end{bmatrix} \quad n \leq m \tag{4.10}$$

and

$$\boldsymbol{\Sigma_1} = \mathrm{diag}\left\{\sigma_1, \sigma_2, \ldots, \sigma_k\right\}, \quad k = \min\left\{m, n\right\}, \tag{4.11}$$

where

$$\overline{\sigma} = \sigma_1 \geq \sigma_2 \geq \ldots \sigma_k = \underline{\sigma} \tag{4.12}$$

It can be shown [25] that the columns of $\boldsymbol{V}$ and $\boldsymbol{U}$ are unit eigenvectors of $\boldsymbol{A}^H\boldsymbol{A}$ and $\boldsymbol{A}\boldsymbol{A}^H$ respectively. They are known as the *right* and *left singular vectors* of the matrix $\boldsymbol{A}$. Note also that if $\boldsymbol{A}$ is Hermitian ($\boldsymbol{A} = \boldsymbol{A}^H$) then the singular values and the eigenvalues coincide.
From a system point of view the vector $\boldsymbol{v}_1(\boldsymbol{v}_m)$ in (4.8) corresponds to the control direction with the largest (smallest) amplification. Furthermore $\boldsymbol{u}_1(\boldsymbol{u}_n)$ is the output direction in which the controls are most (least) effective.

We also define the *condition number* [28] of $\boldsymbol{A}$ as

$$\gamma(\boldsymbol{A}) = \overline{\sigma}(\boldsymbol{A})\overline{\sigma}(\boldsymbol{A}^{-1}) = \frac{\overline{\sigma}(\boldsymbol{A})}{\underline{\sigma}(\boldsymbol{A})} \tag{4.13}$$

If the system is unitary then $\gamma(\boldsymbol{G}) = 1$. If the system is ill-conditioned, i.e., $\gamma(\boldsymbol{G})$ is large. The large condition number may be caused by a small value of $\underline{\sigma}$, thus there exists control direction with very small amplification. It can be also indication of a difficult control problem with strong interaction (related to *Relative Gain Array* (RGA) [66]) or it indicates that the system is sensitive to *unstructured input uncertainty*, like badly identified input gains. Thus, if condition number is large, the careful controller design have to be used for good closed loop performance.
On the other hand, small condition number means that system is insensitive to input uncertainty irrespective of the controller, thus robust performance to input uncertainty is guaranteed.

The unstructured input uncertainty belongs to family of parameters uncertainty described in section 2.1.3 and it can be written in so-called multiplicative form:

$$\acute{\boldsymbol{G}}(z) = \acute{\boldsymbol{G}}(e^{j\omega T_s}) = \left[\boldsymbol{I} + \boldsymbol{L}(e^{j\omega T_s})\right] \boldsymbol{G}(e^{j\omega T_s}), \tag{4.14}$$

with

$$\overline{\sigma}(\boldsymbol{L}(e^{j\omega T_s})) < l_m(e^{j\omega T_s}) \qquad 0 \leq \omega \leq \frac{\pi}{T_s}, \tag{4.15}$$

where $\boldsymbol{L}$ is open loop transfer function described in next section, T_s is sampling time of the system and l_m is the bounding functions, which is small ($\ll 1$) at low frequencies and increase to unity and at higher frequencies. It should be noted that representation of uncertainty in (4.14) can be used to include perturbation effects that are not uncertain at all. For example, a nonlinear element may be quite accurately modeled that way. More information about unstructured uncertainties can be found in [15] or in [49].

4.1.3 Feedback for Multi-variable Systems

We will deal with the standard feedback configuration illustrated in figure 4.1. It consists of the interconnected system ($\boldsymbol{G}$) and controller $\boldsymbol{C}$ forced by reference $\boldsymbol{r}$, measurement noise ($\boldsymbol{n}$), and disturbances ($\boldsymbol{d}$). All disturbances are assumed to be reflected to the measured outputs $\boldsymbol{y}$, all signals are in general multi-variable.

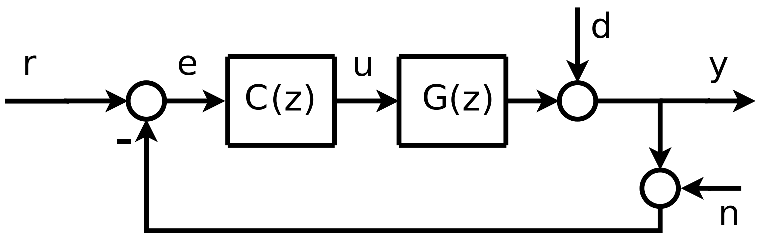

Figure 4.1: Standard feedback configuration.

Then one can write:

$$\boldsymbol{y}(z) = \boldsymbol{T}(z)\boldsymbol{r}(z) + \boldsymbol{S}(z)\boldsymbol{d}(z) - \boldsymbol{T}(z)\boldsymbol{n}(z), \tag{4.16}$$

$$\boldsymbol{e}(z) = \boldsymbol{r}(z) - \boldsymbol{y}(z) = \boldsymbol{S}(z)\boldsymbol{r}(z) - \boldsymbol{S}(z)\boldsymbol{d}(z) + \boldsymbol{T}(z)\boldsymbol{n}(z), \tag{4.17}$$

where we define open loop transfer function $\boldsymbol{L}(z)$, sensitivity $\boldsymbol{S}(z)$, and complementary sensitivity $\boldsymbol{T}(z)$ as

$$\boldsymbol{L}(z) = \boldsymbol{G}(z)\boldsymbol{C}(z) \qquad \boldsymbol{S}(z) = [\boldsymbol{I} + \boldsymbol{L}(z)]^{-1} \qquad \boldsymbol{T}(z) = \boldsymbol{L}(z)\,[\boldsymbol{I} + \boldsymbol{L}(z)]^{-1}. \tag{4.18}$$

Next limitation equation is also valid

$$\boldsymbol{S}(z) + \boldsymbol{T}(z) = \boldsymbol{I}. \tag{4.19}$$

In [15], the author points that the equations (4.16) and (4.17) summarize the fundamental benefits and design objectives inherent in feedback loops. Specifically, (4.17) shows that the tracking error in the presence of reference and disturbances can be made "small" by making the sensitivity $\boldsymbol{S}(z)$ "small".

For SISO systems, the appropriate notion of smallness for the sensitivity is well-understood– we require that the complex scalar $\left[1 + g(e^{j\omega T_s})c(e^{j\omega T_s})\right]^{-1}$ has small magnitude, for all real frequencies ω where the references, disturbances and/or system changes are significant.

The basic idea can be readily extended to MIMO problems through the singular values. Then the corresponding feedback requirements become

$$\overline{\sigma}\left(\boldsymbol{S}(e^{j\omega T_s})\right) = \overline{\sigma}\left([\boldsymbol{I} + \boldsymbol{L}(z)]^{-1}\right) \qquad \text{is small,} \tag{4.20}$$

or conversely

$$ps(e^{j\omega T_s}) \leq \underline{\sigma}\left(\boldsymbol{I} + \boldsymbol{L}(e^{j\omega T_s})\right) \quad \text{for } \omega \leq \omega_0, \tag{4.21}$$

where $ps(e^{j\omega T_s})$ is a positive function and ω_0 specifies the active frequency range. Condition (4.21) on the *return difference* $\boldsymbol{I} + \boldsymbol{L}(e^{j\omega T_s})$ can be interpreted as merely a restatement of the common intuition that large loop gains or tighten loops yield good performance. This follows from the inequalities

$$\underline{\sigma}(\boldsymbol{L}) - 1 \leq \underline{\sigma}(\boldsymbol{I} + \boldsymbol{L}) \leq \underline{\sigma}(\boldsymbol{L}) + 1, \tag{4.22}$$

which show that return difference magnitudes approximate the loop gains, $\underline{\sigma}(\boldsymbol{L})$, whenever these are large compared with unity. Evidently, good multi-variable feedback loop design boils down to achieving high loop gains in the necessary frequency range. But feedback design is not so trivial. It is because the loop gains can not be made arbitrary high over arbitrarily large frequency ranges. They must satisfy certain performance trade-off and design limitations. A major performance trade-off, for example, concerns reference and disturbance error reduction versus sensor noise error reduction. The conflict between these two objectives is evident in (4.17). Large $\underline{\sigma}(\boldsymbol{L}(e^{j\omega T_s}))$ values over a large frequency range make errors due to r and d small. However, they also make errors due to n large because this noise is "passed through" over the same frequency range.

What distinguishes MIMO from SISO loop designs are the functions used to express transfer function "size". The singular values replace absolute values in MIMO case. Thus singular value functions play a design role much like classical Bode plots. For example the $\overline{\sigma}(\boldsymbol{I} + \boldsymbol{L}(e^{j\omega T_s}))$ function is the minimum return difference magnitude of the closed loop system, $\underline{\sigma}(\boldsymbol{L}(e^{j\omega T_s}))$ and $\overline{\sigma}(\boldsymbol{L}(e^{j\omega T_s}))$ are the minimum and the maximum loop gains, and $\overline{\sigma}(\boldsymbol{T}(e^{j\omega T_s}))$ is the maximum closed-loop frequency response. The Condition number $\gamma(\boldsymbol{G})(e^{j\omega T_s})$ shows if the system is ill-conditioned, thus it is strongly dependent to direction of inputs. These can all be plotted as ordinary frequency dependent functions in order to display and analyze the features of a multi-variable design. Such plots will be called *σ-plots*.

4.2 Reduced Output Soft Constrains Handling

In chapter 3 we discussed the Explicit Formulation of MPC, where the optimal control law is introduced as continuous piecewise affine function of state value $\boldsymbol{x}$. Complexity of this law (number of critical regions) depends mainly on the number of constraints. For

usage of the Explicit MPC for high-speed systems, the complexity of the solution (mainly the on-line complexity) must be small, therefore in general there is an effort to reduce the number of constraints to provide the realizable control.
On the other hand, in [62], authors found the *input trajectory parametrization*, where the class of control trajectories with less degrees of freedom is introduced to reduce the dimensions of the optimization problem. Authors of [33] used the goal-oriented *reduced-order* models to reduce the control law complexity.

In this section, the algorithm which provides reduction of control law complexity in case of output soft constraints will be introduced. Although this algorithm can be used for standard formulation of MPC it will be described in conjuction with the Explicit formulation of MPC for better insight.

4.2.1 Algorithm Description

The output constrains are useful in practice when not every of the system outputs are tracking their references and there exists a demand to keep the rest of outputs in some ranges. But adding the hard constrains, to define this ranges, the control law may be infeasible. This solution is not also robust, because presence of disturbances may cause that output would be out of these ranges. In Explicit MPC formulation we can say that there will be no critical region which contains such state value. Therefore, it is reasonable to use the soft constraints defined in section 2.3.4.

If we consider upper and lower limit to control and outputs with p_s the number of outputs controlled in some range, we have q the number of constraints:

$$q = 2\left(mT_c + p_s\, T_p\right). \tag{4.23}$$

Note that control horizon T_c is typically low. For control of high-speed systems it is chosen $T_c = 1$. But prediction horizon T_p is high, typically it is chosen that the output prediction can "see" the steady-state value of the system. Thus, applying the constraints to output, whether hard or soft, over the prediction horizon T_p, the complexity of the explicit control law will increase dramatically. This drawback can be eliminated by using the algorithm described next, where the constraints are considered only in one step of the output prediction. This approach avoids dramatic increase of control law complexity and will be shown that it has a positive influence to feedback features.

In section 2.3.2 we have defined the constraints in form

$$\boldsymbol{G}\,\Delta\boldsymbol{u}_{k,T_c-1} \quad\leq\quad \boldsymbol{W}+\boldsymbol{E}\,\tilde{\boldsymbol{x}}_k, \tag{4.24}$$

with $\boldsymbol{G}\in\mathcal{R}^{q\times(m\cdot T_c+n_S)}$, $\boldsymbol{W}\in\mathcal{R}^{q}$ and $\boldsymbol{E}\in\mathcal{R}^{q\times(\tilde{n}+n_S)}$, n_S is number of slack variables (soft constraints).
Note that each row of matrices $\boldsymbol{G}$, $\boldsymbol{W}$ and $\boldsymbol{E}$ corresponds to one constraint.
Consider a lower and upper soft bounds on output, that is:

$$\underline{y_s}-\underline{\epsilon}\leq y_s\leq\overline{y_s}+\overline{\epsilon}. \tag{4.25}$$

In form of (4.24) we defined these soft constrains over the prediction horizon T_p as (2.74)

$$\begin{bmatrix} \breve{\boldsymbol{S}}_{\boldsymbol{y_s}} & \mathbf{-1} & \mathbf{0} \\ -\breve{\boldsymbol{S}}_{\boldsymbol{y_s}} & \mathbf{0} & \mathbf{-1} \end{bmatrix}\boldsymbol{h}\leq\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}+\begin{bmatrix} -\boldsymbol{V}_{\boldsymbol{y_s}} & \mathbf{1} & \mathbf{0} \\ \boldsymbol{V}_{\boldsymbol{y_s}} & \mathbf{0} & \mathbf{-1} \end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{x}}_k\\ \underline{\overline{\boldsymbol{y}}}\end{bmatrix}, \tag{4.26}$$

where $\breve{\boldsymbol{S}}_{\boldsymbol{y_s}}$ and $\boldsymbol{V}_{\boldsymbol{y_s}}$ are the prediction matrices of the output where the soft constraints are considered.
Consider now not applying constraints (4.25) over whole prediction horizon T_p, but apply them only in one step of the prediction, thus

$$\begin{bmatrix} \breve{\boldsymbol{S}}_{\boldsymbol{y_s}}(T_n) & -1 & 0 \\ -\breve{\boldsymbol{S}}_{\boldsymbol{y_s}}(T_n) & 0 & -1 \end{bmatrix}\boldsymbol{h} \quad\leq\quad \begin{bmatrix}0\\ 0\end{bmatrix}+\begin{bmatrix} -\boldsymbol{V}_{\boldsymbol{y_s}}(T_n) & 1 & 0 \\ \boldsymbol{V}_{\boldsymbol{y_s}}(T_n) & 0 & -1 \end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{x}}_k\\ \underline{\overline{\boldsymbol{y}}}\end{bmatrix}, \tag{4.27}$$

$$1\leq\quad T_n\quad\leq T_p, \tag{4.28}$$

where $\breve{\boldsymbol{S}}_{\boldsymbol{y_s}}(T_n)$ and $\boldsymbol{V}_{\boldsymbol{y_s}}(T_n)$ are the T_n-th row of the prediction matrices $\breve{\boldsymbol{S}}_{\boldsymbol{y_s}}$ and $\boldsymbol{V}_{\boldsymbol{y_s}}$.
Previous construction must be made for all outputs where the soft constraints are assumed. Then we get number of constrains to add

$$n_S=2p_s, \tag{4.29}$$

where p_s is number of outputs controlled in some range.

Note that although the bounds (4.25) are considered in control law design, only n_S constraints were added. Therefore, the increase of control law complexity is not so noticeable as in case where the constrains were assumed over whole prediction horizon and $T_p\,n_S$ constraints have to be added.

4.3 Influence of Reduced Output Soft Constraints to Feedback

In this section the influence to feedback of choice T_n (the step of the prediction), in which the output soft constraints are considered, will be described without any theoretical proofs. These are for another works.

The influence of reduced output soft constraint to feedback loop will be shown on next example. It will be shown that as increaring the parameter T_n, the closed loop system will more and more damp the high frequencies where the system is ill-conditioned and uncertainties or disturbances can arise.

Note that another way how to prevent the excitation of the system on frequencies where it is ill-conditioned is to place the selective filter (e.g. the low pass filter) on the reference input of the system. But this approacch does not defend to the disturbances and model uncertainties which do not pass through the filter.

Example 4.1 (Explicit MPC with reduced output soft constraint): Consider linear time invariant discrete system with 2 control inputs and 2 outputs. The first output will be tracking to a given reference and the second one will be controlled with soft upper bound. Consider also that disturbances enter the system. One disturbance enters into the system additively to the first system state and the second one enters additively to system output, which is tracking to reference. Show the influence of parameter T_n to system in frequency and time domain.
The controlled system is sampled with sample period $T_S = 0.1$ s and its step response is shown in figure 4.2. From that one can deduce, that steady-state gains and settling times in each control direction are very similar. Thus, with increase of frequency of demanded reference, it will be difficult to track the reference with guarantee to meet the constraint on second output. This effect can be also seen in condition number plot, where with increasing the frequency of reference changes, the condition number of the system rapidly increases. From previous sections we can say that this system is sensitive to unstructured input uncertainty and disturbances on high frequencies and the control design should prevent the excitation of the system on these frequencies. It also means that there exists control direction with very small amplification, which leads to non effective control. The σ-plot of this smallest amplification ($\underline{\sigma}(\boldsymbol{G})$) can be seen in top image.

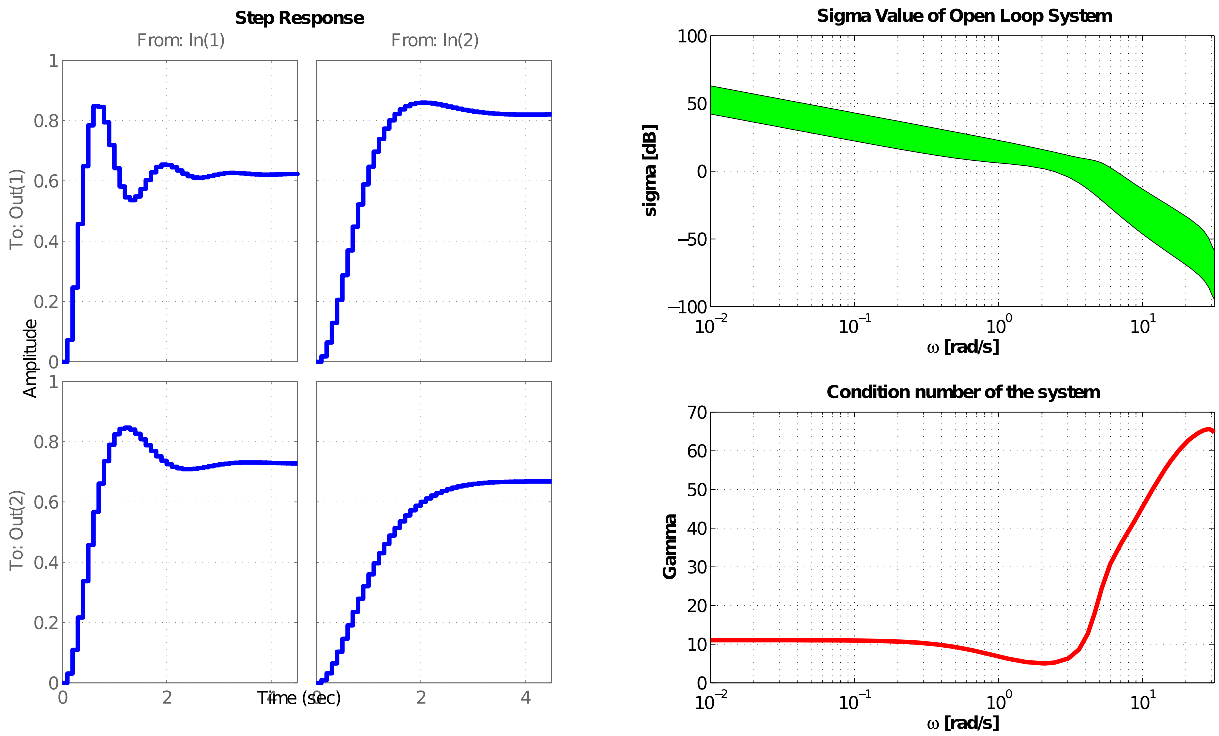

Figure 4.2: Step response, σ-plot of open loop and condition number of the controlled system.

Solution: To get the control law we will use the Explicit Formulation of MPC with control increments with following settings:

$$T_p = 45, \qquad T_c = 1, \qquad Q = 2, \qquad \rho = 10^6, \qquad y_s \leq \overline{y} + \overline{\epsilon}, \tag{4.30}$$

$$\boldsymbol{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} -10 \\ -10 \end{bmatrix} \leq \boldsymbol{u} \leq \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad \boldsymbol{Z} = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{4.31}$$

On the second output y_s the described algorithm will be used.

We will also use the Unknown Input Observer method described in 2.3.6 to ensure the offset free tracking by modeling 1 state and 1 output disturbance, i.e.,

$$\boldsymbol{G}_d = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^T \qquad \boldsymbol{G}_p = \begin{bmatrix} 1 & 0 \end{bmatrix}^T, \tag{4.32}$$

For estimating the system state we will use the Kalman filter augmented to disturbances model.

According to section 2.3.3.2 we construct the matrices of control constraints and using equation (4.27) we construct matrices of output soft constraint for given T_n. Then we assemble the matrices of cost function (2.72) and we compute the explicit control law for a few parameters T_n. Next, the influence of T_n to frequency and time response will be shown.

We will also construct the controller with settings (4.30) and with constraint horizon $T_{cs} = 10$ for comparing. Note that the controller created by using the reduced output soft constraints handling algorithm has only 18 regions and the controller with $T_{cs} = 10$ has 311 regions, i.e., it is much more complex. ✓

Note: In resulting explicit control law we define CR_2 as critical region, where only the soft output constrain is active. In this region, the control law is changed by the choice of T_n, the step of the prediction, in which the output soft constraints are considered. Note that the control law will be also changed in other regions where the soft output constrain is active, for example in regions where the combination of control and soft output constraints is active.
Note also, that control law in region CR_1 will not be influenced by the changing the parameter T_n, because there are no active constraints in this region. □

4.3.1 Frequency Domain

Explicit MPC control law is defined in regions, where each of them has own gain $\boldsymbol{F}^i$ in equation (3.21) of the optimal control. Thus, frequency response of the system will depend to region where the state value $\tilde{\boldsymbol{x}}_k$ lies.

From Note 1 we know that the control law in region CR_1 is independent to any constraints, therefore even to choice of parameter T_n (step of prediction where the soft constraints are considered). Thus frequency response of the system with state in region CR_1 will be independent to choice of parameter T_n. However, the frequency response of the system with state in region CR_2 with changing T_n will be more interesting. Note that the influence of the parameter T_n to response of the system in other regions, where the considered soft constraint is active, is very similar as in region CR_2.

The optimal control law (3.21) in i-th region can be rewritten as

$$\Delta \boldsymbol{u}_k^* = \boldsymbol{F}^i \tilde{\boldsymbol{x}}_k + \boldsymbol{G}^i \tag{4.33}$$

as

$$\Delta \boldsymbol{u}_k^* = \boldsymbol{K_x x}_k + \boldsymbol{K_u u}_{k-1} + \boldsymbol{K_r r}_k + \boldsymbol{K_y} \overline{y}_k, \tag{4.34}$$

where $\boldsymbol{K_x}$, $\boldsymbol{K_u}$ and $\boldsymbol{K_r}$ are gains obtained from controller gain $\boldsymbol{F}^i$. They are pertaining to system state $\boldsymbol{x}_k$, last control moves $\boldsymbol{u}_{k-1}$ and to reference state $\boldsymbol{r}_k$ from equation (2.49). Gain $\boldsymbol{K_y}$ belongs to state, which arises when soft constraint to output was added. Note that in this case the state vector $\boldsymbol{x}_k$ already includes the model of disturbances.

4.3.1.1 Region CR_1

There is no active constraints in region CR_1, thus gain $\boldsymbol{K_y}$ in (4.34) will be zero. Then we can define the sensitivity and complementary sensitivity functions as follows.

4.3.1.1.1 Sensitivity Substitution (4.34) in equation (2.49) one get system with r reference as input and e reference tracking error as output

$$\begin{aligned} \begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{u}_k \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} + \boldsymbol{BK_x} & \boldsymbol{B} + \boldsymbol{BK_u} \\ \boldsymbol{K_x} & \boldsymbol{I}_m + \boldsymbol{K_u} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_{k-1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{BK_r} \\ \boldsymbol{K_r} \end{bmatrix} r_k \\ e_k &= \begin{bmatrix} \boldsymbol{ZC} + \boldsymbol{ZDK_x} & \boldsymbol{ZD}(\boldsymbol{I}_m + \boldsymbol{K_u}) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_{k-1} \end{bmatrix} + \begin{bmatrix} -1 \end{bmatrix} r_k. \end{aligned} \tag{4.35}$$

This system represents the sensitivity function and using the equation (4.3) one can get SISO transfer function. The Bode plot of sensitivity is shown in figure 4.3. One can see that it increases with frequency and it has a maximum on 6 rad/s on which $\boldsymbol{S}$ is greater then 0 dB, i.e. the closed loop system will amplify the disturbances and uncertainties.

4.3.1.1.2 Complementary Sensitivity Substitution (4.34) in equation (2.49) and (2.50) one can gets system with r reference as input and z as output which is tracking to the reference.

$$\begin{aligned} \begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{u}_k \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} + \boldsymbol{BK_x} & \boldsymbol{B} + \boldsymbol{BK_u} \\ \boldsymbol{K_x} & \boldsymbol{I}_m + \boldsymbol{K_u} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_{k-1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{BK_r} \\ \boldsymbol{K_r} \end{bmatrix} r_k \\ z &= \begin{bmatrix} \boldsymbol{ZC} + \boldsymbol{ZDK_x} & \boldsymbol{ZD}(\boldsymbol{I}_m + \boldsymbol{K_u}) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_{k-1} \end{bmatrix} \end{aligned} \tag{4.36}$$

This system represents the complementary sensitivity function and using the equation (4.3) one can get SISO transfer function. The Bode plot of complementary sensitivity is shown in figure 4.3. As one can see, the condition $\boldsymbol{T} \approx 1$ is guaranteed in wide band and high frequencies corresponding to condition number's growth are correctly damped to prevent the measurement noise influence.

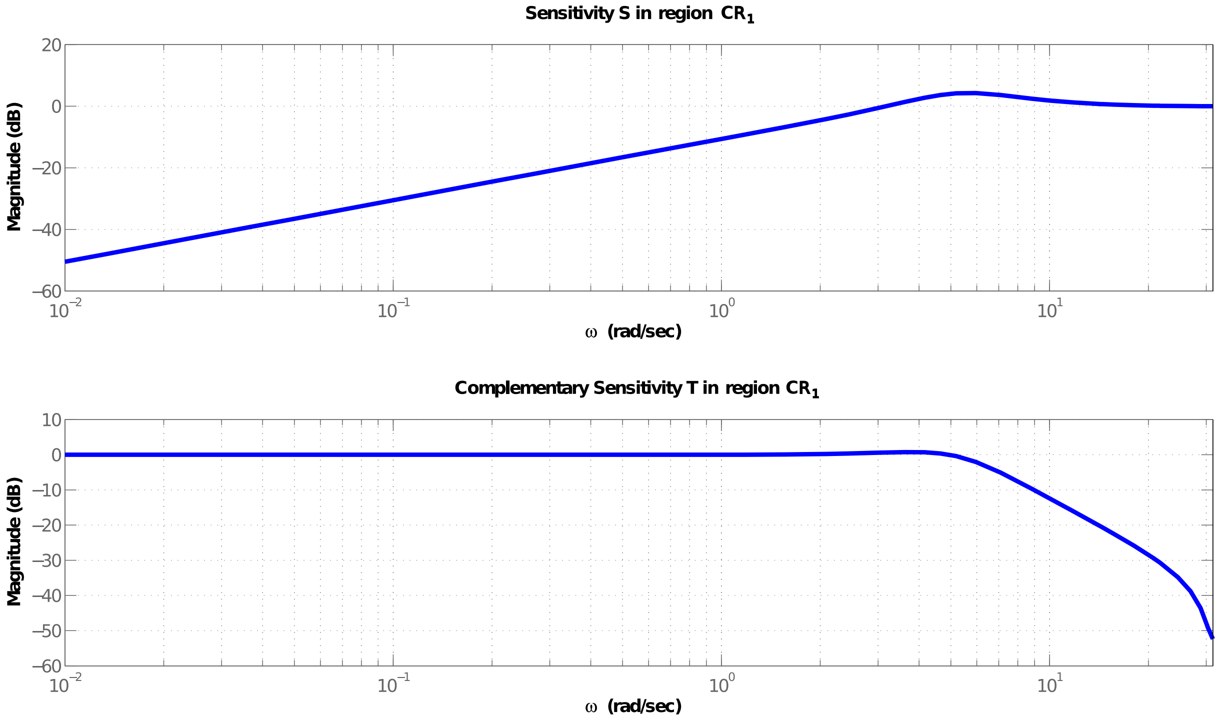

Figure 4.3: Sensitivity and complementary sensitivity in region CR_1.

4.3.1.2 Region CR_2

In region CR_2 the output soft constraint is active, thus gain $\boldsymbol{K_y}$ will not be zero. Also the slack variable $\bar{\epsilon}$ will not be zero. One can actually write:

$$\bar{\epsilon}_k = \boldsymbol{K}_{\boldsymbol{x}_\epsilon} \boldsymbol{x}_k + \boldsymbol{K}_{\boldsymbol{u}_\epsilon} \boldsymbol{u}_{k-1} + \boldsymbol{K}_{\boldsymbol{r}_\epsilon} \boldsymbol{r}_k + \boldsymbol{K}_{\boldsymbol{y}_\epsilon} \overline{y}_k, \tag{4.37}$$

where $\boldsymbol{K}_{\boldsymbol{x}_\epsilon}$, $\boldsymbol{K}_{\boldsymbol{u}_\epsilon}$ and $\boldsymbol{K}_{\boldsymbol{y}_\epsilon}$ are gains obtained from controller gain $\boldsymbol{F}^2$. They are pertaining to system state $\boldsymbol{x}_k$, last control moves $\boldsymbol{u}_{k-1}$ and to reference state $\boldsymbol{r}_k$ from equation (2.49).

4.3.1.2.1 Sensitivity Substitution (4.34) and (4.37) in equation (2.49) one get system with two inputs: r reference and $\overline{y}$ upper bound of second output, and with two outputs: e reference tracking error of first output and $\overline{\epsilon}$ violation of soft upper bound of second output.

$$\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{u}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_{\boldsymbol{x}} & \boldsymbol{B}+\boldsymbol{B}\boldsymbol{K}_{\boldsymbol{u}} \\ \boldsymbol{K}_{\boldsymbol{x}} & \boldsymbol{I}_m+\boldsymbol{K}_{\boldsymbol{u}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_{k-1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}\boldsymbol{K}_{\boldsymbol{r}} & \boldsymbol{B}\boldsymbol{K}_{\boldsymbol{y}} \\ \boldsymbol{K}_{\boldsymbol{r}} & \boldsymbol{K}_{\boldsymbol{y}} \end{bmatrix} \begin{bmatrix} r_k \\ \overline{y}_k \end{bmatrix}$$

$$\begin{bmatrix} e_k \\ \overline{\epsilon}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{Z}\boldsymbol{C}+\boldsymbol{Z}\boldsymbol{D}\boldsymbol{K}_{\boldsymbol{x}} & \boldsymbol{Z}\boldsymbol{D}\left(\boldsymbol{I}_m+\boldsymbol{K}_{\boldsymbol{u}}\right) \\ \boldsymbol{K}_{\boldsymbol{x}_\epsilon} & \boldsymbol{K}_{\boldsymbol{u}_\epsilon} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_{k-1} \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ \boldsymbol{K}_{\boldsymbol{r}_\epsilon} & \boldsymbol{K}_{\boldsymbol{y}_\epsilon} \end{bmatrix} \begin{bmatrix} r_k \\ \overline{y}_k \end{bmatrix}. \tag{4.38}$$

Note that system above represents the sensitivity function. The soft constraint of output y_s were reformulated as "tracking" problem with reference $\overline{y_s}$ and with tracking error $\overline{\epsilon}_k = y_{s_k} - \overline{y_s}_k$. Note also that resulting transfer function is MIMO. Thus, standard Bode plot is unmeaning and the σ-plots for changing parameter T_n are used in figure 4.4. As one can see, for $T_n = 1$ the maximum sigma value of sensitivity $\overline{\sigma}(\boldsymbol{S})$ is greater then 0 dB for high frequencies, where the system is ill-conditioned. Thus all uncertainties and disturbances are amplified.

By increasing the T_n, the sensitivity descends on high frequencies, so the control loop becomes robust. This shows the images with $T_n = 5, 10$.
On the other hand, by further increase of T_n, the peak of the sensitivity increases on middle frequencies. This situation can be seen for example in image for $T_n = 20$. Note when the sensitivity growth, the error tracking also growth. Therefore some methods, which eliminate the tracking error on middle frequencies are needed. One of the these - Unknown Input Observer was described in section 2.3.6. Note that eliminating the tracking error on middle frequencies is simpler than do it on high frequencies, where the system is ill-conditioned.

Comparing the sensitivity function with $T_n = 1$ with sensitivity function in figure 4.6 of the system with controller which is using constraints horizon $T_{cs} = 10$, we can deduce that they are the same. Thus, the reduced output soft constraint algorithm provides much less complex controller (18 regions) with the same frequency response as more complex controller (311 regions) in region CR_2.

4.3.1.2.2 Complementary Sensitivity Substitution (4.34) and (4.37) in equation (2.49) one get system with two inputs: r the reference and $\overline{y}$ the upper bound of second output. And with two outputs: output z which tracking the reference and output y_s

which is controlled below the upper soft constraints.

$$\begin{aligned}\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{u}_k \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}_{\boldsymbol{x}} & \boldsymbol{B}+\boldsymbol{B}\boldsymbol{K}_{\boldsymbol{u}} \\ \boldsymbol{K}_{\boldsymbol{x}} & \boldsymbol{I}_m+\boldsymbol{K}_{\boldsymbol{u}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_{k-1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}\boldsymbol{K}_{\boldsymbol{r}} & \boldsymbol{B}\boldsymbol{K}_{\boldsymbol{y}} \\ \boldsymbol{K}_{\boldsymbol{r}} & \boldsymbol{K}_{\boldsymbol{y}} \end{bmatrix} \begin{bmatrix} r_k \\ \overline{y}_k \end{bmatrix} \\ \boldsymbol{y} &= \begin{bmatrix} \boldsymbol{C}+\boldsymbol{D}\boldsymbol{K}_{\boldsymbol{x}} & \boldsymbol{D}\left(\boldsymbol{I}_m+\boldsymbol{K}_{\boldsymbol{u}}\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_{k-1} \end{bmatrix}. \end{aligned} \tag{4.39}$$

Note that

$$\boldsymbol{y} = \begin{bmatrix} z & y_s \end{bmatrix}^T. \tag{4.40}$$

The system above represents the complementary sensitivity function $\boldsymbol{T}$. The soft constraint of second output were again reformulated as the "tracking" problem as in sensitivity case. Note that main objective for control design in MIMO case is to keep $\overline{\sigma}(\boldsymbol{T})$ very close to 1 in wide band from low to middle frequencies and forcefully damp the high frequencies, where the system uncertainties and measure noise take effect. Another objective is to narrow the band between the maximum amplification of $\overline{\sigma}(\boldsymbol{T})$ and minimum amplification $\underline{\sigma}(\boldsymbol{T})$, to eliminate the influence of direction of the reference vector.

The σ-plots of complementary sensitivity function for system in region CR_2 for a few parameters T_n are shown in figure 4.5. As one can see, for $T_n = 1$ the frequency band on which the $\overline{\sigma}(\boldsymbol{T})$ is closed to 1 is narrow. Also the high frequencies are not damped enough, thus all the system uncertainties together with measure noise will influence the quality of control (match with condition number in figure 4.2). The band between the maximum and minimum amplification is wide too, therefore the control would be difficult. By increasing the parameter T_n, the band between the maximum and minimum amplification is narrowing at the first look (compare for example the σ-plots for $T_n = 1$ and $T_n = 5$). The frequency band, where the $\overline{\sigma}(\boldsymbol{T})$ is closed to 1 is extended, although by further increase of T_n the frequency band is too wide, so that it intervenes to frequencies where the system is ill-conditioned. Note that this drawback is not so distinct compared to choice of low T_n, where these frequencies are not damped enough.
Comparing the complementary sensitivity function with $T_n = 1$ with complementary sensitivity function in figure 4.6 of the system with controller which using constraints horizon $T_{cs} = 10$, we can deduce that they are very similar.

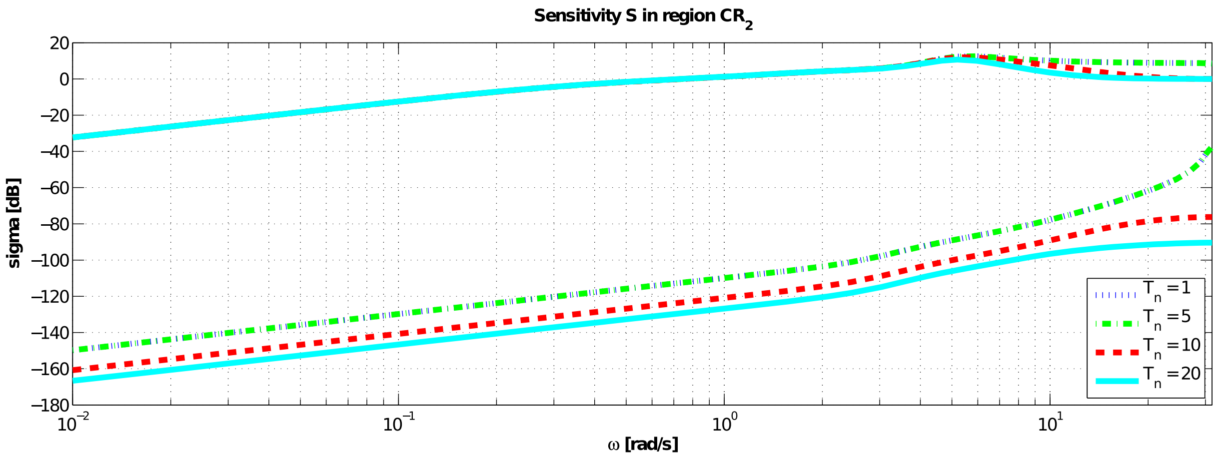

Figure 4.4: The σ-plots of sensitivity in region CR_2 for $T_n = 1, 5, 10, 20$.

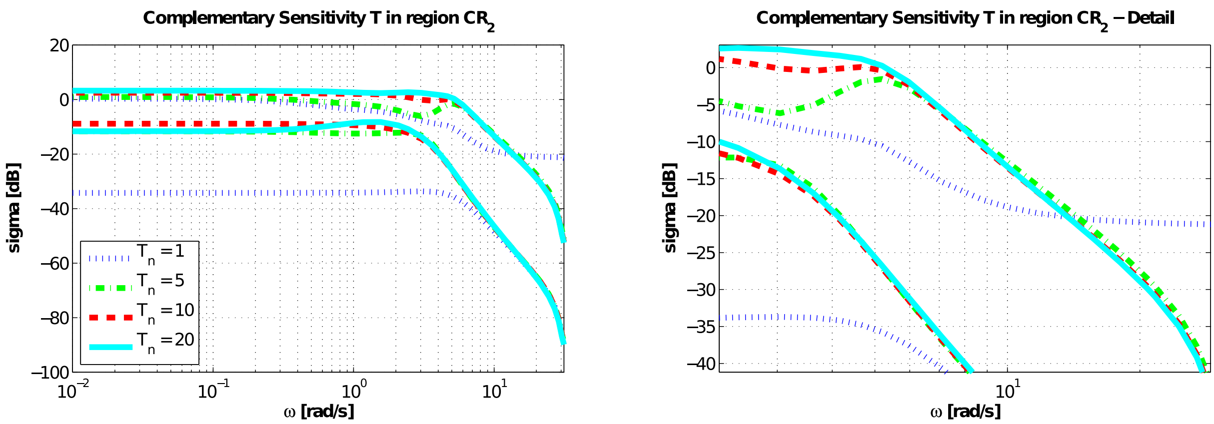

Figure 4.5: The σ-plots of complementary sensitivity in region CR_2 for $T_n = 1, 5, 10, 20$.

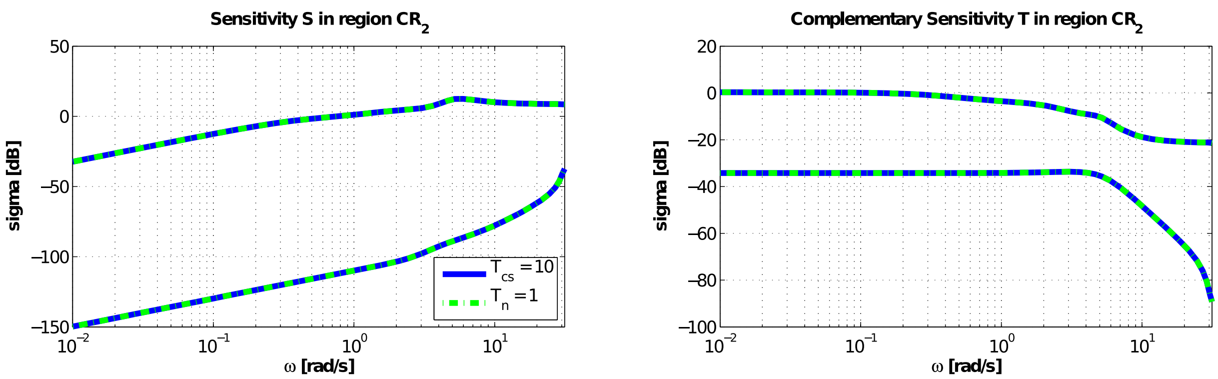

Figure 4.6: Comparing of the σ-plots of sensitivity and complementary sensitivity in region CR_2 for controller with $T_{cs} = 10$ and with $T_n = 1$.

4.3.2 Time Domain

In the previous section, we showed that by increasing the parameter T_n, the closed loop more and more damps the high frequencies of reference changes, thus the system is not excited in band of frequencies where it is ill-conditioned.

This damping of high frequency will also take effect in time domain. It will decelerate the system dynamics and the controller will control the system more carefully. The influence of parameter T_n will be shown on nominal model first, then we will focus on case where the disturbances and model uncertainties will be present.

4.3.2.1 Nominal Model

The closed loop responses of nominal model is shown in figure 4.7. Here one can see that, the controller with $T_n = 1$ is too aggressive. Once the soft constraint of second output is violated, the controller produces the maximum available control action to solve this violation. This leads to oscillations on the both outputs (at time $10 - 16$ s), therefore the offset-free tracking is not possible thus the choice of $T_n = 1$ is not suitable.

By increasing the parameter T_n (for example to $T_n = 20$) the dynamics of closed loop is damped, so the controller produces more conservative control action which leads to offset-free tracking. Note that this is because the controller takes the violation of the output soft constrain into account after 20 steps of the prediction, where the system outputs are almost in steady state (we assume stable system) and probably near the soft constraint, moreover it does not consider the violation of the output soft constraint over the prediction up to step T_n. The controller then implements the strategy: "Care only what you can fix, other ignore". It means that the controller does not produce large control actions to eliminate the soft constraint violation immediately in very close future, but rather the controller tries to eliminate the violation in distant future.

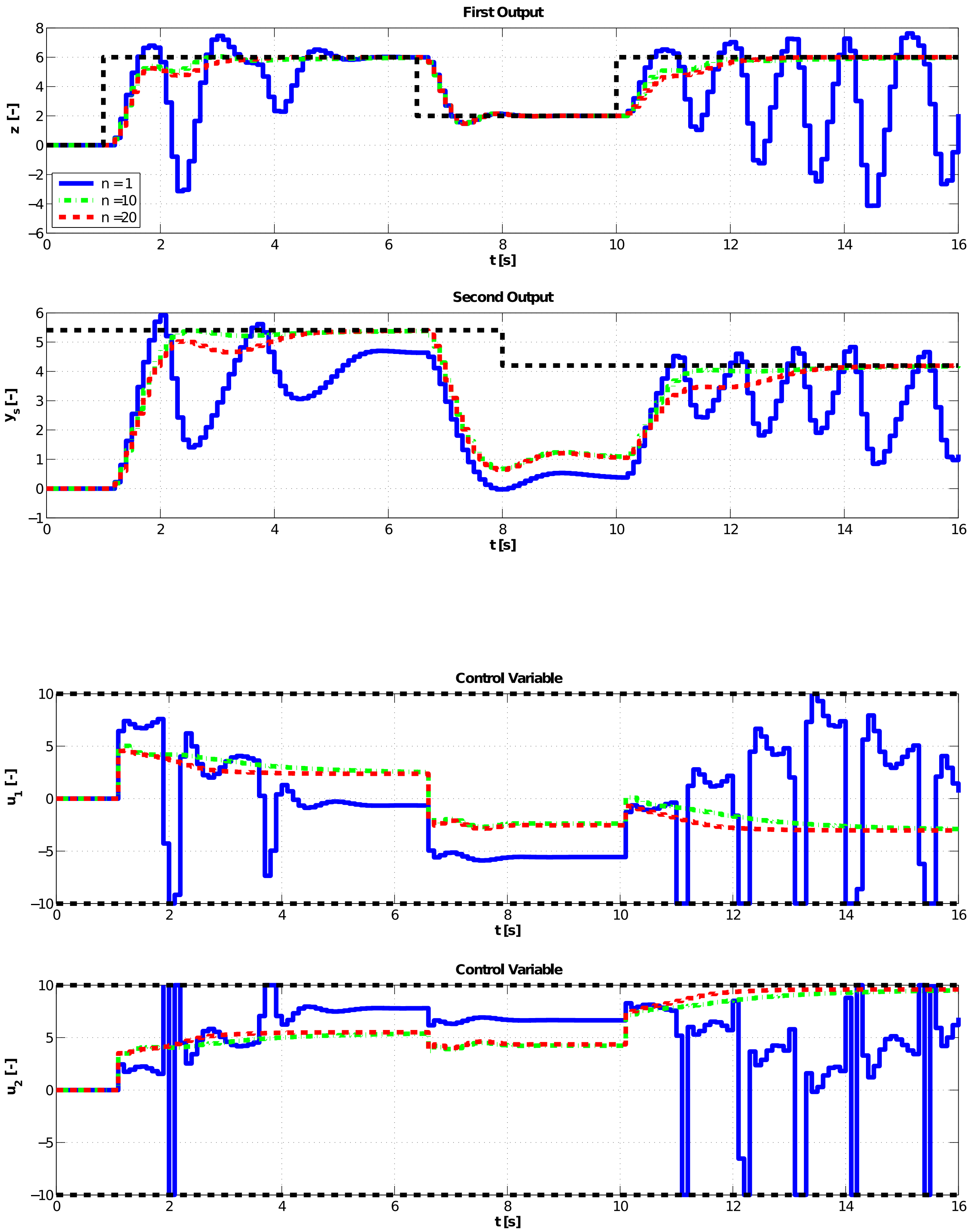

Figure 4.7: Closed loop responses of nominal model for controller with $T_n = 1, 10, 20$

4.3.2.2 Uncertain Model

In previous section the influence of T_n to control loop with nominal model were introduced. It was shown that by increasing the parameter T_n, the controller produces more conservative control actions. Now we will consider the presence of the input uncertainty together with limited disturbances. We will show that by increasing the parameter T_n the closed loop system will have better performance. Moreover, we will show that quality of control will be better then in case of usage standard MPC controller which considers the output soft constraint over the constraint horizon T_{cs}.

In next simulations we will assume an uncertain model with bad identified input gains, so that the first input gain of the real system is by about 30 % greater and the second input gain is by about 50% greater then the model which MPC uses for prediction and which observer uses. The state and output disturbances which enter to system are shown in figure 4.10.

The system outputs controlled by controllers with $T_{cs} = 10$ or with different parameter T_n are shown in figure 4.8. One can see that closed loop response for controller with $T_{cs} = 10$ or with $T_n = 1$ are very similar. This is because both of them are sensitive to input uncertainty as it was discussed in section 4.3.1. Thus it is no wonder that quality of control is really poor in both of these cases.

On the other hand, by increasing the parameter T_n (for example to $T_n = 20$), the quality of control is much better. One can see that although the soft constraint of the second output is violated (time 10-15s) it does not influence the control quality of the first output which is tracking a given reference.
Note that the disturbances are successfully rejected independently of choice of T_n or T_{cs}.

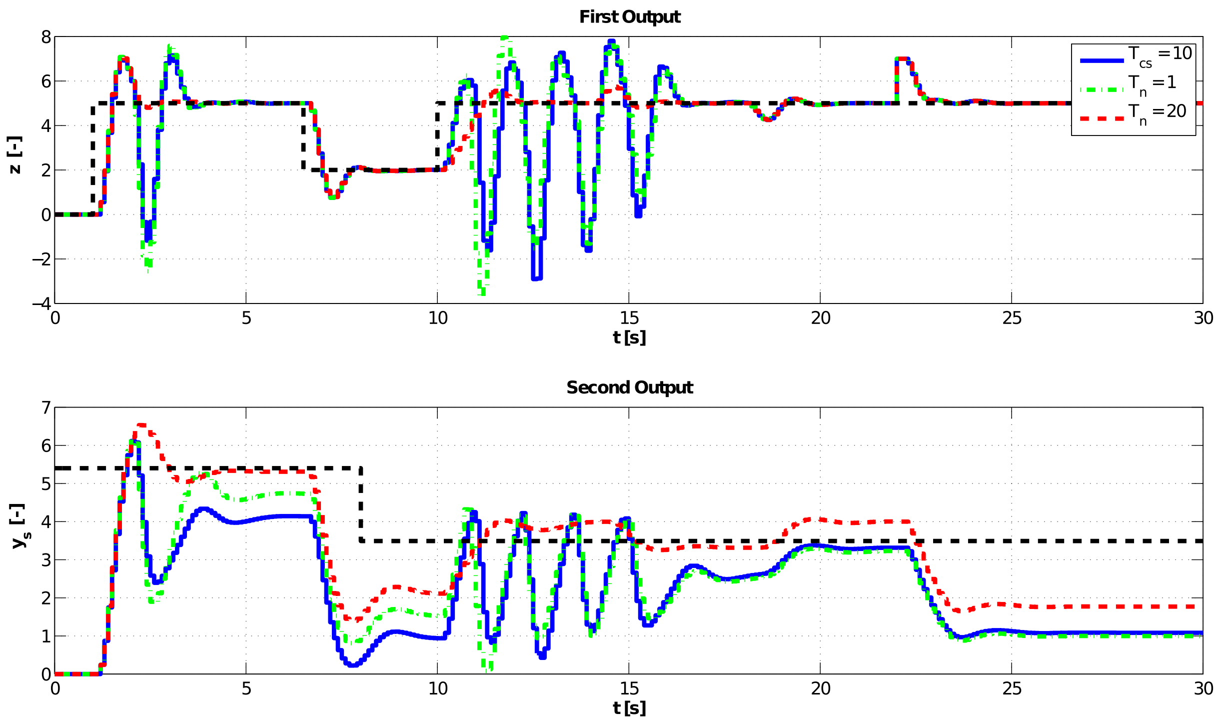

Figure 4.8: Closed loop responses (outputs) for uncertain system model controlled by controller with $T_{cs} = 10$ and $T_n = 1, 20$

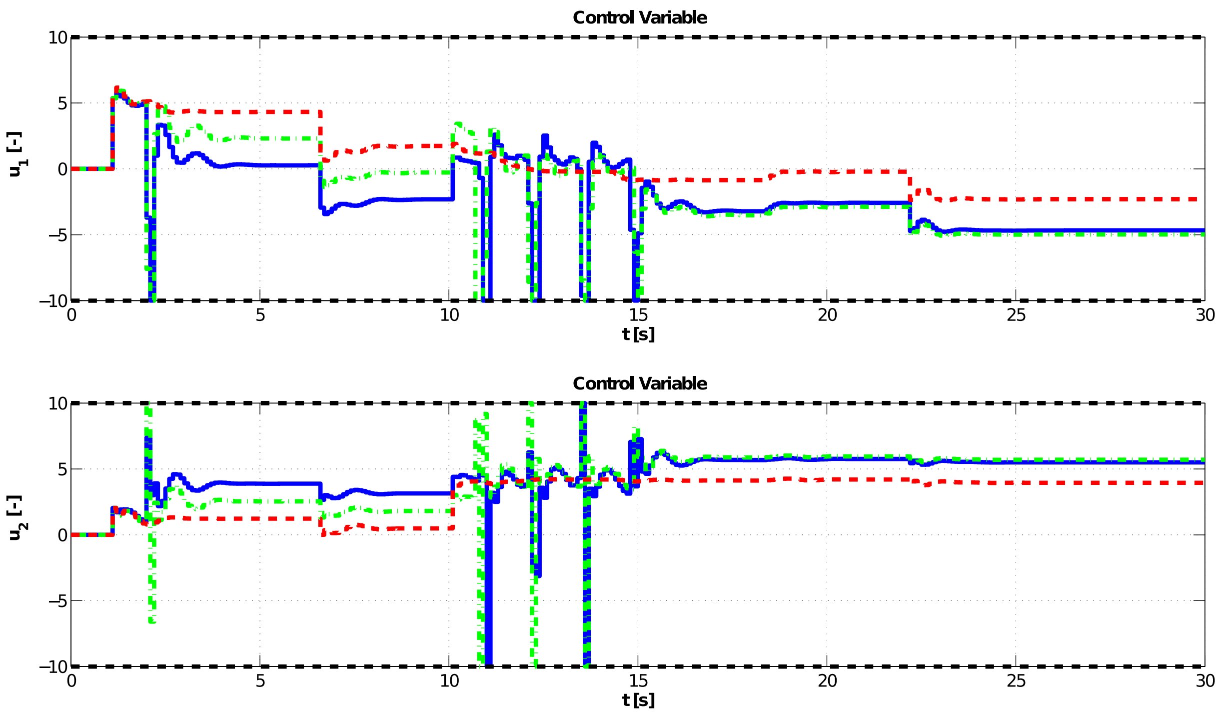

Figure 4.9: Closed loop responses (control variables) for uncertain system model controlled by controller with $T_{cs} = 10$ and $T_n = 1, 20$

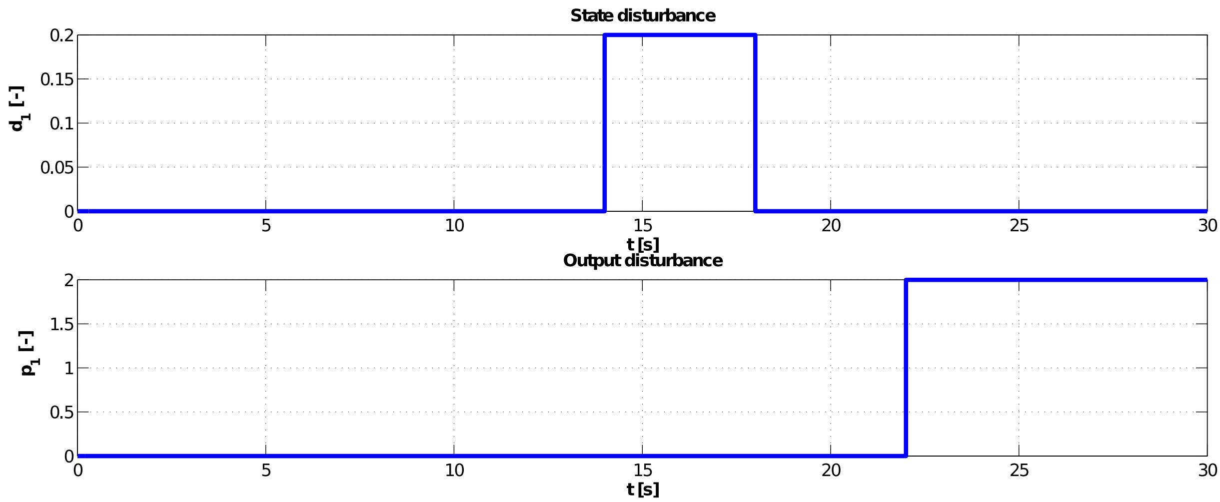

Figure 4.10: Disturbances entering the system.

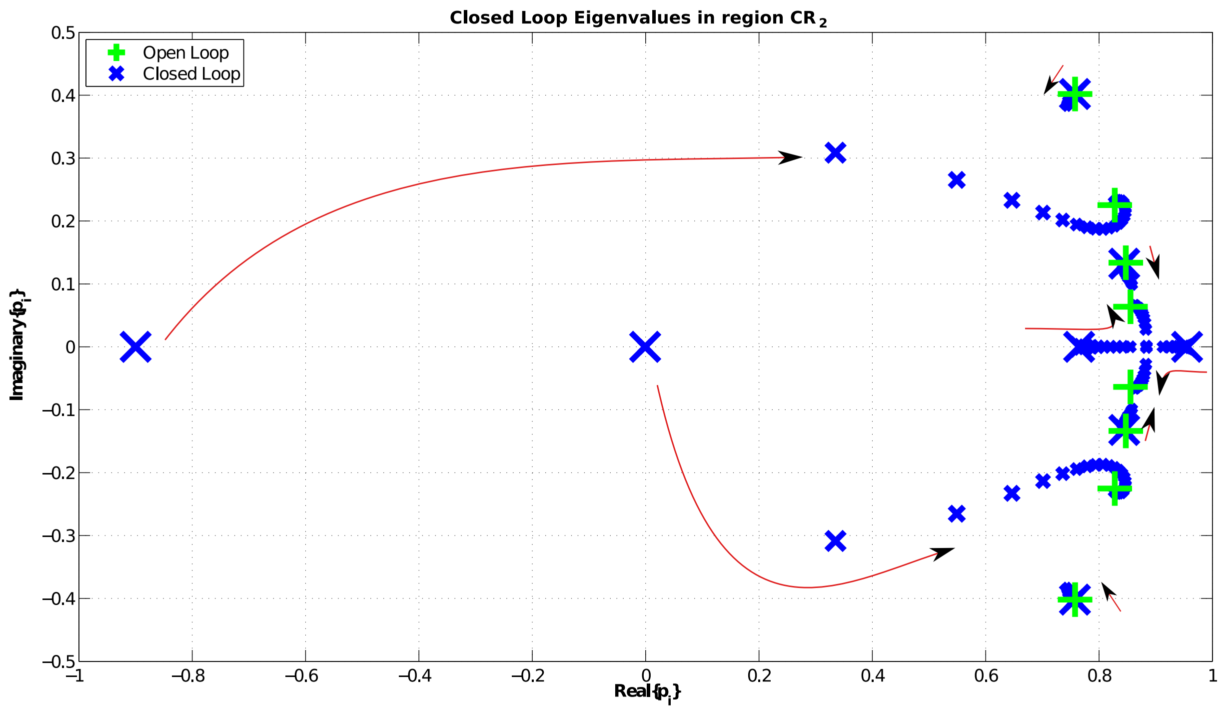

Figure 4.11: The eigenvalues trajectories of discrete closed loop system in region CR_2 for $T_n = 1, 2, \ldots, 25$.

4.3.3 Positions of Closed Loop Eigenvalues in region CR_2

We can also watch the influence of choice of T_n to the trajectories of eigenvalues of the closed loop system with state in region CR_2. The trajectories for our example are shown in figure 4.11. The arrows show directions of eigenvalues trajectories for increasing T_n.

Note that the changing T_n from 1 to 2 causes that one fast oscillating eigenvalue together with another one will change to slower complex pair. By further increasing of T_n the complex pair are more and more moved close to 1, thus the system is more and more like low pass filter. It can be matched with complementary sensitivity σ-plots in 4.5, where by increasing the parameter T_n the high frequencies are more and more damped. As one can see, the complex eigenvalues are noticeable moved but the real ones are moved much less. Thus the influence of choice of T_n to positions of eigenvalueas is not straightforward and deeper research of eigenvalues behavior is needed. This research is out of scope of this work thus it is left for future work.

Chapter 5

Practical Usage of Explicit Model Predictive Control

In this chapter the complete design of Explicit MPC which will use the output soft constraints handling algorithm described in previous chapter for nonlinear model of port injection spark-ignited gasoline engine will be done. Note that the engine is the high-speed non-linear uncertain system [29], thus for control it, the robust and very fast controller is needed.
An emission reduction and fuel economy are often contrasting goals therefore the engine control unit essentially contains a very complex collection of simple loops, mostly with PID controller, with a huge amount of manually or semi-manually calibrated tables [20]. Thus it it effort to use the MPC to control the engines to reduce the complexity of the control design [19],[51].

The chapter is organized as follows: in the first section the port injection spark-ignited gasoline engine model is introduced according to [29]. The second section descibes the design of the explicit MPC which uses the output soft constraints handling algorithm described in Chapter 4 for model of engine and gives some simulation results.

5.1 Port Injection Spark-ignited Gasoline Engine

The majority of modern passenger cars are still equipped with port injection spark-ignited (SI) gasoline engines [29]. The premixed and stoichiometric combustion of the the Otto

process permits an extremely efficient exhaust gas purification with three-way catalyst converters and produces very little particulate matter (PM).
The torque of a stoichiometric SI engine is controlled by the quantity of air/fuel mixture in the cylinder during each stroke. Typically, this quantity is varied by changing the intake pressure and by that the density of the air/fuel mixture. Thus, a throttle plate is used upstream in the intake system.
In next text, SI engine will be briefly described.

5.1.1 Mean-Value Models

The reciprocating engines in passenger cars clearly differ in at least two aspects from continuously operating thermal engines gas turbines:

- the combustion process itself is highly transient (Otto or Diesel cycle, with large and rapid temperature and pressure variations)
- the thermodynamic boundary conditions, that govern the combustion process (intake pressure, composition of air/fuel mixture, etc.) are not constant

The thermodynamic and kinetic process in the first class of phenomena are very fast (few milliseconds for full Otto cycle) and usually not accessible for control purposes. Moreover, the models necessary to describe these phenomena are rather complex and are not useful for the design of real-time control systems.
Usually, the second class of phenomena are taken into account using control-oriented models (COM), and simplifies the fast combustion characteristics as static effects. The underlying assumption is that once all important thermodynamic boundary conditions at the start of an Otto cycle are fixed, the combustion itself will involve in an identical way each time the same initial starting conditions are imposed. Clearly, such models will not be able to reflect all phenomena.
In the COM paradigm, the engine has several input (control) signals, one main disturbance signal (load torque) and several output signals.
The reciprocating behavior of the engine induces another dichotomy in the COM used to describe the engine dynamics:

Mean value models (MVM) continuous COM, which neglect the discrete cycles of the engine and assume all processes and effects are spread out over the engine cycle

Discrete event models (DEM) COM that explicitly take into account the reciprocating behavior of the engine

In MVM, the time t is the independent variable, while in DEM, the crankshaft angle Φ is the independent variable. In MVM, the reciprocating behavior is captured by introducing delays between cylinder-in and cylinder-out effects. For example, the torque produced by the engine does not respond immediately to an increase in the manifold pressure. Only after the induction-to-power-stroke (IPS) delay $\tau_{IPS} = \frac{2\pi}{\omega_e}$ has elapsed the new engine torque will be active.

5.1.2 Port-Injection SI Engines

A typical port-injected SI engine system has the structure shown in figure 5.1 In a mean value approach, the reciprocating behavior of the cylinders is replaced by a continuously working volumetric pump that produces exhaust gases and torque.

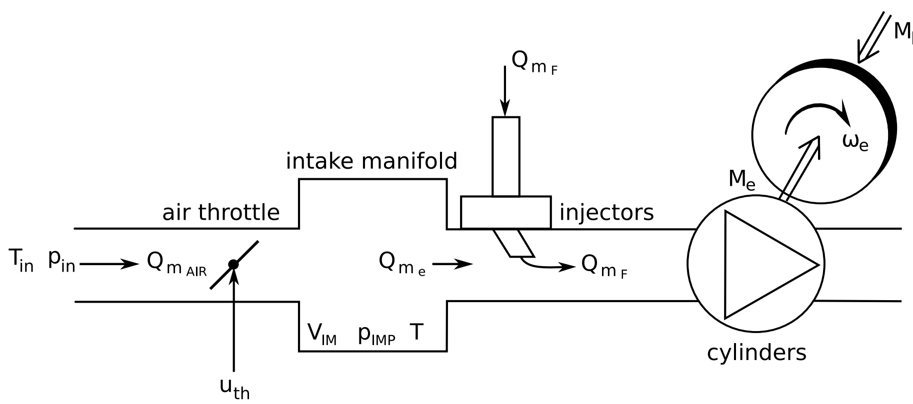

Figure 5.1: Abstract mean-value SI engine structure (according to [29])

The following signal definitions have been used in figure 5.1:

T_{in}	intake temperature
p_{in}	intake pressure (atmospheric)
$Q_{m_{AIR}}$	fresh air mass flow through the valve
u_{th}	control signal of the throttle
V_{IM}	intake manifold volume
p_{IMP}	intake manifold pressure
T	intake manifold temperature
Q_{m_e}	air mass flow entering the cylinders
Q_{m_F}	fuel mass flow entering the cylinders
M_e	engine torque
M_l	load torque
ω_e	engine speed

Next, mean-value models of the most important subsystems of SI are introduced.

5.1.2.1 Air system

5.1.2.1.1 Receiver The basic building block in the air intake system and also in the exhaust part is a *receiver*, i.e., a fixed volume for which the thermodynamic states (pressures, temperatures, etc., as shown in figure 5.2) are assumed to be the same over the entire volume.

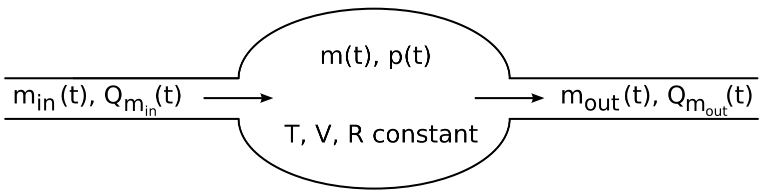

Figure 5.2: Inputs, states and outputs of a receiver (according to [29])

The inputs and outputs are the mass and energy flows, the reservoirs store mass and thermal energy, and the level variables are the pressure and temperature. If one assumes that no heat or mass transfers through the walls and that no substantial changes in potential or kinetic energy in the flow occur, then the following differential equation describe such a receiver.

$$\frac{d}{dt}m(t) = Q_{m_{in}}(t) - Q_{m_{out}}(t), \tag{5.1}$$

with m is air mass [kg] and $Q_{m_{in}}$, $Q_{m_{out}}$ [kg/s] are input, output air mass flow to receiver. Consider the ideal gas law

$$pV = mRT, \tag{5.2}$$

where p [Pa] is the absolute pressure of the gas, V [m^3] is the volume of the gas, R [J/(kg K)] is the specific gas constant and T [K] is the absolute temperature.
Then one can write

$$\frac{d}{dt}p(t) = \frac{RT}{V}\left[Q_{m_{in}}(t) - Q_{m_{out}}(t)\right] \tag{5.3}$$

where the temperature T is assumed to be constant.

5.1.2.1.2 Valve Mass Flow One important variable in engine control is mass flow of *fresh air*. This is controlled by the throttle in the intake manifold system. The flow of fluids (in this case fresh air) between two reservoirs (environment and cylinders) is determined by valves or orifices whose inputs are pressures upstream and downstream. The difference between these two level variables drives the fluid in a nonlinear way through such restrictions. As this problem is at the heart of fluid dynamics in this work, next simplification will be done:

- no friction in the flow,
- no inertial effects in the flow (the piping around the valves is small compared to the receivers to which they are attached)
- completely isolated conditions (no additional energy, mass, etc.)
- all flow phenomenon zero dimensional, i.e., no spatial effects need be considered

For compressible fluids (as air is), the most important flow control block is the *isothermal orifice*. The key assumptions for modeling this device are following:

- No losses occur in the accelerating part (pressure decreases) up to the narrowest point. All the potential energy stored in the flow is converted isentropically [1] into kinetic energy.
- After the narrowest point, the flow is fully turbulent and all of the kinetic energy gained in the first part is dissipated into thermal energy. Moreover, no pressure recuperation takes place.

[1] without change of entropy

The consequence of this is that the pressure in the narrowest point of the valve is approximately equal to the downstream pressure and that the temperature of the flow before and after orifice is the same.
Using the the thermodynamic relationship for isentropic expansion the following equation for the flow can be obtained

$$Q_{m_{AIR}}(t) = c_d \, A(t) \, \frac{p_{in}(t)}{\sqrt{RT}} \, \Psi\left(\frac{p_{in}(t)}{p_{IMP}(t)}\right), \tag{5.4}$$

where $Q_{m_{AIR}}$ [kg/s] is mass flow through the valve, A [m^2] is open area of the throttle, c_d is discharge coefficient, p_{in} [Pa] is pressure upstream of the valve, p_{IMP} [Pa] is pressure downstream of the valve (intake manifold pressure), and $\Psi(.)$ is approximately defined by

$$\Psi\left(\frac{p_{in}(t)}{p_{IMP}(t)}\right) \approx \begin{cases} \frac{1}{\sqrt{2}} & \text{for } p_{IMP} < \frac{1}{2}\, p_{in} \\ \sqrt{\frac{2p_{IMP}(t)}{p_{in}(t)}\left[1 - \frac{p_{IMP}(t)}{p_{in}(t)}\right]} & \text{for } p_{IMP} \geq \frac{1}{2}\, p_{in} \end{cases} \tag{5.5}$$

Open area of the throttle A is controlled by the actuator and following can be written

$$A(t) = \frac{1}{100}\left(\frac{\pi d^2}{4} - A_{leak}\right) u_{th}(t) + A_{leak} = A_C \, u_{th}(t) + A_{leak}, \tag{5.6}$$

with d [m] is throttle diameter, u_{th} [%] is control variable of a throttle, A_{leak} [m^2] throttle opening area when $u_{th} = 0$.

5.1.2.1.3 Engine Mass Flow Ragarding the air system, the engine itself can be approximated as a volumetric pump, i.e., a device that enforces a volume flow approximately proportional to its speed. A typical formulation for such a model is

$$Q_{m_e}(t) \quad = \quad \rho_{in}(t) \, Q_{V_e} = \rho_{in}(t) \, \eta_{VE} \, \frac{V_d}{N} \, \frac{\omega_e(t)}{2\pi}, \tag{5.7}$$

where ρ_{in} [kg/m^3] is density of the gas at the engine's intake (related to the intake pressure and temperature by the ideal gas law (5.2)), η_{VE} is volumetric efficiency, which descibes how far the engine differs from an ideal volumetric device (see below), V_d [m^3] is displacement volume - the volume swept by all the pistons of an engine in a single movement from top dead center to bottom dead center, N is number of revolutions per cycle ($N = 2$ for four-stroke and $N = 1$ for two-stroke engines), ω_e [rad/s] is engine speed.
The volumetric efficiency η_{VE} determines the engine's ability to aspire mixture or air and is of special importance at full-load conditions. It is rather difficult to predict it reliably since many effects influence it (internal exhaust gas recirculation, ram effects in the intake

runner, resonance in the manifold, etc.). Note also that evaporation of the fuel in the intake port causes a substantial reduction of the temperature of the inflowing mixture. This effect can increase volumetric efficiency.
At first approximation, volumetric efficiency can be formulated as

$$\eta_{VE}(t) = a_{VE}\, \omega_e^2(t) + b_{VE}\, \omega_e(t) + c_{VE}, \tag{5.8}$$

where coefficients a_{VE}, b_{VE}, c_{VE} are found during the identification of the engine model.

5.1.2.1.4 Air to Fuel Ratio Air to fuel ratio (AFR) is the mass ratio of air to fuel present during combustion. When all the fuel is combined with all the free oxygen, typically within a engine's combustion chamber, the mixture is chemically balanced and this AFR is called the stoichiometric mixture (often abbreviated to stoich). AFR is an important measure for anti-pollution and performance tuning reasons.

$$\lambda(t) = \frac{Q_{m_{AIR}}(t)}{Q_{m_F}(t)} \tag{5.9}$$

with $Q_{m_{AIR}}$ [kg/s] is mass flow through the valve and Q_{m_F} [kg/s] is fuel mass flow into cylinders.
Note that mixture dynamic is omitted for simplicity.

5.1.2.2 Mechanical System

5.1.2.2.1 Torque Generation The primary objective of an engine is to produce mechanical power. Its speed is a level variable, i.e., it is not arbitrary assignable. However, the torque can be changed arbitrarily at will, by providing the certain amount of mixture into the cylinder and/or its composition. The mean-value engine torque is a nonlinear function of many variables (fuel mass in cylinder, AFR, engine speed, ignition or injection timing, etc.).
Detailed thermodynamic simulations are necessary to correctly predict the engine torque. However, for control purposes such simulations are too time-consuming. Thus, alternative approaches have been investigated. Mostly, some physical insight to separate the different influencing variables and divide the modeling task into several low-dimensional problems is used.
Often, there is a problem to determinate the engine efficiency as many effects influence it. In this work, only *fuel efficiency* would be considered. Also no frictions are also assumed.

The fuel efficiency η_F consists of two main factors: *thermodynamic* and *combustion efficiency.*

$$\eta_f = \eta_t \, \eta_c \tag{5.10}$$

The thermodynamic efficiency η_t of the engine strongly depends to its speed and typically, it has a parabolic form. At very low speeds, the relatively large heat losses through the wall reduce engine efficiency, while at very high speeds, the combustion times become unfavorably large compared to the available interval in the expansion stroke. Note that η_t has magnitude substantially smaller than 1 since it incorporates the basic thermodynamic efficiency mechanism.
Since engine efficiency has parabolic form one can write

$$\eta_t(t) = a_t \, \omega_e^2(t) + b_t \, \omega_e(t) + c_t, \tag{5.11}$$

The combustion efficiency η_c is caused by fuel which has not burnt. Note that, this coeficient is very close to 1.

Engine output power P [W] is

$$P(t) = H_l \, Q_{m_F}(t) \, \eta_f \tag{5.12}$$

where H_l[J/kg] is heating value of the fuel, Q_{m_F} [kg/s] is fuel mass flow into cylinders.
For rotary machine we can also write

$$P(t) = M_e(t) \, \omega_e(t) \tag{5.13}$$

where M_e [Nm] is generated torque.
Thus one can write

$$M_e(t) = \frac{1}{\omega_e(t)} \eta_t \, \eta_c(t) H_l \, Q_{m_F}(t) \tag{5.14}$$

5.1.2.2.2 Engine Speed In a mean-value setting, modeling the engine speed behavior is straightforward process. In fact, the inertia is assumed to be constant and friction and other losses are already included in the torque model described in section before.
The only relevant reservoir is the engine flywheel. It stores kinetic energy and the differential equation for the corresponding level variable ω_e is

$$J \, \frac{d}{dt} \omega_e(t) = M_e(t) - M_l(t), \tag{5.15}$$

where J [m^2 kg] is engine inertia, M_l is load torque which depends on engine speed, drive train properties and external loads like road slope or wind disturbances. The control algorithm has to eliminate the influence of it.

5.1.3 State space model

Now consider the engine model illustrated in figure 5.3. For such model there are two inputs (control variables): Q_{m_F} - fuel mass flow into cylinders, u_{th} - control variable of a throttle. There are also two system outputs: ω_e - engine speed, λ - air to fuel ratio. Model has two internal states: ω_e - engine speed and p_{IMP} - *intake manifold pressure* (IMP) defined from (5.3), (5.7), (5.4) as

$$\frac{d}{dt}\,p_{IMP}(t) = \frac{RT}{V_{IM}}\,\left[Q_{m_{AIR}}(t) - Q_{m_e}(t)\right], \tag{5.16}$$

where $V_{IM}[\mathrm{m}^3]$ is intake manifold volume.

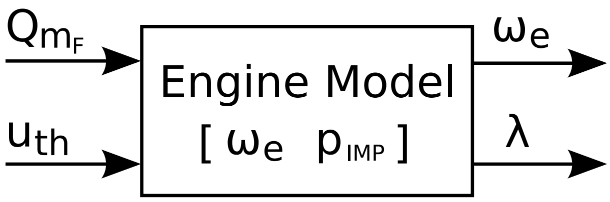

Figure 5.3: Considered engine model with inputs, outputs and states

Consider constant, time invariant input manifold temperature T, constant input pressure p_{in} and four stroke engine, that is

$$T(t) = T \qquad p_{in}(t) = p_{in} \qquad N = 2. \tag{5.17}$$

Now state space model construction will be descibed for such a model. First, substitute the equation of generated torque (5.14) together with engine efficiency (5.11) to (5.15)

$$J\,\frac{d}{dt}\,\omega_e(t) = J\,\dot{\omega}_e(t) = \left(a_t\omega_e(t) + b_t + \frac{c_t}{\omega_e(t)}\right) H_l\,\eta_c\,Q_{m_F}(t) - M_l(t). \tag{5.18}$$

then put together the equation of fresh air mass flow (5.4) with the equation of volumetric pump (5.7) and intake manifold pressure equation (5.16)

$$\begin{aligned}\frac{d}{dt}\,p_{IMP}(t) \quad &= \quad \dot{p}_{IMP}(t) = \frac{RT}{\sqrt{RT}}\,\frac{p_{in}}{V_{IM}}\,c_d\,(A_C\,u_{th}(t) + A_{leak})\,\Psi\left(\frac{p_{in}}{p_{IMP}(t)}\right) - \\ &\quad - \frac{p_{IMP}(t)}{N}\,\frac{V_d}{V_{IM}}\,\frac{\omega_e(t)}{2\pi}\,\left(a_{VE}\,\omega_e^2(t) + b_{VE}\,\omega_e(t) + c_{VE}\right)\end{aligned} \tag{5.19}$$

Let's construct the output equation of AFR now. From (5.12) and (5.4) we get:

$$\lambda(t) = c_d\,(A_C\,u_{th}(t) + A_{leak})\,\frac{p_{in}}{\sqrt{RT}}\,\Psi\left(\frac{p_{in}}{p_{IMP}(t)}\right)\frac{1}{Q_{m_F}(t)} \tag{5.20}$$

5.1.3.1 Linearization

As one can see, process equations (5.20), (5.18), (5.19) are not linear. For linear control it is necessary to linearize them. Controller will then work in neighborhood of working point

$$p_{IMP} = p_{IMP,0} \quad \omega_e = \omega_{e,0} \quad u_{th} = u_{th,0} \quad Q_{m_F} = Q_{m_F,0} \quad \lambda = \lambda_0 \quad M_l = M_{l,0}. \tag{5.21}$$

From equation of the air throttle (5.5) it is also clear that there will exist two separate models for each of validity condition. Next, we will focus on model, where

$$p_{IMP}(t) \geq \frac{1}{2}\, p_{in} \tag{5.22}$$

Thus, we define linearized engine model [2] as

$$\Delta\dot{\omega}_e = \left.\frac{\partial\dot{\omega}_e}{\partial\omega_e}\right|_{w.p.}\Delta\omega_e + \left.\frac{\partial\dot{\omega}_e}{\partial p_{IMP}}\right|_{w.p.}\Delta p_{IMP} + \left.\frac{\partial\dot{\omega}_e}{\partial Q_{m_F}}\right|_{w.p.}\Delta Q_{m_F} + \left.\frac{\partial\dot{\omega}_e}{\partial u_{th}}\right|_{w.p.}\Delta u_{th} \tag{5.23}$$

$$\Delta\dot{p}_{IMP} = \left.\frac{\partial\dot{p}_{IMP}}{\partial\omega_e}\right|_{w.p.}\Delta\omega_e + \left.\frac{\partial\dot{p}_{IMP}}{\partial p_{IMP}}\right|_{w.p.}\Delta p_{IMP} + \left.\frac{\partial\dot{p}_{IMP}}{\partial Q_{m_F}}\right|_{w.p.}\Delta Q_{m_F} + \left.\frac{\partial\dot{p}_{IMP}}{\partial u_{th}}\right|_{w.p.}\Delta u_{th} \tag{5.24}$$

$$\Delta\lambda = \left.\frac{\partial\lambda}{\partial\omega_e}\right|_{w.p.}\Delta\omega_e + \left.\frac{\partial\lambda}{\partial p_{IMP}}\right|_{w.p.}\Delta p_{IMP} + \left.\frac{\partial\lambda}{\partial Q_{m_F}}\right|_{w.p.}\Delta Q_{m_F} + \left.\frac{\partial\lambda}{\partial u_{th}}\right|_{w.p.}\Delta u_{th}. \tag{5.25}$$

After derivation we have

$$J\,\Delta\dot{\omega}_e(t) = H_l\, Q_{m_F,0}\, \eta_c \left(a_t - \frac{c_t}{\omega_{e,0}}\right)\Delta\omega_e(t) + \frac{H_l}{\omega_{e,0}}\,\left(a_t\,\omega_{e,0}^2 + b_t\,\omega_{e,0} + c_t\right)\eta_c\,\Delta Q_{m_F}(t) \tag{5.26}$$

$$\begin{aligned}\Delta\dot{p}_{IMP}(t) = & -\frac{p_{IMP,0}}{2\pi N}\,\frac{V_d}{V_{IM}}\,\left(3a_{VE}\,\omega_{e,0}^2 + 2b_{VE}\,\omega_{e,0} + c_{VE}\right)\Delta\omega_e(t) + \\ & + \left[\frac{RT}{\sqrt{RT}}\,c_d\left(A_C\,u_{th,0} + A_{leak}\right)\frac{p_{in} - 2p_{IMP,0}}{p_{in}^2 F} - \right. \\ & \left. -\frac{\omega_{e,0}}{2\pi N}\,\frac{V_d}{V_{IM}}\,\left(a_{VE}\,\omega_{e,0}^2 + b_{VE}\,\omega_{e,0} + c_{VE}\right)\right]\Delta p_{IMP}(t) + \\ & + \frac{p_{in}}{V_{IM}}\,\frac{RT}{\sqrt{RT}}\,c_d\,A_C\,F\Delta u_{th}(t)\end{aligned} \tag{5.27}$$

$$\begin{aligned}\Delta\lambda(t) = & \; c_d\left(A_C u_{th,0} + A_{leak}\right)\frac{p_{in}}{\sqrt{RT}}\,\frac{1}{Q_{m_F,0}}\,\frac{p_{in} - 2p_{IMP,0}}{p_{in}^2 F}\,\Delta p_{IMP}(t) + \\ & + c_d\,A_C\frac{p_{in}}{\sqrt{RT}}\,\frac{1}{Q_{m_F,0}}\,F\Delta u_{th}(t) - c_d\,\left(A_C\,u_{th,0} + A_{leak}\right)\frac{p_{in}}{\sqrt{RT}}\,\frac{F}{Q_{m_F,0}^2}\,\Delta Q_{m_F}(t)\end{aligned} \tag{5.28}$$

[2]Time indexes are omitted for simplicity.

where $F = \sqrt{\frac{2p_{IMP,0}}{p_{in}}\left[1 - \frac{p_{IMP,0}}{p_{in}}\right]}$.

It should be denoted that load torque M_l is not included in linearized equations of the system, because it is unmesured input disturbance which control algorithm must eliminate.

Thus the state space model in neighborhood of working point (5.21) is

$$\begin{aligned} \Delta\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}_c\Delta\boldsymbol{x}(t) + \boldsymbol{B}_c\Delta\boldsymbol{u}(t) \\ \Delta\boldsymbol{y}(t) &= \boldsymbol{C}_c\Delta\boldsymbol{x}(t) + \boldsymbol{D}_c\Delta\boldsymbol{u}(t), \end{aligned} \tag{5.29}$$

where the matrices $\boldsymbol{A}_c$, $\boldsymbol{B}_c$, $\boldsymbol{C}_c$, $\boldsymbol{D}_c$ are parametrized by equations (5.26), (5.21) and

$$\Delta\boldsymbol{x}(t) = \begin{bmatrix} \Delta\omega_e(t) \\ \Delta p_{IMP}(t) \end{bmatrix} \qquad \Delta\boldsymbol{u}(t) = \begin{bmatrix} \Delta Q_{m_F}(t) \\ \Delta u_{th}(t) \end{bmatrix} \qquad \Delta\boldsymbol{y}(t) = \begin{bmatrix} \Delta\omega_e(t) \\ \Delta\lambda(t) \end{bmatrix}. \tag{5.30}$$

5.1.3.2 Model Parametrization and Verification

The nonlinear model of the SI engine ((5.20), (5.16), (5.18)) and linearized model (5.26) were implemented in MATLAB® and Simulink®[3] with following parameters [4]:

$$\begin{array}{llll} H_l = 4.7\ [MJ/kg] & V_d = 2\ [dm^3] & V = 1\ [dm^3] & J = 0.4\ [kg/m^2] \\ R = 287\ [J/(kg\ K)] & T = 293.15\ [K] & p_{in} = 100\ [hPa] & N = 2\ [-] \\ \eta_c = 0.9\ [-] & c_d = 0.1\ [-] & & \end{array}$$

For linear control, the following working poit was choosen:

$$\begin{array}{lll} p_{IMP,0} = 51.7\ [kPa] & \omega_{e,0} = 302.2\ [rad/s] & u_{th,0} = 27\ [\%] \\ Q_{m_F,0} = 1.7\ [g/s] & \lambda_0 = 16\ [-] & M_{l,0} = 100\ [Nm]. \end{array}$$

The continuous nonlinear and linearized model have been sampled with sample period $T_S = 50\ ms$. The figure 5.4 shows the comparing of the step responses of the nonlinear and for the linearized model in neighborhood of working point. From the step response of the fuel mass flow Q_{m_F} to engine's speed ω_e (top image) one can see the strong non-linearity effect ($\frac{1}{\omega_e}$ term in (5.18)) which causes diferent behavior of the nonlinear and linearized model even for small step of Q_{m_F}. The intuitive independecy of the engine's

[3]MATLAB and Simulink are registered trademarks of The MathWorks, Inc.

[4]The list of all parameters is stated in *SI_engine_init.m* script on attached CD.

speed to position of the throttle u_{th} is also clear from this figure. It means that for increasing the engine's speed it is neccessary to increase the fuel mass flow into the cylinders. The increasing of the fresh air mass flow, which is controlled by the throttle position u_{th} will influence only the AFR λ. It can be seen in the second image, where the fast dynamics of the AFR is also evident. Note that the direct feedthrough of the fuel mass flow Q_{m_F} to AFR, which is evident from (5.20) can be seen here too.

After discretization of the linearized model we have

$$\begin{aligned} \Delta\boldsymbol{x}_{k+1} &= \boldsymbol{A}\Delta\boldsymbol{x}_k + \boldsymbol{B}\Delta\boldsymbol{u}_k \\ \Delta\boldsymbol{y}_k &= \boldsymbol{C}\Delta\boldsymbol{x}_k + \boldsymbol{D}\Delta\boldsymbol{u}_k, \end{aligned} \tag{5.31}$$

where the system matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ are the discrete equivalent of system (5.29) and

$$\Delta\boldsymbol{x}_k = \begin{bmatrix} \Delta\omega_{e_k} \\ \Delta p_{IMP_k} \end{bmatrix} \qquad \Delta\boldsymbol{u}(t) = \begin{bmatrix} \Delta Q_{m_F k} \\ \Delta u_{th_k} \end{bmatrix} \qquad \Delta\boldsymbol{y}_k = \begin{bmatrix} \Delta\omega_{e_k} \\ \Delta\lambda_k \end{bmatrix}. \tag{5.32}$$

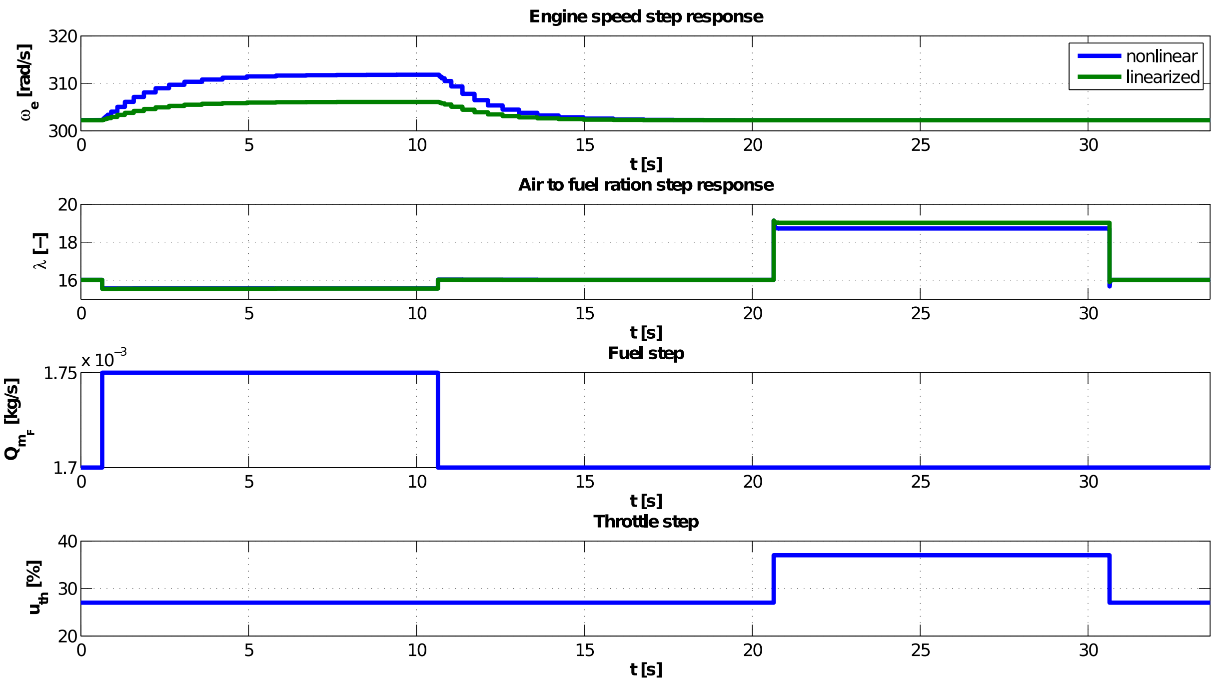

Figure 5.4: Step responses of nonlinear and linearized model of the SI engine.

5.2 The Controller Design

From figure 5.5 is clear that the slowest mode of the engine is in steady state aproximately in 5s, therefore the prediction horizont was choosen to $T_p = 100$. For reduce the complexity of the result controller the control horizont was choosen to $T_c = 1$ and the algorithm for the soft output constraints described in chapter 4 was used.
Consider following requirements:

- offset-free tracking of a given engine speed ω_e
- range control of the AFR λ among the upper $\overline{\lambda}$ and lower $\underline{\lambda}$ constraints
- limited control action of fuel mass flow $0.3 \leq Q_{m_F} \leq 5$ [mg/s]
- limited control action of position of the throttle $5 \leq u_{th} \leq 100$ [%]

From the σ-plot of condition number of the engine's model in figure 5.5 it is clear that engine is ill-conditioned system on high frequencies and therefore sensitive to input uncertainty [59].

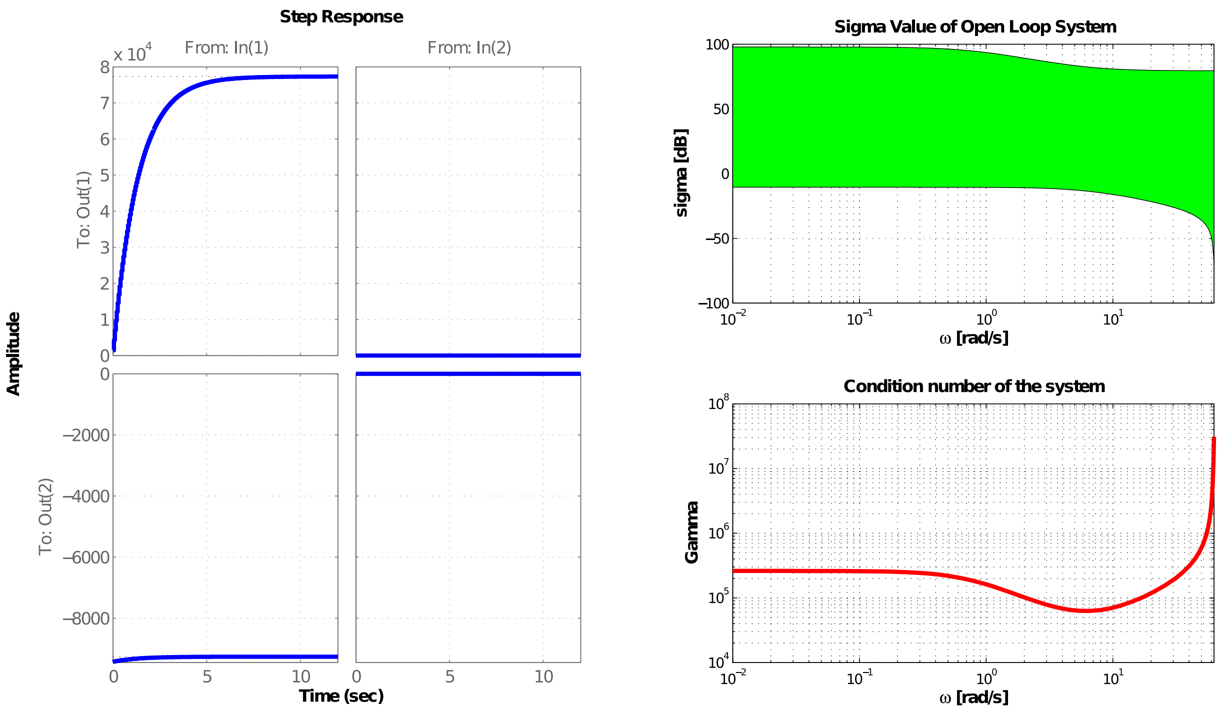

Figure 5.5: Step response and σ-plots of open loop and condition number of the linearized model of the SI engine.

Note: For distinguish the Δu-formulation of MPC and Δ symbol of increment in neighborhood of working point for linearized system model the new labeling for control increment which is result of MPC will be considered. The formulation δu_k means the control move of MPC in step k, that is

$$\delta \boldsymbol{u}_k = \Delta \boldsymbol{u}_k - \Delta \boldsymbol{u}_{k-1} = (\boldsymbol{u}_k - \boldsymbol{u}_0) - (\boldsymbol{u}_{k-1} - \boldsymbol{u}_0) = \boldsymbol{u}_k - \boldsymbol{u}_{k-1}, \tag{5.33}$$

where $\boldsymbol{u}_0$ is value of control in working point. □

5.2.1 Formulation

From equation of the engine speed (5.18) it is clear that one state disturbance enters into the system. This disturbance represents the engine's load M_l which depends on the road slope or wind which affects to a car. From the equation of AFR (5.9) is also clear its nonlinearity, thus it is reasonably to modeling this as the output disturbance. Thus the disturbance model for engine would be

$$\boldsymbol{G}_d = \begin{bmatrix} -1 & 0 \end{bmatrix}^T \qquad \boldsymbol{G}_p = \begin{bmatrix} 0 & 1 \end{bmatrix}^T. \tag{5.34}$$

Thus according to (2.107) it is neccessary to augment the system (5.32). Moreover, in order to use the Δu-formulation of MPC the system model (5.32) has to be augmented with last control move and reference, i.e.

$$\begin{aligned} \Delta \tilde{\boldsymbol{x}}_{k+1} &= \tilde{\boldsymbol{A}}\, \Delta \tilde{\boldsymbol{x}}_k + \tilde{\boldsymbol{B}}\, \delta \boldsymbol{u}_k \\ \Delta \boldsymbol{y}_k &= \tilde{\boldsymbol{C}}\, \Delta \tilde{\boldsymbol{x}}_k + \tilde{\boldsymbol{D}}\, \delta \boldsymbol{u}_k, \end{aligned} \tag{5.35}$$

where the system matrices $\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{B}}$, $\tilde{\boldsymbol{C}}$ and $\tilde{\boldsymbol{D}}$ are built according to (2.107) and (2.49).

From the second requirement of range control of AFR among the upper and lower constraints it is clear that 2 slack variables and 2 states have to be add to cost function and to system model.

Finally, according to (2.72) one can find the optimal control of the engine as the solution of the cost function

$$\boldsymbol{h}^* = \arg\min_{\boldsymbol{h}} J(\boldsymbol{h}|\tilde{\boldsymbol{x}}_k) = \frac{1}{2}\boldsymbol{h}^T \acute{\boldsymbol{H}} \boldsymbol{h} + \begin{bmatrix} \tilde{\boldsymbol{x}}_k \\ \hline \boldsymbol{\lambda}_k \end{bmatrix}^T \acute{\boldsymbol{F}} \boldsymbol{h} + \begin{bmatrix} \tilde{\boldsymbol{x}}_k \\ \hline \boldsymbol{\lambda}_k \end{bmatrix}^T \acute{\boldsymbol{Y}} \begin{bmatrix} \tilde{\boldsymbol{x}}_k \\ \hline \boldsymbol{\lambda}_k \end{bmatrix}, \tag{5.36}$$

$$\text{s.t.} \qquad \boldsymbol{G}\, \delta \boldsymbol{u}_{k,T_c-1} \leq \boldsymbol{W} + \boldsymbol{E}\, \tilde{\boldsymbol{x}}_k, \tag{5.37}$$

with

$$\boldsymbol{h} \;=\; \begin{bmatrix} \delta \boldsymbol{u}_{k,T_c-1} & \boldsymbol{\epsilon} \end{bmatrix}^T \qquad \underline{\overline{\boldsymbol{\lambda}_k}} = \begin{bmatrix} \overline{\lambda_k} & \underline{\lambda_k} \end{bmatrix}^T \tag{5.38}$$

and where the constraints matrices $\boldsymbol{G}$, $\boldsymbol{W}$ and $\boldsymbol{E}$ are built according to section 2.3.2 and 4.2. On the AFR output, the algorithm which were introduced in chapter 4 is used.

The two controllers were obtained using the MPT Toolbox [37] with following settings:

$$q = 500, \quad \boldsymbol{R} = \begin{bmatrix} 10000 & 0 \\ 0 & 100 \end{bmatrix}, \quad \boldsymbol{\rho} = \begin{bmatrix} 10e6 & 0 \\ 0 & 10e6 \end{bmatrix}, \quad T_c = 1, \quad T_p = 100. \tag{5.39}$$

The first controller uses the constraints horizon $T_{cs} = 10$ and the second one uses our algorithm of output soft constraints handling described in Chapter 4 for reducing the control law complexity with $T_n = 50$.

Note that result controller, which uses our algorithm, consists of 36 regions and the second one consists of 260 regions. Thus the controller which uses our algorithm is much less complex.

For estimating the system state, the Kalman filter 2.3.5.3 augmented to disturbances model (5.34) was used.

5.2.2 Assessment

Figure 5.6 shows that thanks to Unknown Input Observer method the offset-free tracking of engine speed is guaranteed even for large steps of references from the neighborhood of the working point ($\omega_{e,0} = 302.2\ [rad/s]$) where the model is uncertain or where the model is not valid (see 5.8, where the intake manifold pressure must be greater than the condition of validity 5.22).

Note again that model of engine is nonlinear and used MPC works only with linearized model. The off-set free tracking is also guaranteed when the load (disturbance) is present. At 16 s the load of 50 $[Nm]$ enters to the system (see figure 5.8) and the controller arranges that the engine speed is returned back to reference value.

Air to fuel ratio response shows that the soft constraints are satisfied if it is possible. The peaks of AFR are caused by the large changes of fuel mass flow shown in figure 5.7 (mainly due to the direct feedthrough).

As one can see the closed loop responses are the same for the much more complex controller which uses the constraints horizon and for the less complex controller which uses our constraints handling algorithm, thus it is possible to use our algorithm even for uncertain nonlinear models.

Note that, research of influence of parameter T_n to frequency response of engine leads to very similar results as were introduced in example 4.1. But because the eigenvalues of the engines's model are real, the influence is not so evident.

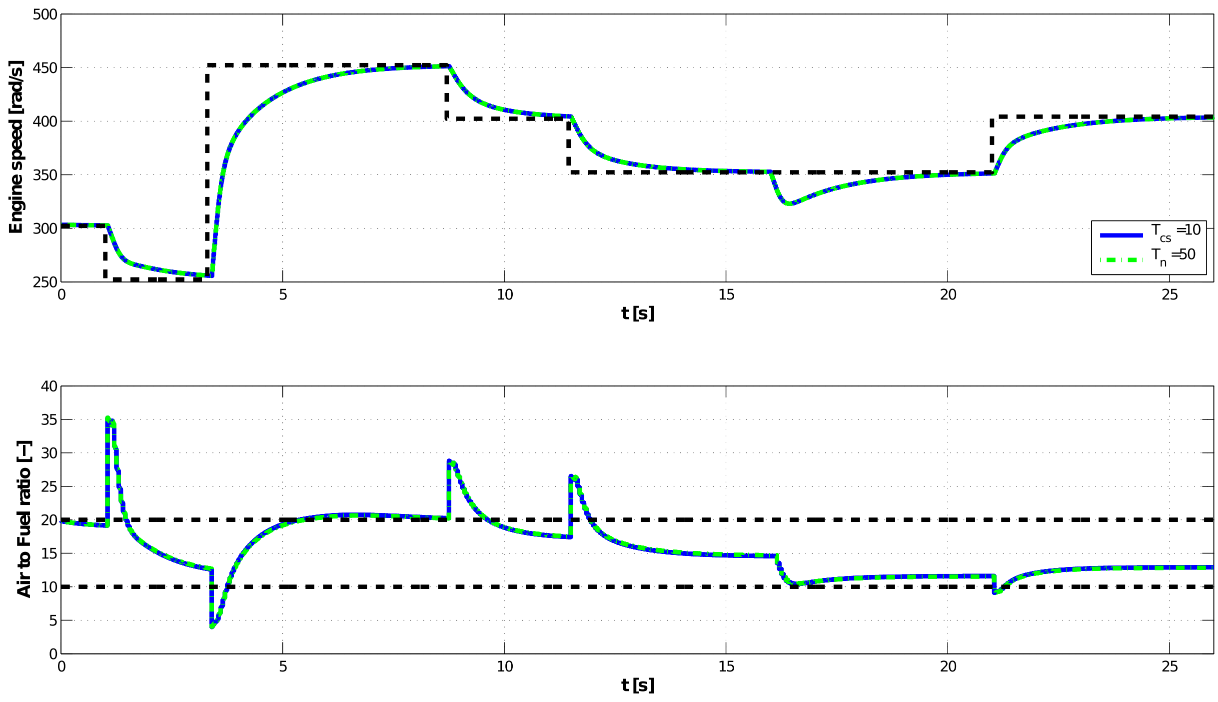

Figure 5.6: Offset-free tracking of the engine speed and range control of the AFR.

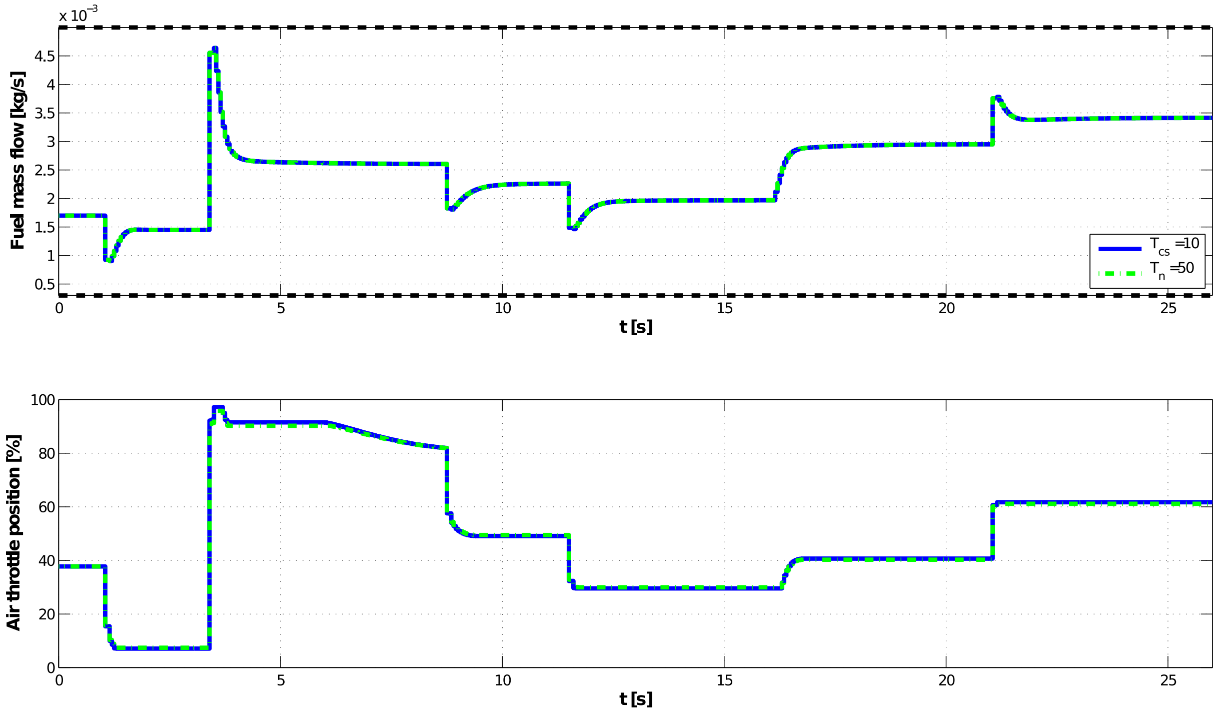

Figure 5.7: Control actions of fuel and air throttle during the simulation.

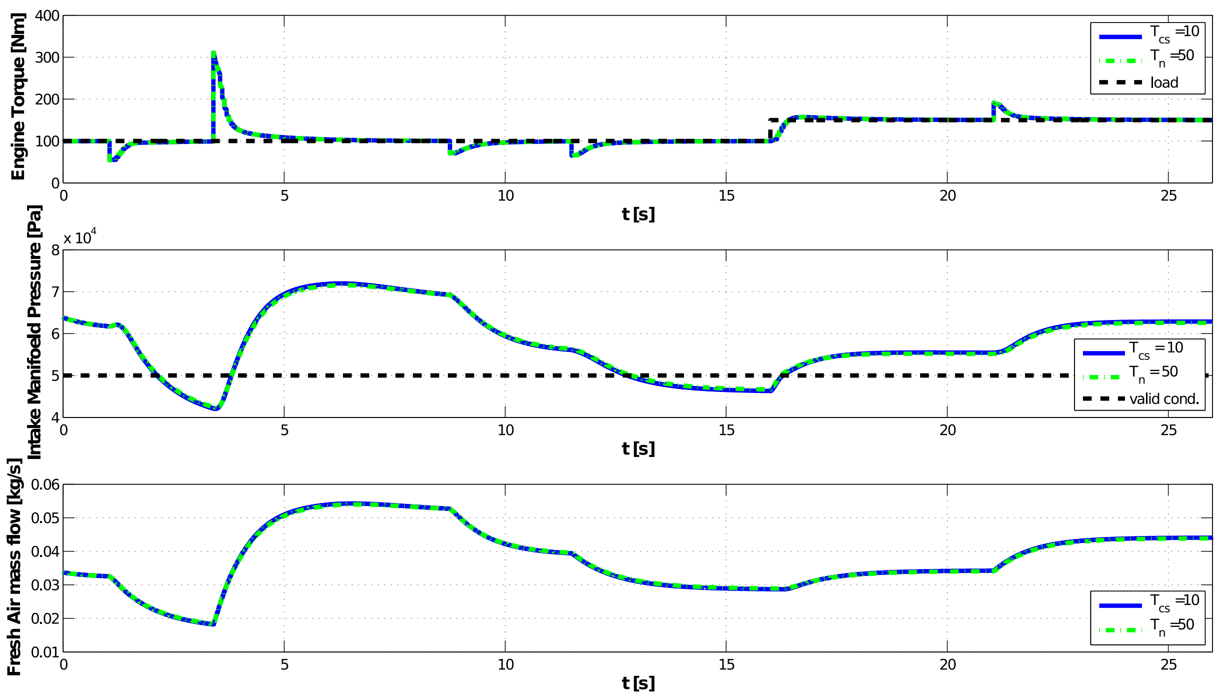

Figure 5.8: The Kalman's estimation of variables of the engine during the simulation.

Chapter 6

Conclusions

In the first chapter of this work the main principles of MPC were introduced. The Unknown Input Observer method which allows an estimation of the unknown input to the controlled system to eliminating steady-state offset was described there too. The abilities of this method were shown on examples of MPC control of the linear models and later (in Chapter 5) the method was used to eliminate the tracking error of engine speed of nonlinear model of port injection spark-ignited gasoline engine.
In Chapter 3 the Explicit MPC were described together with its main drawback - the large growth of complexity of a mp-QP solution with increasing number of constraints. In Chapter 4 the influence of model uncertainty to constraints handling for MPC with small prediction horizon was shown. The focus was on the soft output constraints. The hard output constraints were not considered because they can cause the infeasible solution of MPC.

In Chapter 4 the new output soft constraints handling algorithm for MPC was introduced too. The main idea of the algorithm is that the output soft constraints are considered only in one step of the outputs prediction. Therefore when the constraint is violated the controller ignores it up to the step where the constraint is considered, thus no control action is generated immediately after the violation. This leads to more careful control and the choice of the step, where the constraint is considered, is a parameter of how much the control will be careful. The influence of the algorithm on frequency and time response of the result closed loop system was shown on example. It was shown that with increasing the step, where the constraints are considered, the closed loop system will more and more damp the high frequencies where the system is ill-conditioned and uncertainties or disturbances can arise. Thus the result controller will be more robust.

The abilities of the algorithm were also shown on control of the nonlinear model of the port injection spark-ignited gasoline engine, where the controller which used our algorithm reached similar closed loop performance as more complex controller.

It was also shown that this algorithm does not increase the complexity of the resulting Explicit MPC so much, thus the range control which will use this algorithm can be used even for high-speed systems.

The algorithm presented in this work assumes that all of the output soft constraints are considered in one step. In the future works, the constraints can be considered in more steps or in different steps for each system output/constraint. Also the stability guarantee proofs of range control with using described algorithm needs to be found.

Bibliography

[1] Akesson, I.N.: *MPC in Column control*
ABB, BA ATCF Control and Force Measurement, 2002

[2] Antsaklis, P.J.; Michel, A.N.: *A Linear System Primer*
USA, New York: Birkhäuser, 2007, ISBN: 9780817644604

[3] Badreddine, B.; Faouzi, B.; Mekki, K.: *Robust MPC for nonlinear multivariable systems*
Control & Automation, 2007. MED '07. Mediterranean Conference, Jun 2007

[4] Bartlett, R.A.; Biegler, L.T.: *A dual, active-set, Schur-complement method for large-scale and structured convex quadratic programming*
Optimization and Engineering, Volume 7, Mar 2006, Pages: 5-32

[5] Bemporad, A.; Morari, M.: *Robust model predictive control: A survey*
Lecture Notes in Control and Information Sciences, Volume 245,1999, Pages: 207-226

[6] Bemporad, A.; Borrelli, F.; Morari, M.: *Model Predictive Control Based on Linear Programming - The Explicit Solution*
Automatic Control, IEEE Transactions on, Volume 47, Issue 12, Dec 2002 Pages: 1974-1985

[7] Bemporad, A.; Morari, M.; Dua, V.; Pistikopoulos, E.F.: *The explicit linear quadratic regulator for constrained systems*
Automatica, Volume 38, Issue 1, Jan 2002, Pages: 3-20

[8] Bezzo, F.; Micheletti, F.; Muradore, R.; Barolo, B.: *Using MPC to control middle-vessel continuous distillation columns*
Journal of Process Control, Volume 15, Issue 8, Dec 2005, Pages: 925-930

[9] Borrelli, F.; Pekar, J.; Baotic, M.; Stewart, G.: *On the Computation of Linear Model Predictive Control Laws*
Preprint submitted to 48th IEEE Conference on Decision and Control, Recieved Mar 6, 2009

[10] Borrelli, F.; Baotic, M.; Pekar, J.; Stewart, G.: *On the Complexity of Explicit MPC Laws*
Preprint submitted to European Control Conference 2009, Recieved Nov 1, 2008

[11] Borrelli, F.; Morari, M.: *Offset Free Model Predictive Control*
Decision and Control, 46th IEEE Conference, 2007

[12] Boyd, S.P; Vandenberghe, L.: *Convex optimization*
Great Britan,: Cambridge University Press, 2004

[13] Cagienard, R.; Grieder, P.; Kerrigan, E.C.; Morari, M.: *Move blocking strategies in receding horizon control*
Journal of Process Control, Volume 17, Issue 6, Jul 2007, Pages: 563-570

[14] Couchman, P.; Kouvaritakis, B.; Cannon, M.: *MPC on state space models with stochastic input map*
Decision and Control, 2006 45th IEEE Conference on, San Diego, CA, Dec 2006, Pages: 3216-3221

[15] Doyle, J.; Stein, G.: *Multivariable feedback design: Concepts for a classical/modern synthesis*
Automatic Control, IEEE Transactions on, Volume 26, Issue 1, Feb 1981, Pages: 4-16

[16] Dua, V.; Pistikopoulos, E. N: *An algorithm for the solution of multiparametric mixed integer linear programming problems*
Annals of Operations Research, Volume 99, Dec 2000, Pages: 123-139

[17] Ekman, M: *Bilinear black-box identification and MPC of the activated sludge process*
Journal of Process Control, Volume 18, Issues 7-8, Aug-Sep 2008, Pages: 643-653

[18] Fattouh, A.; Sename, O.; Dion, J.M. *An unknown input observer design for linear time-delay systems*
Decision and Control, 1999. Proceedings of the 38th IEEE Conference on, Volume 4, Dec 1999, Page: 4222-4227

[19] Ferreau, H.J.: *An Online Active Set Strategy for Fast Solution of Parametric Quadratic Programs with Applications to Predictive Engine Control*
Diplom thesis, University of Heidelberg, 2006

[20] Ferreau, H.J.; Ortner, P.; Langthaler, P.; Re, L.; Diehl, M.: *Predictive control of a real-world Diesel engine using an extended online active set strategy*
Annual Reviews in Control, Volume 31, Issue 2, Feb 2007, Pages: 293-301

[21] Francis, B.A.; Wonham, W.M.: *The internal model principle for linear multivariable regulators*
Applied Mathematics and Optimization, Volume 2, Number 2, June, 1975, Pages: 170-194

[22] Franklin, G.F.; Powell, J.D.; Emami-Naeini, A.: *Feedback control of dynamic systems - 5th Edition*
USA, New Jersey, Pearson Education, 2006, ISBN: 0131499300

[23] Fu, Y.; Wu, D.; Zhang, P.; Duan, G.: *Design on unknown input observer with* H_∞ *performance for linear time-delay systems*
Journal of Systems Engineering and Electronics, Volume 17, Issue 3, Sep 2006, Pages: 606-610

[24] Garcia, D.; Karimi, A.; Longchamp, R.: *PID Controller Design for Multivariable Systems Using Gershgorin Bands*
The 16-th IFAC World Congres, 2005

[25] Gentle, J.E.: *Numerical Linear Algebra for Applications in Statistics*
Springer, 1998, ISBN: 978-0-387-98542-8

[26] Gill, P. E.; Gould, N.I.M, Murray, W.; Saunders, M.A.; Wright, M.H: *A weighted gram-schmidt method for convex quadratic programming*
Mathematical Programming, Volume 30, Oct 1984, Pages: 176-195

[27] Grieder, P.; Morari, M.: *Complexity Reduction of Receding Horizon Control*
IEEE Conference on Decision and Control, Volume 3, Issue 9-12, Dec 2003, Pages: 3179-3184

[28] Gohberg, I.; Koltracht, I.: *Structured condition number for linear matrix structures*
IMA Preprint, Sep 1992

[29] Guzzella, L.; Onder, C.H.: *Introduction to Modeling and Control of Internal Combustion Engine Systems*
Germany, Berlin, Springer, 2004, ISBN: 3-540-22274

[30] Havlena, V.; Findejs, J.: *Application of model predictive control to advanced combustion control*
Control Engineering Practice, Volume 13, Issue 6, 2005, Pages: 671-680

[31] Havlena, V.; Štecha, J.: *Moderní teorie řízení - textbook[cz]*
Czech Republic, Czech Technical University in Prague, Faculty of Electrical Engineering, 1999

[32] Hegreneas, O.; Gravdahl, J.; Tøndel, P.: *Spacecraft attitude control using explicit model predictive control*
Automatica, Volume 41, Issue 12, 2005, Pages: 2107-2114

[33] Hovland, S.; Gravdahl, J.T., Willcox, K.E.: *Explicit Model Predictive Control for Large-Scale Systems via Model Reduction*
Massachusetts Inst of Tech Cambridge Dept of Aeronautics and Astronautics, Dec 2006

[34] Chia, J.W.: *Quasi time – optimal PID control of multivariable systems*
Journal of the Chinese Institute of Engineers, Volume 22, Issue 5, Sep 1999, Pages: 617-625

[35] Kalman, R. E.: *A New Approach to Linear Filtering and Prediction Problems*
Transaction of the ASME - Journal of Basic Engineering, 1960, Pages: 35-45

[36] Katsuhiko, O.: *Matlab for Control Engineers*
Prentice Hall, 2007, ISBN: 0136150772

[37] Kvasnica, M.; Grieder, P.; Baotic, M.; Christophersen, F.J.: *Multi-Parametric Toolbox (MPT)*
Institut fur Automatik, ETH - Swiss Federal Institute of Technology, CH-8092 Zurich, 2006

[38] Kwakernaak, H.: Sivan, R.: *Linear Optimal Control Systems*
John Wiley & Sons, New York, 1972, ISBN: 0471511102

[39] Laub, A.J.: *A schur method for solving algebraic Riccati equations*
Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on, Volume 17, Jan 1978, Pages: 60-65

[40] Lee, J.H.; Morari, M.; Garcia, C.E.: *State-space Interpretation of Model Predictive Control*
Automatica, Volume 30, Issue 4, 1994, Pages: 707-717

[41] Lee, J.H.; Gelormino, M.S.; Morari, M.: *Model predictive control of multi-rate sampled-data systems: a state-space approach*
International Journal of Control, Volume 55, Issue 1, Jan 1992 , Pages: 153-191

[42] Ling. K.V; Maciejowski, J.; Wu, B.F.: *Multiplexed Model Predictive Control*
A *STAR project: Model Predictive Control on a Chip, Jul 2006

[43] Lu, J. Z.: *Challenging control problems and emerging technologies in enterprise optimization. in '6th IFAC Symposium on Dynamics and Control Process Systems'.*
IFAC, Korea, Chejudo Island, 2001

[44] Luenberger, D.G.: *Observing the State of a Linear System*
IEEE Transactions on Military Electronics, Apr 1964, Pages: 74-80

[45] MacFarlane, A. G. J.; Kouvaritakis, B.: *A design technique for linear multivariable feedback systems*
International Journal of Control, Volume 25, Issue 6, Jun 1977, Pages: 837-874

[46] Mayne, D. Q.; Seron, M. M.; Rakovic, S.V.: *Robust model predictive control of constrained linear systems with bounded disturbance*
Automatica, Brief paper accepted on the 23th of August 2004.

[47] Meader, U.; Morari, M.: *Offset-Free Reference Tracking for Predictive Controllers*
Decision and Control, 46th IEEE Conference, 2007

[48] Morari, M.; Lee, J.H.: *Model predictive control: past, present and future*
Computers and Chemical Engineering, Volume 23, 1997, Pages: 667-682

[49] Morari, M.; Zafiriou, E.: *Robust Process Control*
Prentice-Hall, Inc., New Jersey, 1989, ISBN: 0-13-782153-0

[50] Muske, K.R.; Badgwell, T.A.: *Disturbance modeling for offset-free linear model predictive control*
Journal of Process Control, Volume 12, Issue 5, Aug 2002, Pages: 617-632

[51] Ortner, P.; Re, L.: *Predictive Control of a Diesel Engine Air Path*
Control Systems Technology, IEEE Transactions on, Volume 15, Issue 3, May 2007, Pages: 449-456

[52] Pekař, J.: *Robust Model Predictive Control - Dissertation*
Czech Republic, Czech Technical University in Prague, Faculty of Electrical Engineering, 2005

[53] Qin, S. J.; Badgwell, T. A.: *A survey of industrial model predictive control technology*
Control Engineering Practice, Volume 11, 2003, Pages: 733-764

[54] Rao, C. V.; Wright, S. J.; Rawlings, J. B.: *Application of Interior-Point Methods to Model Predictive Control*
Journal of Optimization Theory and Applications, Volume 99, Dec 1998, Pages: 723-757

[55] Ricker, N.L.: *Model predictive control with state estimation*
Ind. Eng. Chem. Res., Volume 29, Issue 3, Mar 1990, Pages: 374-382

[56] Rossiter, J.A.: *Model-Based Predictive Control: A Practical Approach*
USA, Florida: CRC Press LCC, 2003

[57] Řehoř, J.: *Explicitní řešení úlohy prediktivní regulace - Diploma Thesis [cz]*
Czech Republic, Czech Technical University in Prague, Faculty of Electrical Engineering, 2008

[58] Simon, D.: *Optimal State Estimation, Kalman, H_∞, and Nonlinear Approaches*
USA, A John Wiley & Sons, Inc., 2006, ISBN: 100471708585

[59] Skogestad, S.; Havre, K.: *The use of RGA and condition number as robustness measures*
Computers & Chemical Engineering, Volume 20, Supplement 2, 1996, Pages: S1005-S1010

[60] Štěcha, J.: *Optimální rozhodování a řízení - textbook[cz]*
Katedra řídicí techniky, Czech Technical University in Prague, 1999

[61] The MathWorks, Inc.: *Matlab Help*
Matlab R2008b, 2008

[62] Tøndel, P.; Tor, A.J.: *Complexity Reduction In Explicit Linear Model Predictive Control*
The 15-th IFAC World Congress, 2002

[63] Wang, Y.; Boyd, S.: *Fast Model Predictive Control Using Online Optimization*
The 17-th IFAC World Congress, Jul 2008

[64] Wang, Q-G.; He, Y.; Ye, Z.; Lin, Ch; Hang, Ch.Ch.: *On loop phase margins of multivariable control systems*
Journal of Process Control, Volume 18, Issue 2, Feb 2008, Pages: 202-211

[65] Wang, Q-G,; Lin, Ch; Ye, Z.; Wen, G.; He, Y.; Hang, Ch.Ch.: *A quasi-LMI approach to computing stabilizing parameter ranges of multi-loop PID controllers*
Journal of Process Control, Volume 17, Issue 1, Jan 2007, Pages: 59-72

[66] Wang, H.F.; Chen, S.J.; Zhang, P.X.; Cao, Y.J.: *RGA applications in the control of flexible AC transmission systems*
Micro-NanoMechatronics and Human Science, 2005 IEEE International Symposium on, Volume 3, Dec 2003, Pages: 27-30

[67] Welch, G.; Bishop, G.: *An Introduction to the Kalman Filter*
USA, University of North Carolina at Chapel Hill, 2006

[68] Wikipedia contributors: *Model predictive control [online].*
Last update 2009-03-27 [cit. 2009-03-31],
http://en.wikipedia.org/wiki/Model_predictive_control

[69] Wikipedia contributors: *Quadratic programming [online].*
Last update 2009-04-05 [cit. 2009-04-07],
http://en.wikipedia.org/wiki/Quadratic_programming

Appendixs A

Contents of the enclosed CD

For this work the CD with the source codes of described algorithm is enclosed.

- directory *doc*: contains the electronic version of this work
- directory *matlab*: constains the Matlab source codes (m-files) of the described algorithm and Simulink models