INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Short Course on Model Predictive Control

Učební texty k semináři

Autoři:

Prof. Ing. Jan Štecha, CSc. (Czech Technical University in Prague)
Ing. Jaroslav Pekař, Ph.D. (Honeywell Prague Laboratory)

Datum:

14.5. 2010

Centrum pro rozvoj výzkumu pokročilých řídicích a senzorických technologií
CZ.1.07/2.3.00/09.0031

TENTO STUDIJNÍ MATERIÁL JE SPOLUFINANCOVÁN EVROPSKÝM SOCIÁLNÍM FONDEM A STÁTNÍM ROZPOČTEM ČESKÉ REPUBLIKY

Contents

Chapter 1

Introduction

The present age is the age of science and amazing technologies. The progress in the science that has been made during the last several decades has opened future growth opportunities in all areas of our everyday life. We can mention namely computer, medical, physical and chemical science, communication technologies, nanotechnologies, aerospace, but also chemical, petrochemical, machinery and automotive industries, oil refining and many others as excellent examples of areas with the great progress. All these areas require deep understanding of fundamental physical principles and relations. The results of R&D (Research and Development) activities are various technologies, procedures, complex and large-scale systems, machines, production lines, etc. Integral parts of such systems are sensors and actuators. The sensors and actuators are utilized by electronic systems which are used for monitoring and to ensure their accurate, reliable and safe operation during the lifetime.

Model based predictive control (MPC) is a technology offering a systematic approach for controlling the multivariable constrained dynamical systems. MPC technology uses a model of the controlled system to predict the future response. The responses are functions of the system input trajectories, parameters and disturbances. The task for MPC controller is to compute the optimal trajectories for the system inputs which are subject of the control so that the defined objectives for the control loop will be satisfied. In the industrial control, these inputs are known as manipulated variables (MVs). The control objectives are expressed by using a cost function (known also as penalty function). The cost function penalizes undesired behavior of the system over the prediction horizon and it has usually additive form. The individual terms of the sum describes different goals for the controller. The importance of the particular goals are expressed by using the so called weighting coefficients. The next important feature of MPC is the ability to handle the constraints in a nature and systematic manner. These constraints are defined namely by the technological, economical, but also safety restrictions of the controlled system.

The MPC problem is formulated as an optimization problem which has to be solved periodically at each sampling period. This is a real differentiator of MPC when compared to classical control methods. In the efficient MPC formulations, the optimization problem is expressed as a mathematical programming problem which can be solved quite efficiently for a certain class of problems. However, the efficiency of the solvers is still a limiting factor for MPC applications in several areas, especially if the controlled systems are nonlinear and/or are sampled with fast sampling period. The linearity of the controlled system is an important factor when implementing the MPC controller. If the MPC controller is designed for the linear system, all the constraints are linear and the cost function is quadratic, we refer to a linear MPC controller. The linear MPC controller is translated to a quadratic programming problem (QP) for which there exist very efficient solvers. The first class of the most efficient solvers is based on active sets, the second is known as interior point methods.

The things becoming complicated when the system is nonlinear. In this case there are several options how to achieve a good MPC control. The simplest one is just to ignore the nonlinearity at all and to design the controller for a certain operating point of the system. With some luck, the controller will be robust enough and will behave acceptably and we are done (...preferred in the practice). If the nonlinearity cannot be simply ignored, the linear MPC controller in its basic formulation cannot be used and must be extended. The most efficient and systematic solution, at least from the theoretical point of view, is to utilize the fully nonlinear controller. This means that the controller utilizes the nonlinear model of the process to compute the predictions and to handle the constraints. The resulting optimization problem is then nonlinear which may be a challenge to solve in the real-time at each sampling period, as it is required. Another interesting question, when decided for the nonlinear MPC, is how to get a reliable nonlinear model.

It is well known that the success of the MPC application depends on the model quality, i.e. how accurately the model describes the controlled system. Better accuracy means usually better control performances and less complications with the robustness of the solution. The modeling phase in the MPC design procedure plays a very important role. The designer must decide on whether to use the linear or nonlinear approach, select suitable model structure, decide on the model complexity, prepare the identification experiment collect representative data and to fit the model parameters. All these steps

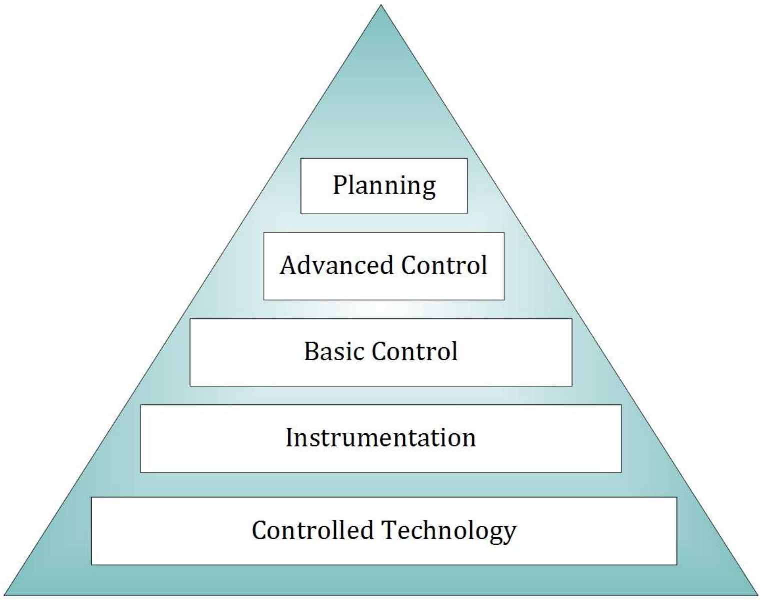

Figure 1.1: Hierarchical structure of a control system

are very important for the successful MPC story.

1.1 Advanced control technologies

Available computer technologies enable implementing the control systems in a wide range of applications. Selection of appropriate control technology and of particular control algorithms is influenced by many factors, e.g. number of actuators, controlled variables (CV), required sampling period, safety and reliability requirements, physical structure of the controlled system, communication limitations, etc. As an example, we can mention a commonly used structure of the control system in the process industry - hierarchical structure. The hierarchical structure is depicted on Fig. 1.1

1.1.1 Instrumentation

This layer represents the basic actuators and sensors of the controlled technology. The number of input/output points depends on the technology but in general, in the process industry, it may be very large (more than several thousands). Typically, it is required to write/read all the values periodically with a time period corresponding to the system character. The measured values are marked by the time stamp and are stored in a process history database which is a very efficient way how to organize the process data.

1.1.2 Basic Control

The basic control layer is usually a core system that ensures the basic functionality and safety operation of the technology. The basic control must be reliable system which often provides a backup solution for the advanced control layer. It contains various technologies to achieve the mentioned goals. If we refer to basic control, we have to think about the technology as a whole and not only about the elemental control loops manipulating the plant actuators. This includes namely the overall control strategy and its hierarchy. It integrates all the basic control modes (manual, automatic, cascade control) but also monitoring and visualization tools. This layer can be seen as the gate for the advanced control and optimization layer because it provides plant prestabilization and reduces the nonlinearities (linearization like effect). The feedback loops are typically implemented by PID controllers.

1.1.3 Advanced Control

The control algorithms which contain some advanced functionality are included in the advanced control layer. These algorithms interact directly with the basic control and perform coordination of individual parts and control loops of basic control strategies. The coordination is usually done through the setpoints, based on master/slave system. Note that the sampling periods in the advanced control layer are slower than in the basic control because the rejection of fast disturbances is a job for the basic control. The main goal is to ensure the optimal operation of the plant under the given conditions which are driven namely by actual technological and economical conditions and by resource restrictions. It is clear that the advanced control algorithms have to work with multivariable systems, must be able to handle prescribed constraints and should ensure the optimal operation. The MPC control technology is therefore an ideal candidate for this position. This layer may contain also the so called real-time optimization (RTO) module. RTO module is usually a model based optimization algorithm that computes the goals for the advanced control of individual plant units or processes. RTO may be static or dynamic and is used for the internal coordination of individual parts of the plant. It is not a surprise, that RTO may be formulated and implemented by an MPC controller.

1.1.4 Planing

The top supervisory layer in the industrial control systems are planing and scheduling. These are usual entry points for the plant technologists and

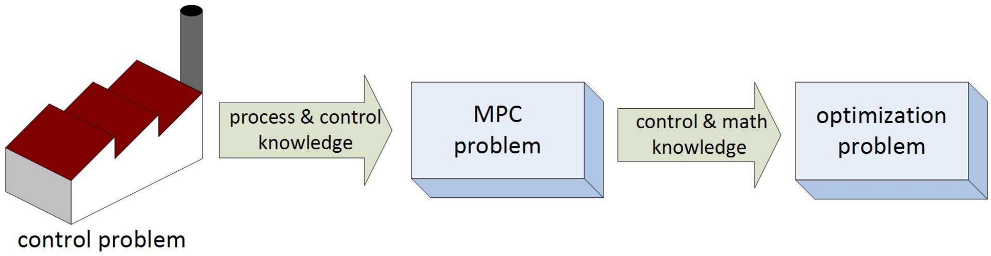

Figure 1.2: A way from the control to an optimization problem

managers. This layer is based on economic-related informations and should provide a complex overview about the plant performance. The main tools here are the databases, visualization tools and specialized computation routines. The planing layer specifies the goals for the advanced control layer in the form of various setpoints, constraints, optimality conditions, resource availability and resource allocation, schedules, etc.

Main enablers for the advanced control and planing tools are the efficient mathematical optimization algorithms and powerful computers which can host the computation routines. The software architecture of the control applications must be very flexible. The solution has to be modular, easily reusable, extendable, but also user friendly. The last mentioned feature is very important and may be a key(!) for success of an advanced control technology.

The MPC methodology should be seen as a tool which enables to delivery the decided goals specified for the controlled process and not as a technology which could replace all the control techniques. The success of applications depends namely on the skills of the application engineers responsible for MPC implementation to a particular process. Translation of the MPC problem to an optimization problem is relatively simple and straightforward, as it will be shown later in this text. The difficult thing may be the formulation of the control problem as a MPC problem. This is very important for practical applications and it requires excellent understanding of the process (from the control point of view) and very good knowledge of MPC, see Fig. 1.2 .

1.2 Classical approach to discrete time dynamic system optimization

In this section three general optimization methods of discrete time dynamic system are presented. In the first part it is variational approach based on mathematical programming. The second general method is discrete maximum principle and the last one is dynamic programming.

1.2.1 Mathematical programming approach to discrete system optimization

Let us have discrete time dynamic system described by state space difference equation

$$x(t+1) = f\big(x(t), u(t), t\big) \ , \quad t = t_0, \ldots, t_1{-}1 \tag{1.1}$$

with initial condition $x(t_0) = x_0$. The problem is to find the control sequence $u(t_0), \ldots, u(t_1{-}1)$, which minimizes the criterion in the form

$$J = h\big(x(t_1)\big) + \sum_{t=0}^{t_1-1} g\big(x(t), u(t), t\big), \tag{1.2}$$

where $(t_1 - t_0)$ is the optimality horizon. It is the problem of mathematical programming - the minimization of the criterion (1.2) with $(t_1 - t_0)$ limiting conditions in the form of equations (1.1). Such problem can be solved using Lagrange vector $\lambda(t)$. Let us define augmented criterion (Lagrangian)

$$\bar{J} = h\big(x(t_1)\big) + \sum_{t=t_0}^{t_1-1} \Big\{ g\big(x(t), u(t), t\big) + \lambda^T(t+1)\Big(f\big(x(t), u(t), t\big) - x(t+1)\Big)\Big\} . \tag{1.3}$$

The Hamiltonian is defined

$$H\big(x(t), u(t), t\big) = g\big(x(t), u(t), t\big) + \lambda^T(t+1) f\big(x(t), u(t), t\big), \tag{1.4}$$

where $t = t_0, \ldots, t_1{-}1$. The Lagrangian can be written in the form

$$\begin{aligned} \bar{J} &= h\big(x(t_1)\big) - \lambda^T(t_1) x(t_1) + H(x(t_0), u(t_0), t_0) + \\ &+ \sum_{t=t_0+1}^{t_1-1} \left\{ H\big(x(t), u(t), t\big) - \lambda^T(t) x(t) \right\} . \end{aligned}$$

In the following simple notation is used

$$\begin{aligned} H(t) &= H\big(x(t), u(t), t\big), \\ g(t) &= L\big(x(t), u(t), t\big), \\ f(t) &= f\big(x(t), u(t), t\big) \ . \end{aligned}$$

If the function $\bar{J}$ is differentiable with respect to $x(t)$ a $u(t)$, the increment of the criterion $\bar{J}$ along the trajectory of the system state and control equals

$$\begin{aligned} d\bar{J} &= \left[\frac{\partial h(t_1)}{\partial x(t_1)} - \lambda^T(t_1)\right] dx(t_1) + \frac{\partial H(t_0)}{\partial x(t_0)} dx(t_0) + \frac{\partial H(t_0)}{\partial u(t_0)} du(t_0) + \\ &+ \sum_{t=t_0+1}^{t_1-1} \left\{ \left[\frac{\partial H(t)}{\partial x(t)} - \lambda^T(t)\right] dx(t) + \frac{\partial H(t)}{\partial u(t)} du(t) \right\}. \end{aligned} \tag{1.5}$$

Vector is always considered as a column vector and the derivative of the scalar function $g(x)$ of vector argument x is a row vector

$$\frac{\partial g}{\partial x} = \left[\frac{\partial g}{\partial x_1}, \ldots, \frac{\partial g}{\partial x_n} \right]$$

and the increment of this function equals

$$dg(x) = \frac{\partial g(x)}{\partial x} dx = \frac{\partial g}{\partial x_1} dx_1 + \cdots + \frac{\partial g}{\partial x_n} dx_n.$$

If the function $g(x, y)$ equals $g(x, y) = y^T A x$, then its derivative equals

$$\frac{\partial g}{\partial x} = y^T A, \qquad \frac{\partial g}{\partial y} = x^T A^T.$$

The derivative of vector function $f(x)$ of vector argument x equals Jacobi matrix

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x} \\ \vdots \\ \frac{\partial f_m}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}.$$

The increment of the state $dx(t)$ caused by increment $du(t)$ follows from (1.1) as

$$dx(t+1) = \frac{\partial f(t)}{\partial x(t)} dx(t) + \frac{\partial f(t)}{\partial u(t)} du(t),$$

Its influence to the criterion can be neglected if the Lagrange coefficients are properly chosen. In formal way the necessary condition of optimum equals $\partial \bar{J}/\partial x(t) = 0$. From this follows the necessary conditions

$$\begin{aligned} \frac{\partial H(t)}{\partial x(t)}^T - \lambda(t) &= 0, \qquad t = t_0 + 1, \ldots, t_1 - 1, \\ \frac{\partial h(t_1)}{\partial x(t_1)}^T - \lambda(t_1) &= 0 \end{aligned}$$

and from the definition of the function $H(t)$ as in (1.4) the difference equation are obtained

$$\lambda(t) = \frac{\partial g(t)}{\partial x(t)}^T + \frac{\partial f(t)}{\partial x(t)}\lambda(t+1) \ , \quad t = t_0, \ldots, t_1{-}1$$

with end condition

$$\lambda(t_1) = \frac{\partial h(t_1)}{\partial x(t_1)}^T .$$

The increment of the criterion (1.5) equals

$$d\bar{J} = \sum_{t=t_0}^{t_1-1} \frac{\partial H(t)}{\partial u(t)} du(t), \tag{1.6}$$

because for fixed initial condition $x(t_0)$ the increment is of course $dx(t_0) = 0$. The expression $\partial H(t)/\partial u(t)$ equals the gradient of the criterion J with respect to control sequence $u(t)$ and with the limitation given by system equation (1.1). The necessary condition for optimal control sequence $u^*(t)$ is zero increment of the criterion (1.6) for arbitrary $du(t)$ in the neighbourhood of $u^*(t)$. The sequence $u^*(t)$ must be stacionary point of the criterion J, so

$$\frac{\partial H(t)}{\partial u(t)} = 0 \ , \qquad t = t_0, \ldots, t_1{-}1.$$

For the solution of discrete optimal control problem it is necessary to find the solution the system of difference equations

$$\begin{aligned} x(t+1) &= \left(\frac{\partial H(t)}{\partial \lambda(t+1)}\right)^T = f(x(t), u(t), t) \ , \qquad (1.7)\\ \lambda(t) &= \left(\frac{\partial H(t)}{\partial x(t)}\right)^T = \left(\frac{\partial g(t)}{\partial x(t)}\right)^T + \frac{\partial f(t)}{\partial x(t)}\lambda(t+1) \ \ , t = t_0, \ldots, t_1{-}1, \end{aligned}$$

with boundary conditions

$$\begin{aligned} x(t_0) &= x_0, \\ \lambda(t_1) &= \left(\frac{\partial h(t_1)}{\partial x(t_1)}\right)^T \qquad (1.8) \end{aligned}$$

and the control $u^*(t)$ given by the condition

$$\frac{\partial H(t)}{\partial u(t)} = \frac{\partial g}{\partial u(t)} + \frac{\partial f(t)}{\partial u(t)}\lambda(t+1) = 0. \tag{1.9}$$

The solution of leads to two point boundary value problem, because initial conditions (1.8) for vector $x(t)$ are given in time $t = t_0$ and initial conditions for λ are given in time $t = t_1$. Both equations (1.7) are coupled by control vector $u(t)$, which is given by equation (1.9). It is the reason for the difficulty of the general problem of optimal control discrete time dynamic system. In this approach there is no limitation in control sequence $u(t)$. Such problem can be completely solved for linear system and quadratic optimality criterion, so called LQ problem.

1.2.2 Maximum principle for discrete time problem

For the optimization of continuous time system the celebrated Pontriagin maximum principle was developed. In such approach the limitation of the control vector can be respected. In analogous way the discrete maximum principle was developed, which is next given without proof. Let us again have state space equations of discrete time system in the form

$$x(t+1) = f(x(t), u(t)) \tag{1.10}$$

where $t \in \mathcal{Z}$ is discrete time, $x(t)$ is state of the system, $u(t)$ is the control which belongs to the limitation set $\mathcal{U}$ and the system is for simplicity time independent. The problem is to find optimal control $u^*(t)$ minimizing the criterion

$$J = h(x(t_1), u(t_1)) + \sum_{t=t_0}^{t_1-1} g(x(t), u(t))dt \tag{1.11}$$

For the solution of such problem the Hamiltonian is formed

$$H(x(t), u(t), p(t+1)) = -g(x(t), u(t)) + p^T(t+1)f(x(t), u(t)) \tag{1.12}$$

The maximum principle states that optimal control maximizes the Hamiltonian, so

$$u^*(t) = \arg\max_{u(t)\in\mathcal{U}} H(x(t), u(t), p(t+1)) \tag{1.13}$$

but only in case if the reachability set $R(z) = \{z : z = f(x, u), u \in \mathcal{U}\}$ is convex for all $x(t)$. The system equation (1.10) is given by

$$x(t+1) = \left(\frac{\partial H(x(t), u(t), p(t+1))}{\partial p(t+1)}\right)^T \tag{1.14}$$

Equation for the so called conjugate system equals

$$p(t) = \left(\frac{\partial H(x(t), u(t), p(t+1))}{\partial x(t)}\right)^T \tag{1.15}$$

For the system (1.10) it is usually known the initial condition $x(t_0)$ and for the conjugate system (1.15) the final condition equals

$$p(t_1) = -\frac{\partial h(x(t_1))}{\partial x(t_1)}.$$

Discrete maximum principle changes the problem of optimal control to two point boundary value problem of the two sets of difference equations and maximization of Hamiltonian with respect to control $u(t)$. Utilizing maximum principle the limitation of control vector $u(t) \in \mathcal{U}$ can be accepted.

1.2.3 Dynamic programming

Dynamic programming, connected with the name R. Bellman, is based on two simple principles. The first one is called principle of optimality. Principle of optimality has different formulation as necessary and sufficient condition. For our case of optimality criterion as (1.11) it is the necessary and sufficient condition. Problem of optimal control is the multistep optimization problem, in each time t in the control interval, optimal control $u^*(t)$ must be chosen.

Principle of optimality states that from arbitrary state $x(t)$ our next decission must be optimal, without respect how the state is reached by previous decissons. It follows from well known proverb "Don't cry on the spilled milk". It is based on obvious fact that you cannot change the past but your future must be controlled in optimal way.

The next principle is the **principle of invariant imbeding**. Single problem can be nested on the whole set of similar problems and solving such set of problems the solution of original problem is obtained.

For discrete time dynamic system (1.10) with initial condition $x(t_0) = x_0$, we are looking for such control sequence $u(t)$ in time interval $t \in T \equiv [t_0, t_1 - 1]$, which minimizes the criterion (1.11) with respect to all limitation of state $x(t) \in X$ and control $u(t) \in U$. Such single problem is imbeded to the whole set of problems of optimal control of dynamic system (1.10) with free initial time which is denoted as $i \in T$ and free initial state which is denoted as $s \in X$. In such case the optimality criterion equals

$$J\left(i, s, u(t_0), \ldots, u(t_1 - 1)\right) = h\left(x(t_1)\right) + \sum_{k=i}^{k_1 - 1} g\left(x(k), u(k), k\right) \tag{1.16}$$

The final time t_1 is fixed. By the solution of the whole set of problems our original problem is solved for $i = t_0$ and $s = x_0$. Let us introduce optimal function $V(s, i)$, which is also called **Bellman function**

$$V(s, i) = \min_{u(i), \ldots u(t_1 - 1)} J\left(i, s, u(t_0), \ldots, u(t_1 - 1)\right) \tag{1.17}$$

Simple modification of previous relation leads to

$$V(s,i) = \min_{u(i)} \left\{ g\left(s,u(i),i\right) + \min_{u(i+1),\dots} \left[h\left(x(t_1)\right) + \sum_{t=i+1}^{t_1-1} g\left(x(t),u(t),t\right) \right] \right\}$$

but the second term in previous relation equals shifted optimal function

$$V(s(i+1),i+1) = V\left(f\left(s,u(i),i\right),i+1\right).$$

Optimal function $V(s,i)$ is the solution of functional recursive equation (**Bellman equation**)

$$V(s,i) = \min_{u(i)} \left\{ g\left(s,u(i),i\right) + V\left(f\left(s,u(i),i\right),i+1\right) \right\}. \tag{1.18}$$

In final time t_1 the optimal function equals

$$V(s,k_1) = h\left(s(k_1)\right) \tag{1.19}$$

which is the boundary condition for Bellman equation (1.18) and $h(s(t_1))$ is the target term of the optimality criterion (1.11). Computation of the optimal function is in principle very simple. The Bellman function $V(s,i)$ is computed backvard in time starting from the final condition (1.19) for all states and in each time step minimization with respect to control $u(i)$ must be solved. There are two problems during the solution of Bellman equation. In general case the grig of states $x(t)$ must be chosen in which optimal function is computed. For great dimension of state vector the grid of states has large dimension which grows exponentially. Such phenomenon called Bellman "curse of dimensionality". Another problem is the necessity to interpolate and extrapolate in the grid of states which make the computation of Bellman function difficult. Using Bellman equation the limitation of system states $x(t) \in \mathcal{X}$ and control $u(t) \in \mathcal{U}$ can be respected. The closed form of the solution of Bellman equation can be obtained in case of quadratic optimal control of linear discrete or continuous time systems.

1.2.4 Quadratic Optimal Control of Linear System

The general results are now used to solve the problem of optimal control linear discrete time system with quadratic criterion, so called **LQ problem**. Stochastic version of such problem is called **LQG problem** (the G is for Gaussian noise in the system state equation). Let us have linear discrete time system

$$x(t+1) = A(t)x(t) + B(t)u(t) \tag{1.20}$$

with initial state $x(0) = x_0$ and optimality criterion in quadratic form

$$J = \frac{1}{2}x^T(t_1)Sx(t_1) + \frac{1}{2}\sum_{t=t_0}^{t_1-1}\left\{x^T(t)Q(t)x(t) + u^T(t)R(t)u(t)\right\}, \tag{1.21}$$

where S is positive semidefinite matrix, $Q(t)$, $t = t_0, \ldots, t_1$ is the sequence of positive semidefinite matrices and $R(t)$, $t = t_0, \ldots, t_1{-}1$ is the sequence of positive definite matrices. The sequence of functions $H(t)$ according to (1.4) equals

$$H(t) = \frac{1}{2}x^T(t)Q(t)x(t) + \frac{1}{2}u^T(t)R(t)u(t) + \lambda^T(t+1)\big(A(t)x(t) + B(t)u(t)\big). \tag{1.22}$$

According to (1.7)–(1.9) the state $x(t)$, costate $\lambda(t)$ and control $u(t)$ is given by the set of difference equations

$$x(t+1) = \frac{\partial H(t)}{\partial \lambda(t+1)} = A(t)x(t) + B(t)u(t)\ , x(t_0) = x_0, \tag{1.23}$$

$$\lambda(t) = \frac{\partial H(t)}{\partial x(t)} = Q(t)x(t) + A^T(t)\lambda(t+1)\ , \lambda(t_1) = Sx(t_1), \tag{1.24}$$

$$0 = \frac{\partial H(t)}{\partial u(t)} = R(t)u(t) + B^T(t)\lambda(t+1). \tag{1.25}$$

From the (1.24) follows

$$u(t) = -R^{-1}(t)B^T(t)\lambda(t+1). \tag{1.26}$$

Because matrices $R(t)$ must be positive definite the existence of its inversion is guaranted. From (1.23) and (1.24) follows

$$\begin{aligned} x(t+1) &= A(t)x(t) - B(t)R^{-1}(t)B^T(t)\lambda(t+1)\ , \quad & x(0) = x_0 \\ \lambda(t) &= Q(t)x(t) + A^T(t)\lambda(t+1)\ , & \lambda(t_1) = Sx(t_1). \end{aligned}$$

Such system can be written in matrix form

$$\begin{bmatrix} x(t+1) \\ \lambda(t) \end{bmatrix} = \begin{bmatrix} A(t) & -B(t)R^{-1}(t)B^T(t) \\ Q(t) & A^T(t) \end{bmatrix} \begin{bmatrix} x(t) \\ \lambda(t+1) \end{bmatrix}.$$

From the solution of this boundary value problem, the optimal control (1.26) is obtained. Previous matrix equation can be written in the form

$$\begin{bmatrix} I & B(t)R^{-1}(t)B^T(t) \\ 0 & A^T(t) \end{bmatrix} \begin{bmatrix} x(t+1) \\ \lambda(t+1) \end{bmatrix} = \begin{bmatrix} A(t) & 0 \\ -Q(t) & I \end{bmatrix} \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} \tag{1.27}$$

and for regular matrix $A(t)$ (if the discrete time system originated from sampling of continuous time system, its matrix $A(t)$ is alvays regular)

$$\begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = \begin{bmatrix} A(t) & 0 \\ -Q(t) & I \end{bmatrix}^{-1} \begin{bmatrix} I & B(t)R^{-1}(t)B^T(t) \\ 0 & A^T(t) \end{bmatrix} \begin{bmatrix} x(t+1) \\ \lambda(t+1) \end{bmatrix}, \tag{1.28}$$

or

$$\begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = \begin{bmatrix} A^{-1}(t) & A^{-1}(t)B(t)R^{-1}(t)B^T(t) \\ Q(t)A^{-1}(t) & A^T(t) + Q(t)A^{-1}(t)B(t)R^{-1}(t)B^T(t) \end{bmatrix} \begin{bmatrix} x(t+1) \\ \lambda(t+1) \end{bmatrix}.$$

Initial condition $\lambda(t_1) = Sx(t_1)$ from (1.24) can be substituted to (1.28) and so

$$\begin{aligned} \begin{bmatrix} x(N-1) \\ \lambda(N-1) \end{bmatrix} &= \begin{bmatrix} A(N-1) & 0 \\ -Q(N-1) & I \end{bmatrix}^{-1} \\ & \begin{bmatrix} I & B(N-1)R^{-1}(N-1)B^T(N-1) \\ 0 & A^T(N-1) \end{bmatrix} \begin{bmatrix} I \\ Q(N) \end{bmatrix} x(N). \end{aligned}$$

From this follows that Lagrange vector $\lambda(t)$ can be expressed in the form

$$\lambda(t) = P(t)x(t),$$

where $P(t)$ is some matrix. From (1.25) follows

$$\begin{aligned} 0 &= R(t)u(t) + B^T(t)P(t+1)x(t+1) \\ &= R(t)u(t) + B^T(t)P(t+1)(A(t)x(t) + B(t)u(t)) \\ &= \left[R(t) + B^T(t)P(t+1)B(t)\right] u(t) + B^T(t)P(t+1)A(t)x(t). \end{aligned}$$

From previous relation optimal control sequence equals

$$\begin{aligned} u(t) &= -\left[R(t) + B^T(t)P(t+1)B(t)\right]^{-1} B^T(t)P(t+1)A(t)x(t) \qquad (1.29) \\ &= -K(t)x(t). \end{aligned}$$

Instead of regularity of the matrix $R(t)$ it is sufficient to fulfill only weaker condition which is the regularity of the matrix $(R(t) + B^T(t)P(t+1)B(t))$. Quadratic optimal control results in linear time dependent state feedback with gain (**Kalman gain**)

$$K(t) = \left[R(t) + B^T(t)P(t+1)B(t)\right]^{-1} B^T(t)P(t+1)A(t). \qquad (1.30)$$

After substitution of this control to (1.24) the following relation is obtained

$$\begin{aligned} P(t)x(t) &= Q(t)x(t) + A^T(t)P(t+1)A(t)x(t) + A^T(t)P(t+1)B(t)u(t) \\ &= Q(t)x(t) + A^T(t)P(t+1)A(t)x(t) - \\ &- A^T(t)P(t+1)B(t)\left[R(t) + B^T(t)P(t+1)B(t)\right]^{-1} \times \\ &\times B^T(t)P(t+1)A(t)x(t). \end{aligned}$$

Such condition is valid for arbitrary state $x(t)$, so the sequence of matrices $P(t)$ is given by matrix difference equation

$$\begin{aligned} P(t) &= A^T(t)P(t+1)A(t) + Q(t) - \qquad (1.31) \\ &- A^T(t)P(t+1)B(t)\left[R(t) + B^T(t)P(t+1)B(t)\right]^{-1} B^T(t)P(t+1)A(t) \end{aligned}$$

with end condition

$$P(t_1) = S. \tag{1.32}$$

Matrix equation (1.31) is called **Riccati Difference Equation**. Riccati difference equation can be written in the form

$$P(t) = \big(A(t)-B(t)K(t)\big)^T P(t+1)\big(A(t)-B(t)K(t)\big)+K^T(t)R(t)K(t)+Q(t), \tag{1.33}$$

From previous relation follows that if $S = S^T \geq 0$ then all matrices $P(t)$ are also symmetrix and positive semidefinite.

Steady state solution of Riccati equation

Let us have time invariant discrete time system

$$x(t+1) = Ax(t) + Bu(t) \tag{1.34}$$

and optimality criterion

$$J = \frac{1}{2}x^T(t_1)Sx(t_1) + \frac{1}{2}\sum_{t=t_0}^{t_1-1}\left\{x^T(t)Qx(t) + u^T(t)(t)u(t)\right\}, \tag{1.35}$$

where S and Q are positive semidefinite constant matrices and R is positive definite constant matrix. Optimal control $u^*(t)$ minimizing previous criterion results in linear state feedback

$$u(t) = -K(t)x(t), \tag{1.36}$$

It can be shown that for increasing optimality horizon ($t_1 \to \infty$) Kalman gain $K(t)$ is approaching to constant matrix K and also matrix $P(t)$ given by Riccati equation is approaching to constant matrix P. Such constant matrix P is the silution of **Algebraic Riccati Equation**

$$P = A^TPA - A^TPB\left[R + B^TPB\right]^{-1}B^TPA + Q, \tag{1.37}$$

Such algebraic equation results from matrix Riccati equation if $P = P(t) = P(t+1)$. Linear optimal feedback gain (Kalman gain) equals

$$K = \left[R + B^TPB\right]^{-1}B^TPA. \tag{1.38}$$

Two question must be answered. If the limiting condition of Riccati equation eqists ($\lim_{(t_1-t)\to\infty} P(t) = P$) and equals to the solution of algebraic Riccati equation P and if is stable the feedback optimal system with state matrix $(A - BK)$. Both conditions are fulfilled if the controlled system (1.34) is stabilizable (couple (A, B) is stabilizable) and the system with output matrix C_Q is detectable (couple (C_Q, A) is detectable), where $C_Q^TC_Q = Q$.

1.3 Brief overview of Model Predictive Control

Model predictive control (MPC) is one of the methods that enables optimal control of constrained multivariable dynamical systems. It is interesting to note that MPC has been (re)discovered and widely used by the industrial practitioners before having solid theoretical base. The first known formulation of a moving horizon controller using a linear programming is in [28] and probably first description of MPC control application in [33]. At the present time, MPC is a standard advanced control technology used in the process industry. For the computational limit reasons, the practical MPC applications have been limited to the linear models which may not be accurate enough for some classes of applications. For such classes of systems, it is beneficial to define the nonlinear MPC which requires solution of a constrained nonlinear optimization problem at each sampling period. In nonlinear MPC, the optimization problems are usually solved by using *Sequential Quadratic Programming*.

In the standard formulation of the model predictive control, the optimization problem (usually quadratic program) is solved in each sampling instance and therefore time consumption caused by computation of control action can be significant. It was shown that a linear [5] or quadratic [6] program can be solved explicitly off-line and then the control action is generated by piecewise affine function of the system state. Such an optimization is known as *Multi-Parametric Programming* [18]. This approach is useful namely in the case when we want to use MPC for control the relatively fast sampled, but small, systems.

Another actual topic in the model predictive control is hierarchical, decentralized and distributed control [31, 36, 19, 1]. Today's communication technologies enables coordinated control of large-scale systems, e.g. energy or water distribution networks [29]. The large-scale system controller can be formulated and implemented as a central algorithm which enables the overall optimality. The centralized optimal control enables significant savings during the system lifetime. But there may be problems with centralized controllers, e.g. too large optimization problems, communication limitations, safety reasons, etc. It is very practical to decompose the central optimal controller so that there will be smaller controller for all important subsystems (slave controllers) which are coordinated by a coordinator (master problem).

Currently, the number of successful MPC applications and research work grows rapidly. Some MPC survey papers are [30, 7, 22, 2]. The standard books about the linear MPC are [34, 21, 14] and about non-linear MPC, for example the set of important papers [25]. A real-time iteration scheme for the nonlinear MPC has been proposed in [16]. We can found a number of practical commercial MPC technologies in the literature. A survey of industrial MPC technologies can be found in [30].

Chapter 2

Linear Model Predictive Control

This chapter deals with the fundamentals of the linear model predictive control. The linear formulation is very popular and often used in the practical applications when compared to the nonlinear MPC. The reason is clear - the simplicity. The advanced control technologies, for which the MPC is usually used, are sitting at the top of the basic control strategy. Therefore, there is no need to cover all the operating regimes of the controlled system, especially the emergency states, system startup or shutdown, etc. The objective for the MPC controller in the advanced control layer is to optimize the performances while satisfying the constraints. Of course, there are applications, where the MPC controller is used as the basic controller, directly in the basic control layer. In this case, we can enjoy all the advantages which are offered by the MPC, namely the ability to control the multivariable systems in a very systematic manner, to handle the constraints and to provide the optimal control solution.

2.1 Motivation example

Before starting the formal definitions of MPC, we will give a motivation example, which illustrates how simple the MPC controller may be. Assume a stable linear system with one output and one input. The system is periodically sampled.

- **System model and predictions:** The relation between the system input and output can be approximated based on the truncated step response as

$$y(k) \cong y_0 + \sum_{i=0}^{n} h_i \Delta u(k-i) \ , \tag{2.1}$$

where $y(k)$, $\Delta u(k) = u(k) - u(k-1)$ and h_i are the system output, increments of the system input and the step response coefficients respectively. Then, the prediction at the discrete time $k+j$ is given by (model base predictions)

$$y(k+j|k) \cong y_0 + \sum_{i=0}^{n} h_i \Delta u(k+j-i|k) \ . \tag{2.2}$$

It is easy to show, by using (2.2), that the system output prediction over the prediction horizon of N samples can be expressed as

$$Y_k^{k+N} = 1y_0 + H_1 \Delta U_{k-n}^{k-1} + H_2 \Delta U_k^{k+N} \ , \tag{2.3}$$

where the matrices H_1 and H_2 contain the step response coefficients, Y_k^{k+N}, ΔU_k^{k+N} are the system output predictions and system input increment predictions, ΔU_{k-n}^{k-1} are the past system input increment values, i.e.

$$\begin{aligned}
Y_k^{k+N} &= \begin{bmatrix} y(k) & y(k+1) & \dots & y(k+N) \end{bmatrix}^T \ , \\
\Delta U_k^{k+N} &= \begin{bmatrix} \Delta u(k) & \Delta u(k+1) & \dots & \Delta u(k+N) \end{bmatrix}^T \ , \\
\Delta U_{k-n}^{k-1} &= \begin{bmatrix} \Delta u(k-n) & \Delta u(k-n+1) & \dots & \Delta u(k-1) \end{bmatrix}^T \ .
\end{aligned}$$

Note that the term $H_1 \Delta U_{k-n}^{k-1}$ in (2.3) is related to the initial system state response, known also as autonomous or unforced response. In practical applications, due to disturbances and model uncertainty, this term must be replaced by a known function of the system state.

- **Control problem and MPC formulation:** Now, for example, we would like the system to follow the given reference signal R_k^{k+N} which trajectory is known in advance, over the whole prediction horizon. In other words, the system output should be as close to the reference signal as possible (first control goal), with reasonable control action (second control goal). These requirements can be expressed by defining the tracking error

$$E_k^{k+N} = R_k^{k+N} - Y_k^{k+N} = R_k^{k+N} - 1y_0 - H_1 \Delta U_{k-n}^{k-1} - H_2 \Delta U_k^{k+N} \tag{2.4}$$

  and by minimizing the cost function, which can be defined, for example, as a sum of weighted second norms [1] (MPC problem)

$$J\left(\Delta U_k^{k+N} | R_k^{k+N}\right) = \left\| E_k^{k+N} \right\|_{Q_E}^2 + \left\| \Delta U_k^{k+N} \right\|_{Q_{\Delta U}}^2 \ . \tag{2.5}$$

- **Resulting optimization problem:** The system input trajectory on the prediction horizon can be determined by solving the optimization problem

$$\Delta U_k^{*k+N} = \arg \min_{\Delta U_k^{k+N}} J\left(\Delta U_k^{k+N} | R_k^{k+N}\right) \ , \tag{2.6}$$

  which can be viewed as a simple linear least squares problem. The solution can be found explicitly and the result will be a linear function of the system state and external parameters.

[1] The first control goal (reference tracking) corresponds to the first term in the criterion, the second control goal (actuator activity) to the second one.

The example illustrated basic idea behind the MPC, i.e. using the system model to predict the future behavior, formulation of the control goals, transformation of the control goals as MPC control problem and transformation of the MPC problem to an optimization problem.

2.2 Formulation of linear MPC

A great number of systems and processes work in a steady-state or close to an operating point. It is well known that behavior of systems under such conditions can be usually well approximated by a linear model. In this section, we will formulate and analyze the basic linear MPC controller. The main components of MPC are the following: system model, cost function, constraints and resulting optimization problem.

2.2.1 System models in MPC

In the introductory chapter, we formulated a simple MPC algorithm based on predictions from the step response of the system. A similar prediction model can be derived by using the impulse response. Generally, in the linear MPC, we can use any linear model. In addition to step and impulse based prediction models, we will discuss only ARX and state space models. The modeling stage in MPC design is one of the most important things. The quality of the resulting controller is proportional to the model quality and therefore the model should be as accurate as possible.

Impulse response

The relation between the system input and output can be described by equation

$$y(k) = \sum_{i=0}^{\infty} g_i u(k-i) \ , \tag{2.7}$$

which is known as convolution or weighting sequence model and $y(k)$, $u(k)$, g_i are the system output, input and coefficients of the impulse response, respectively. The model can be used only for stable systems with the finite impulse response (FIR). As it has been shown in the introductory MPC example, the response is truncated and only n most important coefficients are used. Therefore, the prediction model can be described by relation

$$\widehat{y}(k+j|k) = \sum_{i=0}^{n} g_i u(k+j-i|k) \ . \tag{2.8}$$

The advantages of the impulse and step response models are that we do not need to know any prior information about the system (of course, it must be stable and the responses must be finite), i.e. the system is a "black box". These models can be also used for the multivariable systems for which we have

$$y_m(k) = \sum_{l=1}^{p} \sum_{i=0}^{n} g_i^{l,m} u^l(k-i) \;, \tag{2.9}$$

where $y_m(k)$ is the m-th system output, p is number of system inputs, $u^l(\cdot)$ is l-th input and $g_i^{l,m}$ is the sequence of impulse response of l-th input to m-th system output. It is interesting to note, that the first and a number of current practical MPC implementations are based on step or impulse response models.

ARX based models

The ARX based models are popular in the control society because they enable to describe also the basic stochastic properties of the systems. Basic form of the ARX model with a measurable disturbance is

$$y(k) + \sum_{i=1}^{n_a} a_i y(k-i) = \sum_{i=0}^{n_b} b_i u(k-i) + \sum_{i=0}^{n_d} d_i v(k-i) + e(k) \;, \tag{2.10}$$

where $y(k)$, $u(k)$, $d(k)$ and $e(k)$ are system output, input, disturbance and white noise. The resulting prediction model can be written in a vector form

$$\vec{y} = A_p^{-1} \left(-A_t \tilde{y} + B_t \tilde{u} + D_t \tilde{v} + B_p \vec{u} + D_p \vec{v} \right) \;, \tag{2.11}$$

where the matrices are given by the ARX model coefficients

$$[A_t|A_p] = \left[\begin{array}{ccccc|cccccccc} a_n & . & . & . & a_1 & 1 & 0 & . & . & . & . & 0 \\ 0 & a_n & . & . & . & a_1 & 1 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & 0 & a_n & . & . & a_1 & 1 & 0 & . & 0 \\ 0 & . & . & . & 0 & a_n & . & . & a_1 & 1 & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & 0 & . & . & a_n & . & . & a_1 & 1 \end{array}\right] ,$$

$$[B_t|B_p] = \left[\begin{array}{ccccc|ccccccc} b_n & . & . & . & b_1 & b_0 & 0 & . & . & . & . & 0 \\ 0 & b_n & . & . & . & b_1 & b_0 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & 0 & b_n & . & . & b_1 & b_0 & 0 & . & 0 \\ 0 & . & . & . & 0 & b_n & . & . & b_1 & b_0 & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & 0 & . & . & b_n & . & . & b_1 & b_0 \end{array}\right] ,$$

$$[D_t|D_p] = \left[\begin{array}{ccccc|ccccccc} d_n & . & . & . & d_1 & d_0 & 0 & . & . & . & . & 0 \\ 0 & d_n & . & . & . & d_1 & d_0 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & 0 & d_n & . & . & d_1 & d_0 & 0 & . & 0 \\ 0 & . & . & . & 0 & d_n & . & . & d_1 & b_0 & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & 0 & . & . & d_n & . & . & d_1 & d_0 \end{array}\right] .$$

The prediction vectors are

$$\begin{aligned}
\vec{y} &= \begin{bmatrix} \widehat{y}(k) & \cdots & \widehat{y}(k+N-1), \end{bmatrix}^T , \\
\vec{u} &= \begin{bmatrix} u(k) & \cdots & u(k+N-1), \end{bmatrix}^T , \\
\vec{v} &= \begin{bmatrix} v(k) & \cdots & v(k+N-1), \end{bmatrix}^T ,
\end{aligned}$$

and the vectors of past output, input and disturbance values are

$$\begin{aligned}
\tilde{y} &= \begin{bmatrix} y(k-n_a) & \cdots & y(k-1) \end{bmatrix}^T , \\
\tilde{u} &= \begin{bmatrix} u(k-n_a) & \cdots & u(k-1) \end{bmatrix}^T , \\
\tilde{v} &= \begin{bmatrix} v(k-n_a) & \cdots & v(k-1) \end{bmatrix}^T .
\end{aligned}$$

State space model

The state space models are important for the MPC. The reason is that they provide description of multivariable systems and are also important for the analysis. Another advantage is that the state space model can be used also for systems with integrators and unstable systems. A basic form can be written as

$$\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) , \\
y(k) &= Cx(k) + Du(k) .
\end{aligned}$$

The prediction trajectories of the system output are given by

$$\vec{y} = \bar{P}x(k) + \bar{H}\vec{u} , \tag{2.12}$$

where $x(k)$ is the initial state. Vectors $\vec{y}$, $\vec{u}$ and matrices $\bar{P}$, $\bar{H}$ are

$$\begin{aligned}
\vec{y} &= \begin{bmatrix} y(k)^T & y(k+1)^T & \cdots & y(k+N-1)^T \end{bmatrix}^T , \\
\vec{u} &= \begin{bmatrix} u(k)^T & u(k+1)^T & \cdots & u(k+N-1)^T \end{bmatrix}^T ,
\end{aligned}$$

$$\bar{P} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix} , \qquad \bar{H} = \begin{bmatrix} D & & & \\ CB & D & & \\ \vdots & & \ddots & \\ CA^{N-2}B & \cdots & CB & D \end{bmatrix} .$$

Prediction equations for the system state are given by

$$\vec{x} = Px(k) + H\vec{u} , \tag{2.13}$$

where vector $\vec{x}$ and matrices P and H are

$$\vec{x} = \begin{bmatrix} x(k+1)^T & x(k+2)^T & \ldots & x(k+N)^T \end{bmatrix}^T ,$$

$$P = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \qquad H = \begin{bmatrix} B & & & \\ AB & B & & \\ \vdots & & \ddots & \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}.$$

$\succ$

2.2.2 Cost function

The cost function is used to formulate goals for the MPC controller. It has usually additive form where the individual terms express various control requirements. The terms are multiplied by factors defining the relative importance of the control goals. A basic requirement is reference tracking. The corresponding cost function term penalizes the tracking error over a given prediction horizon. The second basic term is a term that specifies actuator behavior. Therefore, the standard cost function has the following form

$$J\left(\vec{u}|x(t_0), t_0\right) = \sum_{i=0}^{N} \left\|Q_p e(t_0 + t_i|t_0)\right\|_p + \sum_{j=0}^{N_u-1} \left\|R_p u(t_0 + \tau_j|t_0)\right\|_p . \quad (2.14)$$

where $t_0 + t_i$ are the sampling times of predicted trajectory, $t_0 + \tau_i$ are the sampling times of the system input, matrices $Q_p \geq 0$, $R_p > 0$ are weighting matrices, $e(t) = r(t) - y(t)$ is the difference between the system output and reference signal (tracking error), $u(t)$ is the system input, $x(t_0)$ is the initial information (does not necessarily be the system state) and $\vec{u} = \{u(t_0 + \tau_0|t_0), \ldots, u(t_0 + \tau_{N_u-1}|t_0)\}$ is the set of future control actions. The l_p norm of a vector x of length n is defined as

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^{n} |x_i|^p} .$$

The cost function in the form (2.14) is suitable for integrating systems. Another cost function which is very often used in practical applications penalizes movements of actuators instead of penalizing the positions. MPC based on a such cost function are referred to as minimum movement controllers. For example

$$J\left(\vec{u}|x(t_0), t_0\right) = \sum_{i=0}^{N} \left\|Q_p e(t_0 + t_i|t_0)\right\|_p + \sum_{j=0}^{N_u-1} \left\|R_p \Delta u(t_0 + \tau_j|t_0)\right\|_p , \quad (2.15)$$

where

$$\Delta u(t_0 + \tau_j|t_0) = u(t_0 + \tau_j|t_0) - u(t_0 + \tau_{j-1}|t_0) .$$

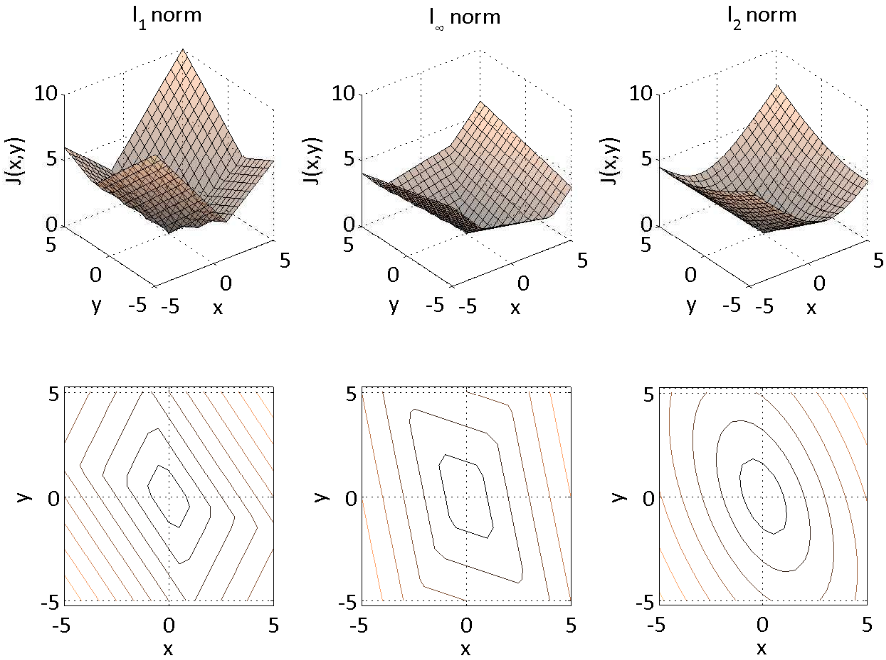

Figure 2.1: Cost functions examples - based on l_1, l_∞ and l_2 norm.

It is clear that we can introduce various terms and cost functions in general. For example probability terms for stochastic system, nonlinear terms, etc. Because the cost function is optimized, we have to be careful when adding the terms. It has to be prepared so that there exist some reliable optimization method for the resulting optimization problem. The penalty functions (2.14) and (2.15) are based on a general p-norm but only l_1, l_∞ and namely l_2 norms are used in the practical applications[2], see Fig. 2.1 and Fig. 2.2 [26]. In the linear MPC, utilization of l_1 and l_∞ leads to *Linear Programming* (LP) and utilization of l_2 norm leads to *Quadratic Programming* (QP). The quadratic norm ensures good performances of the control loop, as we know from the classical LQR[3] controller.

2.2.3 Constraints

A real differentiator for the MPC controllers is the fact that they can handle the system constraints in a straightforward manner. All processes have some constraints, e.g. actuator position and rate of change constraints or constraints for the system output or any internal state

$$u_{\min}(t) \le u(t) \le u_{\max}(t) \ ,$$

$$\Delta u_{\min}(t) \le \Delta u(t) \le \Delta u_{\max}(t) \ .$$

[2] Note that utilization of l_1 norm leads to the dead beat control.

[3] (Linear Quadratic Regulator)

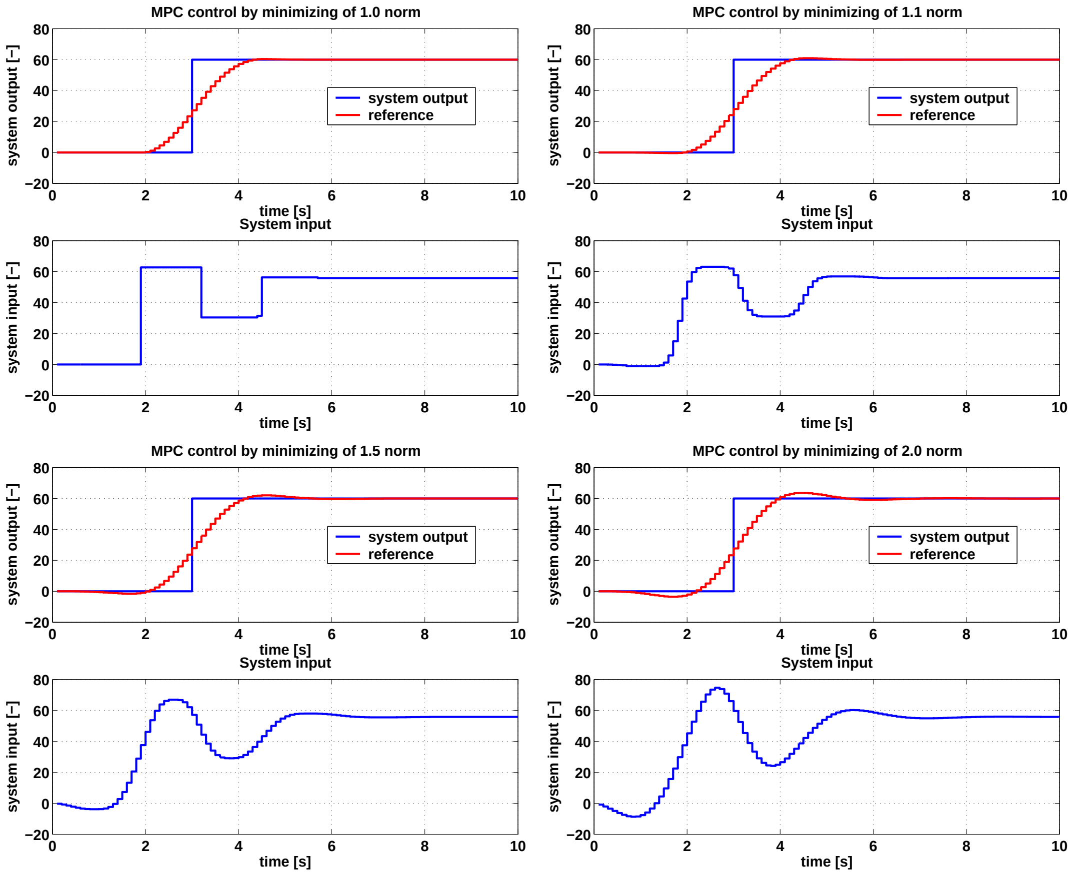

Figure 2.2: An example of unconstrained MPC control for different l_p norms in the cost function (l_1, $l_{1.1}$, $l_{1.5}$ and l_2).

$$y_{\min}(t) \leq y(t) \leq y_{\max}(t) \ ,$$
$$x_{\min}(t) \leq x(t) \leq x_{\max}(t) \ .$$

In general, the constraints are hard or soft:

- Hard constraints - physical limitations of real process, e.g. actuator extreme positions. This type of constraints must not be violated.
- Soft constraints [35] - these can be violated though at some penalty, for example a loss of product quality.

The soft constraints are used whenever there are some disturbances acting directly on the constrained variable, typically all the system states and outputs. They are very important for all practical implementations because the soft constraints ensure the feasibility of the MPC optimization problem. The soft constraints can be formulated by introducing a slack optimization variable or vector. Assume, for example, an upper limit for the system output, then

$$y(t) \leq y_{\max} + \varepsilon$$

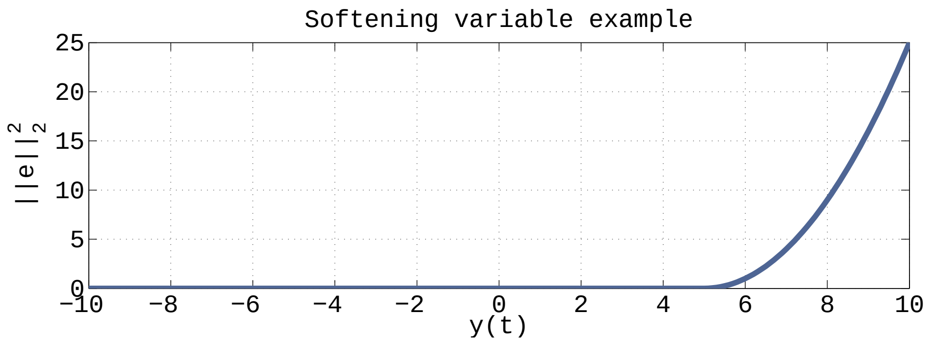

Figure 2.3: Example of soft constraints: $y(t) \leq 5 + \varepsilon$

is referred to as soft constraint. The variable ε is a scalar variable. to finish definition of the soft constraint, we have to introduce term $\|\varepsilon\|_2^2$ into the cost function. The soft constraint may be violated, especially during the transients. Therefore, the weighting factor for the soft constraints must be high enough (relative to other terms) to ensure small violation only. We can have a common scalar slack variable for all soft constraints, we can have a slack variable for each soft constraint or for a subset of soft constraints. Another possibility how to formulate the soft constraints is to penalize the constraints violation directly in the cost function, i.e.

$$\varepsilon \leq y_{\max}$$

and to introduce the term $\|y(t) - \varepsilon\|_2^2$ into the criterion function. The result will be exactly the same as in the first formulation. There will be differences in the QP form structure. Note also that the second formulation introduces box constraints, which may be beneficial for the efficiency of the optimization algorithm. The soft constraints may be seen also as non-symmetric penalty (see Fig. 2.3).

2.2.4 Optimization problem

We have shown that the prediction of a linear system behavior can be expressed by the affine function of the system inputs by using any linear model. Therefore we can focus on the state space models without any restrictions because other linear models can be transformed to this form. The basic MPC control problem can be formulated as an optimization problem

$$\vec{u}^* = \arg\min_{\vec{u}} \; J\left(\vec{u}|x(t_0), t_0\right) \tag{2.16}$$

subject to

- input constraints

$$\begin{aligned} u_{\min}(t_0+t_i) \leq \quad & u(t_0+t_i) & \leq u_{\max}(t_0+t_i) \\ \Delta u_{\min}(t_0+t_i) \leq \quad & \Delta u(t_0+t_i) & \leq \Delta u_{\max}(t_0+t_i) \end{aligned}$$

- output constraints (usually softened)

$$y_{\min}(t_0+t_i) \leq y(t_0+t_i) \leq y_{\max}(t_0+t_i)$$

- system state constraints (usually softened)

$$x_{\min}(t_0+t_i) \leq x(t_0+t_i) \leq x_{\max}(t_0+t_i)$$

- system model equations

$$\begin{aligned} x(t_{i+1}) &= Ax(t_i) + Bu(t_i) \,, \\ y(t_i) &= Cx(t_i) + Du(t_i) \,. \end{aligned}$$

The optimization problem (2.16) with all the constraints defines the MPC problem. When using the l_2 norm in the cost function, the MPC problem for a linear system with the linear constraints can be transformed to a mathematical programming problem of the form

$$\vec{u}^* = \arg\min_{\vec{u}} \frac{1}{2}\vec{u}^T H \vec{u} + \vec{u}^T F \vec{p} \,, \quad s.t. \quad G\vec{u} \leq W + S\vec{p} \,, \tag{2.17}$$

which is a well known quadratic programming problem. $\vec{u}$ is a vector of optimal input trajectories

$$\vec{u} = \left[\begin{array}{cccc} u^T(t_0) & u^T(t_1) & \dots & u^T(t_N) \end{array} \right]^T \,,$$

$\vec{p}$ is the parameter vector containing, for example, system initial state $x(t_0)$, reference signal trajectories, etc. The matrices H and F can be found by using definition of the criterion function (2.14) and the prediction equations for the state space model ((2.12) and (2.13)).

2.2.5 Receding horizon control

It has been shown that the MPC control problem can be transformed to an optimization problem (2.17) which is parameterized by a parameter vector $\vec{p}$. The result of the optimization problem at time t_0 is the optimal future trajectory of the system input $\vec{u}^*$. An immediate idea would be to apply the whole sequence and to compute the new trajectories at the end of prediction horizon, i.e. at time t_N. Such MPC control strategy corresponds to the open loop control, without any feedback during the prediction horizon. It is clear that the open loop control is not able to reject the disturbances acting on the system and therefore such strategy is not practical.

The standard feedback, as we know it for the classical control methods, is introduced by using so called *Receding Horizon Control*. In the receding horizon control, the optimization problem (2.17) is computed at each sampling period after having new system measurements or estimates and we apply only the first control action from the vector $\vec{u}^*$. This strategy ensures the standard feedback control in the MPC. Note that MPC is sometime called directly as receding horizon control.

2.2.6 Blocking strategies

It is clear that the receding control strategy increases on-line complexity of the controller. We need to solve the optimization problem at each sampling period, which may not be practical especially for large-scale systems or systems with fast sampling period. In some situations, we need to used MPC even for such applications and therefore we have to reduce the on-line computation complexity. In the standard MPC formulation, the number of optimization variables corresponds to the number of manipulated variables multiplied by the prediction horizon length. The degrees of freedom is one of the dominant factors of the MPC optimization problem. We can reduce the degrees of freedom by fixing the manipulated variables to be constant over several sampling periods. This strategy is known as blocking [13] and is used by many practical implementations. Extreme blocking would be to enable only one change over the whole prediction horizon, i.e. the system inputs can do a step change at the beginning of the prediction horizon and remain at the new position over the rest of prediction horizon. Such strategy is known as mean control and its property is that the closed loop response is comparable to or slower that the open loop response. This is not a problem in a number of practical applications.

2.2.7 Offset-free tracking

In the classical control methods, the offset-free tracking control is achieved by intruding the integral action to the controller. It is clear that if the MPC controller uses a perfect model and there are no disturbances acting on the system, we will not need to use any additional mechanism to achieve the offset-free tracking, but this is not a realistic assumption. The integral action usually acts on the tracking error. The question is, how to achieve the offset-free tracking property in the model predictive control. There are several possibilities but we will mention only the two most important from the practical point of view. The first option is to introduce the integral term acting on the tracking error into the cost function. This approach copies strategy from the standard PID control and requires implementation of an anti-windup mechanism which may be impractical.

The second approach is based on assumption that there are virtual disturbance variables acting on the system. These virtual disturbances covers the real disturbances, but also model inaccuracy. This approach has been utilized successfully by many industrial MPC applications [23]. It is usually assumed that the virtual disturbances are constant over the prediction horizon. The disturbances can be estimated by using the augmented system state observer, These techniques are known as *Unknown Input Observer.*

The virtual disturbances can be connected to the system in a number of ways [23]. Furthermore, the disturbances may be described by a general linear model. We will show the simplest three examples. Consider a linear model of a controlled process

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \ , \\ \widehat{y}(k) &= Cx(k) + Du(k) \ . \end{aligned}$$

Assume that the disturbance model can be described by the autonomous linear model of the form

$$\begin{aligned} x_d(k+1) &= A_d x_d(k) \ , \\ d(k) &= C_d x_d(k) \ . \end{aligned}$$

- **Disturbance acting on the system output:** In this case, it is assumed that the real system output is given by $y(k) = \widehat{y}(k) + d(k)$. The augmented system model has the form

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ x_d(k+1) \end{bmatrix} &= \begin{bmatrix} A & 0 \\ 0 & A_d \end{bmatrix} \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix} + Du(k) \end{aligned}$$

- **Disturbance acting on the system state:** In this case, the disturbance is assumed to act directly on the system state, i.e.

$$\begin{aligned}
\begin{bmatrix} x(k+1) \\ x_d(k+1) \end{bmatrix} &= \begin{bmatrix} A & C_d \\ 0 & A_d \end{bmatrix} \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) \\
y(k) &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix} + Du(k)
\end{aligned}$$

- **Disturbance acting on the system input:** In this case, the disturbance is connected to the system input, i.e.

$$\begin{aligned}
\begin{bmatrix} x(k+1) \\ x_d(k+1) \end{bmatrix} &= \begin{bmatrix} A & BC_d \\ 0 & A_d \end{bmatrix} \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) \\
y(k) &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix} + Du(k)
\end{aligned}$$

It is clear that we can introduce other virtual disturbance models to achieve the offset-free tracking. The state of the augmented system model is estimated by a suitable observer, e.g. by a Kalman Filter. The final closed loop performance is directly related to the accuracy of the virtual disturbance model structure. In fact, it is not possible to find a good disturbance model for all applications and therefore, the choose of this model can be seen as an additional tuning parameter for the MPC controller. The practical applications are often using constant disturbance, i.e. $A_d = I$.

2.3 Analysis of linear MPC

The classical feedback controllers (PID) can be analyzed in a number of ways. The most important properties are the nominal performance, stability and robustness. In this section, we will show that a similar analysis can be done for the MPC controller. The difference between the classical control and MPC is that the MPC computes directly the sequence of the control actions instead of using a control law[4] which generates the control action. In fact, the optimization problem could be seen as a control law. Why the MPC controller cannot be simple analyzed as a classical controller, e.g. PID? The answer is - due to presence of constraints. If we would have a control law as a result of the optimization problem, we could perform the standard analysis. It can be shown, that we can find a control law for each combination of

[4] A control law in the linear control is an affine function of the system state, e.g. u(k)=Kx(k)+g.

feasible active constraints in the form

$$u(k) = K_i x(k) + g_i \ , \tag{2.18}$$

where the index i is used to denote i-th set of feasible active constraints.

2.3.1 Unconstrained MPC

In this section, we will show how we can derive the control law for the case when there are no active constraints. Note that we can derive a control law for any feasible combination of active constraints in a similar way. Assume that the system can be described by a state space model. Then the prediction model for a prediction horizon of length N is given by (2.12), i.e.

$$\vec{y} = \bar{P}x(k) + \bar{H}\vec{u} \ . \tag{2.19}$$

Further, assume a quadratic cost function defining the tracking MPC problem. The basic cost function is therefore given by

$$J(\vec{u}|x(k), k) = (\vec{r} - \vec{y})^T Q (\vec{r} - \vec{y}) + \vec{u}^T R \vec{u} \ .$$

By definition of MPC, the control action is obtained by minimizing the cost function over the prediction horizon. In our case without the constraints, the optimal input trajectory $\vec{u}^*$ can be found by solving a simple least squares problem. By using (2.19), the optimal control problem is

$$\min_{\vec{u}} J(\vec{u}|x(k), k) = \left(\vec{r} - \bar{P}x(k) - \bar{H}\vec{u}\right)^T Q \left(\vec{r} - \bar{P}x(k) - \bar{H}\vec{u}\right) + \vec{u}^T R \vec{u} \ ,$$

with solution

$$\vec{u}^* = \left(\bar{H}^T Q \bar{H} + R\right)^{-1} \bar{H}^T Q \left(\vec{r} - \bar{P}x(k)\right) \ . \tag{2.20}$$

As a result, we obtained a control sequence over the prediction horizon that is parameterized by the system state at discrete time k and by the sequence of future reference signal. By applying the receding horizon control strategy, we will get a control law in the form

$$u(k)^* = -K^x x(k) + K^r \vec{r} \ , \tag{2.21}$$

where the matrices K^x and K^r are given by the first n_u rows of (2.20) and n_u is the number of system inputs. Having the control law (2.21), we can do the basic analysis of the controller.

Numerical example

Assume the unconstrained MPC tracking problem for the linear system described by

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 1 \end{bmatrix}, \; D = 0 \,.$$

The corresponding cost function is defined as

$$J(\vec{u}|x(k), k) = (\vec{r} - \vec{y})^T \, Q \, (\vec{r} - \vec{y}) + \Delta\vec{u}^T R \Delta\vec{u}$$

where we used penalty for $\Delta\vec{u}$. Then the control law obtained from the least squares solution has the form

$$u(k) = -K^x x(k) + K^r \vec{r} + K^u u(k-1) \,.$$

If we use the prediction horizon $N = 10$, weighting matrix for the tracking error $Q = I$ and penalty on the input movement $R = k_r I$, then we will get for $k_r = 100$

$$\begin{aligned} K^x &= -\begin{bmatrix} 0.0673 & 0.3083 \end{bmatrix} \,, \\ K^r &= \begin{bmatrix} 0.0070 & 0.0111 & \dots & -0.0047 \end{bmatrix} \,, \\ K^u &= 0.4650 \,. \end{aligned}$$

Now we can study the controller behavior. For example, we might be interested in the influence of the tuning parameter k_r which multiplies the penalty of the actuator movements. The simulation is depicted on Fig. 2.4, the step response of the closed loop is depicted on Fig. 2.5 and the frequency response on Fig. 2.6.

2.3.2 Infinite prediction horizon

It is known that the infinite horizon LQR control ensures reasonable stability margins and reasonable control performance. The disadvantage is that it does not enable to handle the constraints in a systematic way. Basic version of MPC controller is based on a finite prediction horizon. We can extend the prediction horizon to infinity by defining the cost function

$$J\left(u(0), \dots, u(\infty)\right) = \sum_{k=0}^{\infty} \left(x(k)^T Q x(k) + u(k)^T R u(k) \right) \tag{2.22}$$

Now we can split the infinite prediction horizon into the two parts, as follows[5]

[5]This approach is also known as a *Dual Mode MPC Control*.

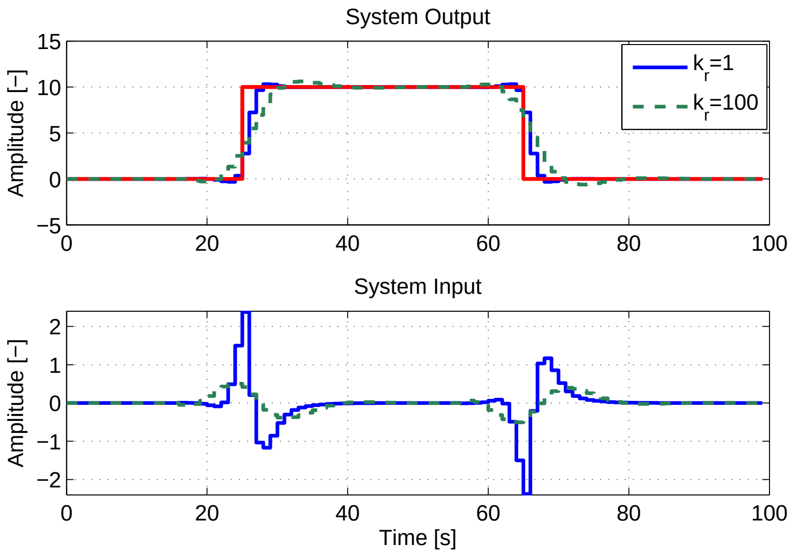

Figure 2.4: Unconstrained MPC example - reference tracking

- Mode 1 control: a finite horizon with N samples over which the control inputs are free variables and they are determined by solving the optimization problem.

- Mode 2 control: the subsequent infinite horizon over which the control inputs are determined by a state feedback law: $u(k) = -Kx(k)$. The gain matrix K is the feedback gain that ensures the unconstrained closed-loop stability.

Then, the cost function (2.22) can be expressed in the form

$$J\left(\vec{u}|x(0)\right) = \sum_{k=0}^{N-1} \left(x(k)^T Q x(k) + u(k)^T R u(k)\right) + \Psi\left(x(N)\right) \ . \tag{2.23}$$

The first part of (2.23) is a standard form of finite horizon cost function and the last term, $\Psi\left(x(N)\right)$, is known as a *terminal penalty term* and corresponds to the value of the cost function on the interval $\langle N,\ \infty)$, i.e.

$$\Psi\left(x(N)\right) = \sum_{k=N}^{\infty} \left(x(k)^T Q x(k) + u(k)^T R u(k)\right) . \tag{2.24}$$

Using the quadratic cost function (2.22), it is straightforward to show that $\Psi(x)$ is a quadratic function

$$\Psi(x) = x^T \Psi x \qquad \Psi \geq 0 \ , \tag{2.25}$$

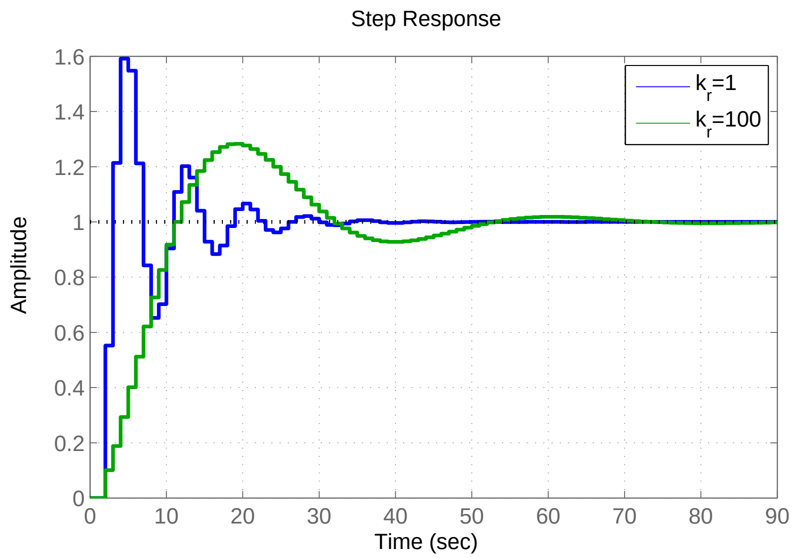

Figure 2.5: Unconstrained MPC example - step response

where the matrix Ψ is the solution of the discrete-time algebraic Riccati equation

$$\Psi = A^T\Psi A - A^T\Psi B\left(R + B^T\Psi B\right)^{-1}B^T\Psi A + Q \tag{2.26}$$

$$K = \left(R + B^T\Psi B\right)^{-1}B^T\Psi A\,. \tag{2.27}$$

The infinite horizon cost function (2.22) can be now rewritten to the final form

$$J\left(\vec{u}|x(0)\right) = \sum_{k=0}^{N-1}\left(x(k)^TQx(k) + u(k)^TRu(k)\right) + x(N)^T\Psi x(N)\,. \tag{2.28}$$

It was shown that the infinite horizon cost function (2.22) can be written as (2.28) where Ψ is the appropriate solution of (2.26). The first part of the criterion is minimized using a standard on-line optimization technique (e.g. quadratic programming) including the system constraints. In the second part, it is considered that the system is controlled by the LQ optimal state feedback. Note that the terminal penalty term is the basic tool when formulating and proving the stability of the MPC controller.

2.3.3 Stability

The stability of MPC controller cannot be simply analyzed as it can be done in the classical control methods. The properties of the MPC closed loop are influenced by all tuning parameters, e.g. by cost function form, prediction and correction horizon length, weighting matrices, etc. We can do the

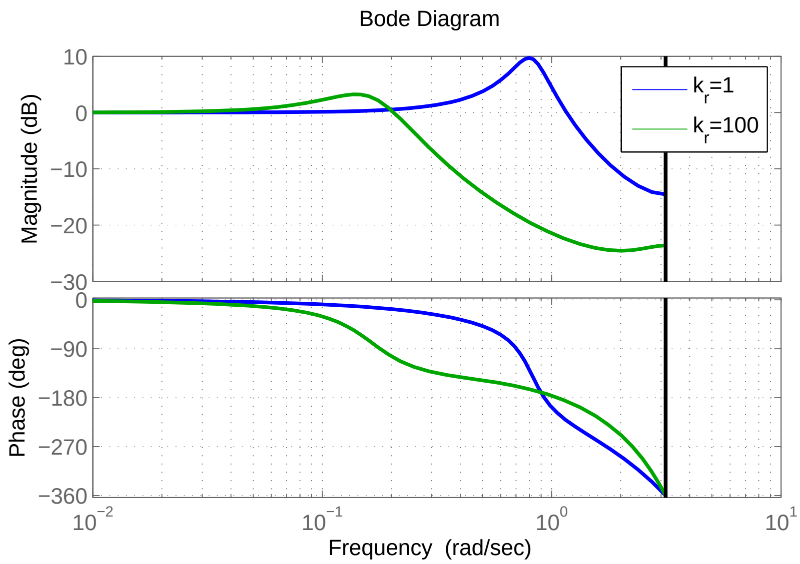

Figure 2.6: Unconstrained MPC example - frequency response

analysis if there are no active constraints or for a selected feasible set of active constraints. The problem is that the MPC controller can be seen as a nonlinear controller. In fact, it can be shown, that the linear MPC control is based on switching the affine control laws, where the number of control laws corresponds to the number of all feasible combinations of active sets. This number may be huge even for relatively small number of constraints. Therefore, it is clear that the analysis may not be so easy, or even impossible.

The stability of the MPC control is not ensured in its basic formulation. On the other hand, it is fair to say that the basic MPC formulation gives very good results and provides a good degree of stability and robustness in practical applications. We will discuss the basic tools which can be used to ensure the nominal stability of the controller during the design stage. There are several ways and the most important are the following:

- *Terminal equality constraints:* [20] If the system origin is stable, then the stability can be ensured by a terminal equality constraints on the system state at the end of the prediction horizon, i.e. $x(k+N) = 0$. It is clear that this can be generalized to $x(k+N) = x_e$, where x_e is a stable equilibrium.

- *Terminal cost function:* [9] The main idea is to add a terminal cost term to the cost function.

- *Terminal constraint set:* The terminal equality constraint can be generalized. The idea is based on assumption that there exist a subspace in the system state space for which it holds that if the system state enters this subspace, then it will stay inside at all future time without violating the constraints.

From the practical point of view, the most important is the combination of the last two and therefore, we will focus on them. Before formulating an MPC algorithm that ensures the nominal stability, we will define *positively invariant set*, *admissible positively invariant set* and *maximal admissible positively invariant set* [10, 30]:

Definition 1 *A positively invariant set Ω is a region of state space with the property that all state trajectories starting from an initial condition within the set remain within the set at all future instants.*

Definition 2 *An admissible positively invariant set Ω is a region of state space with the property that all state trajectories starting from an initial condition within the set remain within the set at all future instants and all considered constraints will be satisfied.*

Definition 3 *The maximal admissible positively invariant set (MAS) is a region of state space of all possible initial states so that all state trajectories starting from an initial condition within the set remain within the set at all future instants and all considered constraints will be satisfied.*

Formally, an admissible positively invariant set Ω can be defined as

$$(A - BK)\, x(k) \in \Omega \quad \forall x(k) \in \Omega \tag{2.29}$$

$$m_{\min} \leq Mx(k) \leq m_{\max} \quad \forall x(k) \in \Omega \, . \tag{2.30}$$

The MAS sets can be approximated (if we cannot compute them exactly) by a polytopic or ellipsoidal sets. An example of polytopic MAS and ellipsoidal MAS are depicted on Fig. 2.7, where Ω_f is a set of all feasible initial states, which can be driven into the MAS Ω withing the given prediction horizon. Now assume MPC control using quasi-infinite prediction horizon

$$J\left(\vec{u}_k | x(k), k\right) = \|x(k+N)\|_{\Psi}^2 + \sum_{i=0}^{N-1} \|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2 \, , \tag{2.31}$$

with constraints

$$G\vec{u} \leq W + Sx(k) \tag{2.32}$$

and additional, stability, constraints

$$x(k+N) \in \Omega \tag{2.33}$$

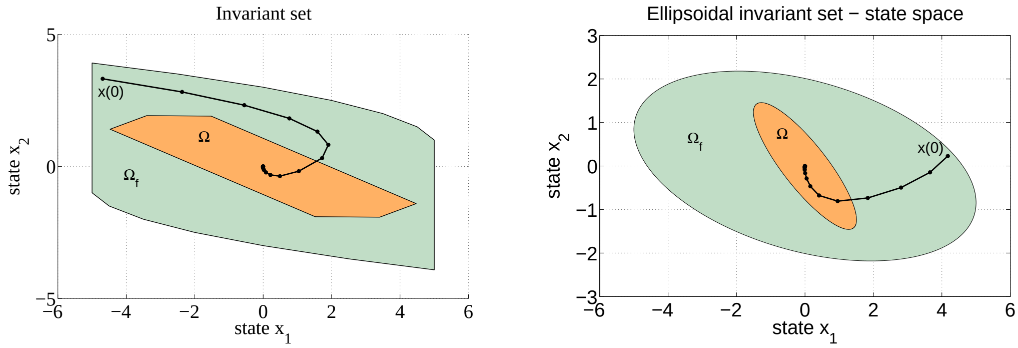

Figure 2.7: An example of polytopic and ellipsoidal MAS (Ω) and set of all feasible initial conditions (Ω_f) [27]

where Ω is MAS (or admissible positively invariant set) for the controlled system. To prove the stability, we need to find a Lyapunov function. It is not a surprise, that a nature candidate for the Lyapunov function is the cost function (2.31), i.e.

$$V(k) = J\left(\vec{u}_k^* | x(k), k\right) \ . \tag{2.34}$$

Assume that the optimal solutions at time k and $k+1$ are

$$\begin{aligned} \vec{u}_k^* &= \left[\begin{array}{cccc} u^*(k|k) & u^*(k+1|k) & \ldots & u^*(k+N-1|k) \end{array}\right] , \\ \vec{u}_{k+1}^* &= \left[\begin{array}{cccc} u^*(k+1|k+1) & u^*(k+2|k+1) & \ldots & u^*(k+N|k+1) \end{array}\right] \end{aligned}$$

and assume that at time k, there exist an estimate of the optimal control sequence for time $k+1$, denoted by $\vec{u}_{k+1}^{shifted}$, i.e.

$$\vec{u}_{k+1}^{shift} = \left[\begin{array}{cccc} u(k+1|k) & u(k+2|k) & \ldots & u(k+N|k) \end{array}\right] . \tag{2.35}$$

From the definition, it holds that

$$V(k+1) = J\left(\vec{u}_{k+1}^* | x(k+1), k+1\right) \leq J\left(\vec{u}_{k+1}^{shifted} | x(k+1), k+1\right) \tag{2.36}$$

and we can continue

$$\begin{aligned} V(k+1) &\leq J\left(\vec{u}_{k+1}^{shifted} | x(k+1), k+1\right) \\ &\leq J\left(\vec{u}_k^* | x(k), k\right) - \|x(k|k)\|_Q^2 - \|u(k|k)\|_R^2 - \|x(k+N|k)\|_\Psi^2 \\ &\quad + \|u(k+N|k)\|_R^2 + \|x(k+N+1|k)\|_\Psi^2 \ . \end{aligned}$$

It holds that

$$V(k) = J\left(\vec{u}_k^* | x(k), k\right)$$

and therefore

$$\begin{aligned} V(k+1) - V(k) \leq\ & -\|x(k|k)\|_Q^2 - \|u(k|k)\|_R^2 - \|x(k+N|k)\|_\Psi^2 \\ & + \|u(k+N|k)\|_R^2 + \|x(k+N+1|k)\|_\Psi^2 \ . \end{aligned}$$

The Lyapunov function must satisfy condition $V(k+1) - V(k) \leq 0$. It is clear that this condition will be satisfied if

$$\|x(k+N|k)\|_{\Psi}^2 \geq \|u(k+N|k)\|_R^2 + \|x(k+N+1|k)\|_{\Psi}^2 \tag{2.37}$$

If there exist a control law for which the condition (2.37) is satisfied, then $V(k)$ is a Lyapunov function and the receding horizon MPC control sequence will stabilize the system. The two basic possibilities are the following:

- $u(k+i) = 0$, $i \geq N$: Then, the condition (2.37) leads to the *Lyapunov equation*

  $$A^T \Psi A - \Psi \leq 0$$

  i.e. a condition, that the system is stable and the weighting matrix Ψ of the terminal penalty term is a Lyapunov equation solution. The set Ω used in (2.33) is an admissible positively invariant set for the open loop system.

- $u(k+i) = -Kx(k+i)$, $i \geq N$: Then, the condition (2.37) leads to the *Algebraic Riccati Equation*, i.e.

  $$(A - BK)^T \Psi (A - BK) + K^T R K \leq \Psi \, .$$

  In this case, the control law K and weighting matrix Ψ must satisfy the algebraic Riccati equation and Ω utilized in (2.33) is corresponding admissible positively invariant set.

It has been shown that the stability can be ensured (and proved) by adding a terminal penalty term to the cost function and a terminal constraints set (known also as stability constraints). This concept can be seen as *Dual mode control* startegy where:

- Mode 1: The system inputs are determined by solving the optimization problem for the finite prediction horizon.

- Mode 2: The system inputs are determined by a state space feedback law. This mode is never applied because of the receding horizon strategy.

The presented concept can be seen as a tool for ensuring stability for general formulation of linear MPC. We should note that, in general, we do not need to follow the concept to ensure the stability of MPC for practical applications. There are many other ad-hoc solutions ensuring the reasonable behavior but usually, these methods are tailored for particular MPC formulations or applications and do not hold for general MPC formulation.

The MPC control is based on solving the constrained optimization problem in each sampling period. Therefore, it is also necessary to analyze the feasibility of this optimization, especially if the receding horizon control is considered. It can be shown that if the problem is feasible for the initial system state, then it is feasible in all subsequent sampling periods (see Fig. 2.7 for illustration). The proof can be found in the literature.

2.3.4 Robustness

All systems models used for control design in practical applications have some uncertainties. These uncertainties are caused by disturbances, by inaccurate identification, incorrect model structure, due to model simplification, etc. Therefore, it is clear that the model does not describes the controlled system accurately and the controller must be robust with respect to these inaccuracies. Robustness is a fundamental question for all feedback control systems. Any statement about the robustness must be connected with a specific uncertainty range and to a specific performance criteria. It is clear that the robust control design may be a very difficult and challenging task. Therefore, for MPC we will present only some of the basic ideas. In the robust control design we have to consider namely:

- Uncertainty description and modeling
- Robust control design
- Robust analysis

The first and the last items may be relative simple when compared to the robust control design, which may be a challenge, or even impossible.

Uncertainty description

There are several approaches how to describe the uncertainties of the controlled system or inaccurate model. Selection of an approach depends mainly on the controller design method. For example, if a controller design method is based on frequency domain, the uncertainties should be also based on the frequency domain. In the MPC context, the most important approaches are the following two [7]:

- The system behavior is described by a set of models, for example, the true plant Σ_0 belongs to a set $\mathcal{S}$, $\Sigma_0 \in \mathcal{S}$, where the set $\mathcal{S}$ is a given family of LTI systems. Mathematically

$$x(k+1) = Ax(k) + Bu(k) \; , \quad (A, B) \in \mathcal{S} \; , \tag{2.38}$$

where

$$\mathcal{S} = \left\{ \sum_{i=1}^{h} \eta_i \left(A_i, B_i\right); \sum_{i=1}^{h} \eta_i = 1,\ \eta_i \geq 0 \right\}.$$

- Unmeasured disturbance signal $w(k)$ acts on the system, where $w(t) \in \mathcal{W}$ and $\mathcal{W}$ is a priory known set. Mathematically

$$x(k+1) = Ax(k) + Bu(k) + Fw(k)\ , \quad w(k) \in \mathcal{W} \tag{2.39}$$

Robust MPC design

In the MPC robust control design, we need to formulate an optimization problem that ensures the robustness. We defined two classes of uncertainties that are often used in the linear MPC. When designing the robust MPC, we can follow the concept presented in the section about the stability, i.e. the dual mode control. First, we need to define robust admissible positively invariant set:

Definition 4 *A robust admissible positively invariant set Ω is a region of state space with the property that all state trajectories of the system controlled by a state feedback starting from an initial condition within the set remain within the set at all future instants for all considered perturbations and any of considered constraints is not violated.*

An example of a robust admissible positively invariant set is depicted on Fig. 2.8. Secondly, we have to formulate a suitable cost function as a function of the uncertain parameter, i.e.

$$J\left(\vec{u}_k | x(k), \theta(k), k\right), \tag{2.40}$$

where $\theta(k)$ is the uncertain parameter, for which we know that $\theta(k) \in \Theta$. In case (2.38), the criterion function will be parameterized by η, i.e.

$$\Theta = \left\{ \eta :\ \sum_{i=1}^{h} \eta_i = 1,\ \eta_i \geq 0 \right\}, (A, B) \in \mathcal{S}\ ,$$

and in case (2.39)

$$\Theta = \mathcal{W}\ ,$$

i.e. $\theta(k) = w(k)$. Assume that there exist a control law $u(k) = -Kx(k)$ for which we can found a robust admissible positively invariant set Ω_{robust} for $\theta(k) \in \Theta$ and a corresponding terminal penalty term. Then the robust MPC optimization problem can be defined as min-max optimization [15]

$$\vec{u}_k^* = \arg\min_{\vec{u}_k} \left\{ \max_{\vec{\theta}_k} J\left(\vec{u}_k | x(k), \vec{\theta}_k, k\right) \right\}$$

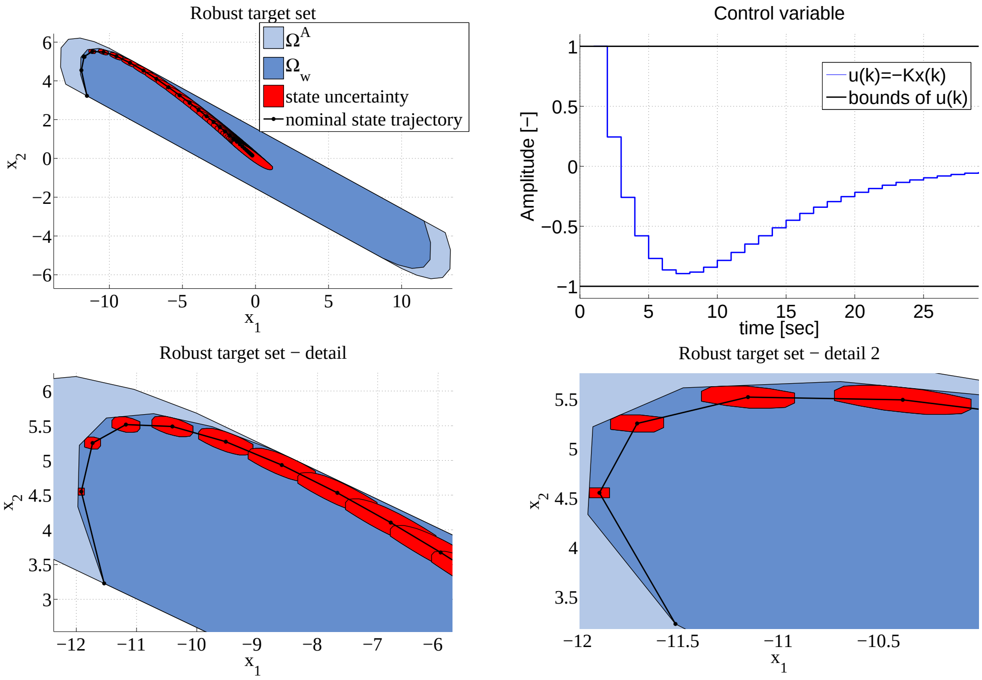

Figure 2.8: Example of admissible positively invariant set Ω^A, robust admissible positively invariant set Ω_W for uncertainty description (2.39) and system state trajectory uncertainty (red sets) [27].

subject to the constraints

$$\vec{u}_k \in \mathcal{U}_k$$

and robust stability constraints (or terminal constraints)

$$\mathcal{X}_k^N \in \Omega_{robust} \ ,$$

where $x(k+N) \in \mathcal{X}_k^N$ and $\mathcal{X}_k^N$ is a set of all possible values of the system state at the end of prediction horizon. An illustrative example of the stability constraints for the case (2.39) is depicted on Fig. 2.9.

The min-max approaches have several important drawbacks: (i) they are computationally demanding (based on dynamic programming), (ii) the resulting control action may be too conservative. Of course there exist many other formulations of robust MPC. For example, it can be shown, that instead of looking for the optimal control sequence, we should be looking for a sequence of the control laws $\kappa_{k+i}\left(x(k+i), k\right)$ when dealing with the robust MPC control. A better formulation would be to perform for each step i in the prediction horizon maximization over $\theta(k+i)$ and immediately minimization over $u(k+i)$. Another important question is whether the constraints should be satisfied for the nominal plant only or for all possible perturbations. All these questions have been discussed by many authors and the practical robust

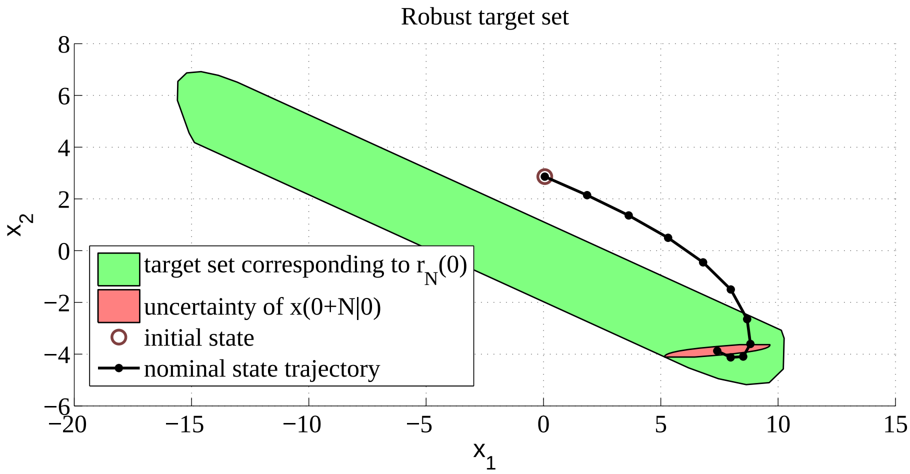

Figure 2.9: Illustration of a robust target set: robust admissible positively invariant set Ω_{robust} (green) and $\mathcal{X}_0^N$ (orange) [27].

MPC formulation is still an area of active research.

2.4 Hybrid systems

Hybrid systems are a special class of dynamical systems that combines both continuous and discrete-value variables. The main components of the hybrid systems are the continuous dynamics (based on first principle), logical components (switches, automate, logical conditions, etc.) and interconnections between the logic and dynamic. The hybrid systems can be used to model systems with several operation modes where each mode has different dynamical behavior. A simple example of a hybrid system is a piece-wise affine (PWA) system, defined as

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) + f_i \\ y(k) &= C_i x(k) + D_i u(k) + g_i \end{aligned}$$

if

$$\begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \mathcal{T}_i, \qquad i = 1, 2, \ldots, n \; .$$

The PWA systems enables to describe a large class of practical applications and are very general. Unfortunately, they are not directly suitable for the analysis and synthesis of optimal control problems. Another useful framework for the hybrid systems is based on Mixed Logical Dynamical (MLD) models [8]. These models transform the logical part of a hybrid system into the mixed-integer linear inequalities by using Boolean variables. The basic form of the MLD system is given by [8]

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_2\delta(k) + B_3z(k) \ , \\ y(k) &= Cx(k) + Du(k) + D_2\delta(k) + D_3z(k) \ , \end{aligned}$$

subject to

$$E_2\delta(k) + E_3z(k) \leq Eu(k) + E_4x(k) + E_5 \ ,$$

where $x(k)$ is a combined continuous and binary state, $u(k)$ and $y(k)$ are the system inputs and outputs (continuous and binary), $\delta(k)$ are auxiliary binary variables and $z(k)$ are auxiliary continuous variables. Now, we can define the optimal control problem for PWA system as

$$J\left(\vec{u}_k|x(k),k\right) = \left\|\Psi x(k+N)\right\|_p + \sum_{i=0}^{N-1} \left\|Qx(k+i)\right\|_p + \left\|Ru(k+i)\right\|_p , \tag{2.41}$$

subject to

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) + f_i \\ y(k) &= C_i x(k) + D_i u(k) + g_i \\ & \quad if \ \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \mathcal{T}_i, \qquad i = 1, 2, \ldots, n \\ & \quad u(k) \in \mathcal{U} \end{aligned}$$

The PWA system can be represented by a MLD model and therefore, the optimal control problem corresponds to the solution of mathematical mixed-integer program. In case of PWA system, if the cost function is quadratic, then the optimization problem leads to Mixed-Integer Quadratic Program and if the the cost function is based on l_1 or l_∞ norm, the optimization problem leads to Mixed-Integer Linear Programming.

2.5 Optimization algorithms

We will conclude this chapter about the linear model predictive control by a brief discussion about suitable optimization methods for solving the optimization problems. It has been shown that the MPC control problem can be formulated as an optimization problem that is solved at each sampling period. Therefore, the performance of the optimization algorithm in MPC is critical. Assume a QP problem in the form

$$\vec{u}^* = \arg\min_{\vec{u}} \frac{1}{2}\vec{u}^T H\vec{u} + \vec{u}^T F\vec{p} \, , \quad s.t. \quad G\vec{u} \leq W + S\vec{p} \, . \tag{2.42}$$

Direct solution to this QP by using general QP solver can be too slow for some applications and therefore this approach is suitable only for relatively slow systems. The modern QP solvers are based on active set or interior-point approach. The active set solvers are iterative algorithms. In each iteration, we are testing the optimality conditions for actual working set of active constraints. If the working set of active constraints does not lead to the optimal solution, then we modify the set by adding or removing the active constraints. In general, the active set solvers are suitable for relatively small problems but they are very efficient in practice, especially in combination with warm-starting strategy. The interior-point methods are based on barrier functions. The constraints are added to the criterion function in the form of a barrier which transforms the original problem to an unconstrained optimization. The interior-point methods are iterative (solution to optimality conditions) and usually require only a small number of iterations when compared to active set solvers. However, the individual iterations are more computationally expensive.

If we need extremely fast sampling periods in the MPC, we can use multi-parametric explicit solution [5, 6]. These optimization algorithms have off-line and on-line parts. The MPC optimization problem is solved explicitly in the off-line part. The explicit solution divides the optimization problem parameter space into a number of regions where each region has associated a control law. A particular region corresponds to a feasible combination of active constraints. All these regions and the control laws are stored for the on-line part. In the on-line part at each sampling period, we simply construct the parameter vector and find the corresponding region. Then we apply the associated control law. Unfortunately, the multiparametric explicit solution is applicable for small systems only due to storage demands. The complexity of parametric explicit solvers are compared with the active set solvers in [12].

Another way how to improve performance of the MPC optimization is to explore the structure of the MPC optimization problem and use this information to design an efficient solver. For example, there are two ways how to add the soft constraints to the optimization problem. One of them leads to simple[6], or box, constraints. if all the constraints in the optimization problem are box, then we can use this information to implement an efficient solver, e.g. based on gradient projection methods or their modifications.

[6] The box constraints are defined so that we have variables with upper and lower limits, e.g. $x_{min} \leq x \leq x_{max}$.

Chapter 3

Nonlinear Model Predictive Control

Today's processes need to be controlled under tight performance specifications which can be only met if the controller works precisely. Nonlinear model predictive control (NMPC) is extension of the well established linear predictive control to the nonlinear world. Linear model predictive control refers to MPC algorithms in which the linear models are used. The nonlinear model predictive control refers to MPC schemes that are based on the nonlinear models. Because NMPC enables the optimal control of constrained nonlinear systems, it is one possible candidate as an advanced control scheme for industrial processes. The nonlinear model predictive control has been intensively studied since the 90s. The fundamentals of NMPC are revieved for example in [3, 32, 4].

3.1 Formulation of nonlinear MPC

The first step in NMPC design is obtaining an accurate system model. Usually, in the practical applications, we are able to find a model based on physical laws. The model should be as accurate as possible to ensure reasonable control performance. Note that the modeling phase in NMPC design is usually the most difficult part. Consider a continous-time nonlinear system of the form

$$\begin{aligned} \dot{x}(t) &= f\left(x(t), u(t)\right) \\ y(t) &= h\left(x(t), u(t)\right) \end{aligned}$$

where $x(t)$ is the system state, $u(t)$ is the system input and $y(t)$ is the system output. The second step in the model predictive control design is definition of the cost function. The general objective function for the nonlinear system on infinite prediction horizon has the integral form

$$J\left(u(t), x(t_0)\right) = \int_{t_0}^{\infty} L\left(x(t), u(t), t\right) dt\ , \tag{3.1}$$

where the function $L\left(x(t), u(t), t\right)$ defines the control objectives. This function reflect the basic requirements on the controller performance and is often defined as a sum of weighted quadratic functions of tracking error and control signal, e.g.

$$L\left(x(t), u(t), t\right) = \|r(t) - y(t)\|_Q^2 + \|u(t)\|_R^2 \ , \tag{3.2}$$

where $r(t)$ is a known reference trajectory. This is the basic form and is modified with respect to a particular application and actual system requirements. The cost function can be split into a finite prediction horizon term and a terminal cost as follows

$$J\left(u(t), x(t_0)\right) = \Psi\left(x(t_N)\right) + \int_{t_0}^{t_N} L\left(x(t), u(t), t\right) dt \ . \tag{3.3}$$

where the terminal cost is ideally given by

$$\Psi\left(x(t_N)\right) = \int_{t_N}^{\infty} L\left(x(t), K\left(x(t)\right), t\right) dt \tag{3.4}$$

Assume that the control signal at the time interval $t \in \langle t_N, \infty\rangle$ is given by a control law $K\left(x(t)\right)$. The general nonlinear MPC problem can be formulated as a nonlinear optimization problem defined as minimization of the cost function

$$\min_{u(t)} J\left(u(t)|x(t_0)\right) \ , \tag{3.5}$$

subject to constraints

$$\begin{aligned}
\dot{x}(t) - f\left(x(t), u(t)\right) &= 0 \ , \\
x(t_0) - x_0 &= 0 \ , \\
g\left(x(t), u(t)\right) &\leq 0 \ , \quad t \in (t_0, t_N) \ , \\
u(t) &\in \mathcal{U} \ , \quad t \in (t_0, t_N) \ , \\
x(t) &\in \mathcal{X} \ , \quad t \in (t_0, t_N) \ , \\
x(t_N) &\in \Omega \ ,
\end{aligned}$$

where the last constraint is the stability constraint. the optimal control trajectory at the time horizon $t \in \langle t_0, t_N\rangle$ can be obtained by solving the above nonlinear constrained optimization problem. The explicit solution is not usually possible and therefore the problem has to be solved by a suitable numerical method.

3.2 Analysis of nonlinear MPC

In this section we will briefly discuss the basic stability results. There exist a number of schemes ensuring the stability of the resulting control system [17]. Most of them modify the MPC control scheme by adding a terminal constraints to the optimization problem and/or terminal costs to the objective function. The terminal cost approximates the infinite horizon control and is usually connected with a local controller. The terminal constraints are selected so that the system state lies in the domain of attraction of the local controller. The principal idea to state the conditions for stability is to select the objective function as a Lyapunov function of the closed-loop system, i.e

$$V\left(t_N, x(t_0)\right) = \Psi\left(x(t_N)\right) + \int_{t_0}^{t_N} L\left(x(\tau), u(\tau)^*, \tau\right) d\tau \tag{3.6}$$

The stability conditions are summarized in the following Theorem:

Theorem 1 *Suppose that $\Psi(x_e) = 0$, $x_e \in \Omega$, $\Omega \subseteq X$ is a closed set and the optimization problem is feasible at t_0. Then the nominal closed-loop system is asymptotically stable for any time $\delta \in (t_0, t_N)$ if there exists a local control law $u(t) = \kappa(x(t))$ for $t \geq t_N$ with $u_e = \kappa(x_e)$ such that:*

$$\frac{\delta\Psi\left(x(t)\right)}{\delta x(t)}\dot{x}(t) + L\left(x(t), u(t), t\right) \leq 0, \quad x(t) \in \Omega, \quad \kappa\left(x(t)\right) \in \mathcal{U} \tag{3.7}$$

The proof of Theorem can be found in the literature. Note that in general, it is not easy to find a terminal penalty Ψ and terminal set Ω satisfying conditions in the Theorem.

3.3 Numerical methods for nonlinear MPC

A commonly used approach to solve the problem (3.5) is reformulation to a finite dimensional nonlinear programming problem (NLP) by a suitable parameterization. The most recent research in the nonlinear MPC suggests to perform this parameterization by using Direct Multiple Shooting method [11, 16]. The nonlinear programming problem can be solved by iterative Sequential Quadratic Programming approach (SQP). To find the optimal solution to the defined NLP, it is usually necessary to perform several iterations which may be a time consuming task. Therefore, it is suggested to perform only one iteration in each sampling period in real-time applications and to use a sub-optimal instead of the optimal solution [16].

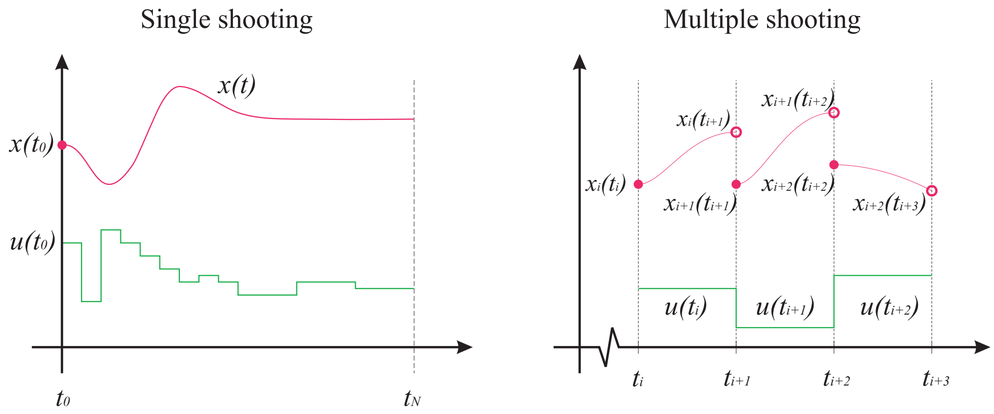

Figure 3.1: Direct single shooting (left) and direct multiple shooting (right)

There are two important direct approaches to solve the nonlinear optimization problems in the real-time optimizations:

- **Direct single shooting** is a basic approach and is similar to the approach used by the standard linear model predictive control. At the initial time, the numerical integration is used to obtain the predicted trajectories as a function of manipulated variable for the prediction horizon, see Fig. 3.1. Having these trajectories, one can perform one iteration of SQP procedure.

- **Direct multiple shooting** [11] is based on re-parameterization of the problem on the prediction horizon. The pieces of system trajectories are found on each time interval numerically together with sensitivity matrices, see Fig. 3.1. The optimization problem is then augmented by auxiliary constraints - continuity conditions.

Such parameterization can be regarded as simultaneous linearization and discretization. One advantage of the multiple shooting methods is that the optimization problem is sparse, i.e. the Jacobians in the optimization problem contain many zero elements which makes the QP subproblem cheaper to built and to solve. The simulation (solution to the model) and optimization are performed simultaneously and the solution to the problem can be parallelized. The direct multiple shooting approach parameterizes the optimization problem by a finite set of parameters, i.e. by system states $x_i(t_i)$ (auxiliary optimization variable) and system inputs $u(t_i)$. The key idea of finite parameterization is to find the sensitivity matrices (or linearization) so that

$$\begin{aligned}
\delta x_i(t_{i+1}) &\approx \Phi(t_{i+1}, t_i)\delta x_i(t_i) + \Gamma\left(t_{i+1}, t_i\right)\delta u_i(t_i)\ , \\
\delta x_{i+1}(t_{i+1}) &= \delta x_i(t_{i+1})
\end{aligned}$$

where $\Phi(t_{i+1}, t_i)$ and $\Gamma(t_{i+1}, t_i)$ are sensitivity matrices defined by

$$\Phi(t, t_0) = \frac{\partial x(t)}{\partial x(t_0)}, \quad \Gamma(t, t_0) = \frac{\partial x(t)}{\partial u(t_0)} . \tag{3.8}$$

The sensitivity of the system state trajectory to the initial condition can be computed by solving the following differential equation

$$\dot{\Phi}(t, t_0) = \frac{\partial f(x(t), u(t))}{\partial x(t)} \Phi(t, t_0) , \quad \Phi(t, t_0) = \frac{\partial x(t_0)}{\partial x(t_0)} = I \tag{3.9}$$

and the sensitivity of the system trajectory to the system input at time t_0 is given by

$$\dot{\Gamma}(t, t_0) = \frac{\partial f(x(t), u(t))}{\partial x(t)} \Gamma(t, t_0) + \frac{\partial f(x(t), u(t))}{\partial u(t)} 1(t - t_0) ,$$

where $1(t - t_0)$ is the unit step defined as

$$1(t - t_0) = \begin{cases} 0 & t < t_0 \\ 1 & \text{otherwise} \end{cases}$$

and the initial condition is

$$\Gamma(t, t_0) = \frac{\partial x(t_0)}{\partial u(t_0)} = 0 \tag{3.10}$$

Calculation of sensitivity matrices for the nonlinear system requires solution to a set of differential equations simultaneously with the system trajectory. This may be a consuming task.

It was shown how the optimization problem can be parameterized by a finite number of parameters in the multiple shooting approach. Using these results, the control problem (3.5) can be formulated as mathematical programming

$$\min_{u(t_i), x_i(t_i)} \sum_{i=0}^{N-1} L_i(x_i(t_i), u(t_i), t_i) + \Psi(x_N(t_N)) \tag{3.11}$$

subject to constraints

$$\begin{aligned} x_{i+1}(t_{i+1}) - x_i(t_{i+1}) &= 0 , \quad t \in (0, N-1) , \\ x_0(t_0) - x(t_0) &= 0 , \\ g(x_i(t_i), u(t_i)) &\leq 0 , \quad t \in (0, N) , \\ u(t_i) &\in \mathcal{U} , \quad t \in (0, N) , \\ x_i(t_i) &\in \mathcal{X} , \quad t \in (0, N) , \\ x_N(t_N) &\in \Omega , \end{aligned}$$

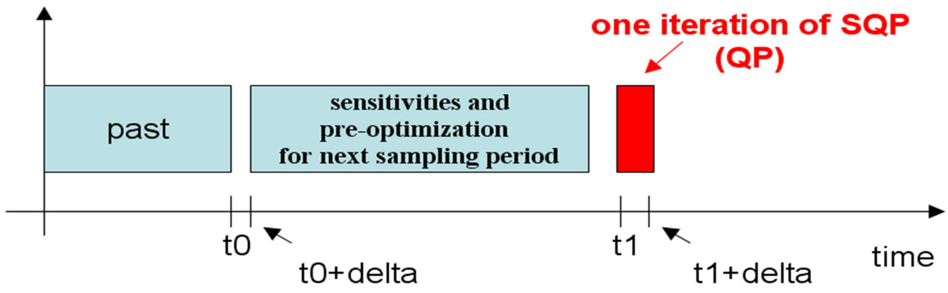

Figure 3.2: Timing diagram for real-time optimizations

The cost function at time interval $t \in (t_i, t_{i+1})$ is equal to

$$L_i\left(x_i(t_i), u(t_i), t_i\right) = \int_{t_i}^{t_{i+1}} L\left(x_i(\tau), u(\tau), \tau\right) d\tau \tag{3.12}$$

and $x_i(t_{i+1})$ is solution of the nonlinear system at time t_{i+1} with initial condition $x_i(t_i)$. The resulting nonlinear programming problem (3.11) can be solved, for example, by a suitable SQP framework [24].

The model based predictive control algorithms are usually formulated with receding horizon where the optimization problem is re-calculated in each sampling period and only the first control action is applied to the system. A very efficient scheme has been proposed in [16]. The timing scheme of this approach is depicted on Fig. 3.2. There are two main phases: preparation phase and feedback phase. During the preparation phase, the algorithm calculates as much as it is possible without knowledge of data that will be available at the beginning of the next sampling period. The feedback phase takes new measurement and calculates the control action that can be immediately sent to the system.

Bibliography

[1] Alessio, A., Bemporad, A. Decentralized Model Predictive Control of Constrained Linear Systems. In *Proceedings of the European Control Conference*, Kos, Greece, 2007.

[2] Allgöwer, F., Zheng, A. *Nonlinear Model Predictive Control.* Birkhäuser Verlag AG, April 2000.

[3] Allgöwer, F., Badgwell, T. A., Qin, J. S., Rawling, J. B., Wright, S. J. *Advances in Control, Highlights of ECC'99.*, chapter Nonlinear predictive control and moving horizon estimation - An introductory overview, s. 391–449 Springer, 1999.

[4] Allgöwer, F., Findeisen, R., Nagy, Z. K. Nonlinear Model Predictive Control: From Theory to Application. *Chin. Inst. Chem. Engrs.*, sv. 35, č. 3, s. 299–315, 2004.

[5] Bemporad, A., Borrelli, F., Morari, M. Model predictive control based on linear programming - the explicit solution. *Transaction on Automatic Control*, sv. 47, č. 12, s. 1974–1985, 2002a.

[6] Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E. N. The explicit linear quadratic regulator for constrained systems. *Automatica*, sv. 38, s. 3–20, 2002b.

[7] Bemporad, A., Morari, M. Robust model predictive control: A survey. *Lecture Notes in Control and Information Sciences*, sv. 245, s. 207–226, 1999.

[8] Bemporad, A., Morari, M. Control of Systems Integrating Logic, Dynamics, and Constraints. *Automatica*, sv. 35, č. 3, s. 407–427, 1999.

[9] Bitmead, R. R., Gevers, M., Wertz, V. *Adaptive optimal control - The thinking man's GPC.* Prentice-Hall, Englewood Cliffs, NJ, 1990.

[10] Blanchini, F. Set invariance in control. *Automatica*, sv. 35, č. 11, s. 1747–1767, 1999.

[11] Bock, H., Plitt, K. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress*, Budapest, 1984. Pergamon Press.

[12] BORRELLI, F., BAOTI´C, M., PEKAŘ, J., STEWART, G. On the Complexity of Explicit MPC Laws. In *Proceedings of European Control Conference*, Budapest, 2009.

[13] CAGIENARD, R., GRIEDER, P., KERRIGAN, E., MORARI, M. Move Blocking Strategie in Receding Horizon Control. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, 2004.

[14] CAMACHO, E. F., BORDONS, C. *Model Predictive Control.* Springer, Great Britain, 1999.

[15] CAMPO, P. J., MORARI, M. Robust Model Predictive Control. In *Proceedings of American Control Conference*, vol. 2, s. 1021–1026, 1987.

[16] DIEHL, M., BOCK, H., SCHÖDER, J. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM J. Control Optim.*, sv. 43, č. 5, s. 1714–1736, 2005.

[17] DIEHL, M. M. *Real-Time Optimization for Large Scale Nonlinear Processes.* PhD thesis, Ruprecht-Karls University, Heidelberg, Germany, 2001.

[18] GAL, T., NEDOMA, J. Multiparametric linear programming. *Management Science*, sv. 18, s. 406–442, 1972.

[19] GEORGES, D. Distributed Model Predictive Control via Decomposition-Coordination Techniques and the use of an Augmented Lagrangian. In *Proceedings of the IFAC Workshop on Nonlinear Model Predictive Control for Fast System*, Grenoble, France, 2006.

[20] KEERTHI, S. S., GILBERT, E. G. Optimal, infinite horizon feedback laws for a general class of constrained discrete time systems: Stability and moving-horizon approximations. *Journal of Optimization Theory and Application*, sv. 57, s. 265–293, 1988.

[21] MACIEJOWSKI, J. *Predictive Control with Constraints.* Prentice Hall, June 2001.

[22] MAYNE, D. Q., RAWLINGS, J. B., RAO, C. V., SCOKAERT, P. O. M. Constrained model predictive control: Stability and optimality. *Automatica*, sv. 36, s. 789–814, 2000.

[23] MUSKE, K. R., BADGWELL, T. A. Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, sv. 12, s. 617–632, 2002.

[24] NOCEDAL, J., WRIGHT, S. J. *Numerical Optimization.* Springer Series in Operations Research. Springer, 1999.

[25] KOUVARITAKIS, B., CANNON, M., editors *Nonlinear predictive control - theory and practice.* The Institution of Electrical Engineers, UK, 2001.

[26] PEKAŘ, J., ŠTECHA, J., HAVLENA, V. Identification and predictive control by p-norm minimization. In *Proceedings of the 23rd IASTED International conference on Modelling Identification, and Control [CD-ROM]*, s. 62–67, Zürich, Switzerland, 2004. IASTED, Acta Press.

[27] PEKAŘ, J. *Robust Model Predictive Control.* PhD thesis, Czech Technical University in Prague, Prague, Czech Republic, 2005.

[28] PROPOI, A. I. Use of linear programming methods for synthesizing sampled data automatic systems. *Automat. Remote Control*, sv. 24, č. 7, s. 837–844, 1963.

[29] PUIG, V., OCAMPO-MARTINEZ, C., CEMBRANO, G., CREUS, R., MINOVES, M. Improving water management efficiency by using optimal control strategies: the Barcelona case study. In *Proceedings of 10th IWA Conference on Instrumentation, Control and Automation*, Cairns, Australia, 2009.

[30] QIN, S. J., BADGWELL, T. A. A survey of industrial model predictive control technology. *Control Engineering Practice*, sv. 11, s. 733–764, 2003.

[31] RAWLINGS, J. B., STEWART, B. T. Coordinating multiple optimization-based controllers: New opportunities and challenges. In *Proceedings of of the 8th International IFAC Symposium on Dynamics and Control of Process Systems*, Cancun, Mexico, 2007.

[32] RAWLINGS, J. B. Tutorial overwiev of model predictive control. *IEEE Contr. Syst. Magazine*, sv. 20, č. 3, s. 38, 2000.

[33] RICHALET, J., RAULT, A., TESTUD, J. L., PAPON, J. Model predictive heuristic control: application to industrial processes. *Automatica*, sv. 14, s. 413–428, 1978.

[34] ROSSITER, J. A. *Model-Based Predictive Control.* CRC Press, USA, 2003.

[35] ROSSITER, J. A., KOUVARITAKIS, B. Constrained stable generalized predictive control. In *Proc. IEE Pt.D*, vol. 4, s. 243–254, 1993.

[36] VENKAT, A. N. *Distributed Model Predictive Control. Theory and Applications.* PhD thesis, The University of Wisconsin-Madison, 2006.

Centrum pro rozvoj výzkumu pokročilých řídicích a senzorických technologií
CZ.1.07/2.3.00/09.0031

Ústav automatizace a měřicí techniky
VUT v Brně
Kolejní 2906/4
612 00 Brno
Česká Republika

http://www.crr.vutbr.cz
info@crr.vutbtr.cz