INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Short Course on Model based Predictive Control

Jan Štecha Jaroslav Pekař

14.května 2010

Tato prezentace je spolufinancována Evropským sociálním fondem a státním rozpočtem České republiky.

Agenda I

- **Introduction**
  - Motivation for advanced control techniques
  - Classical approach
  - Brief history of model predictive control
- **Linear Model Predictive Control**
  - Formulation of linear MPC
  - Analysis of linear MPC
  - Hybrid systems and linear MPC
  - Optimization Algorithms for linear MPC

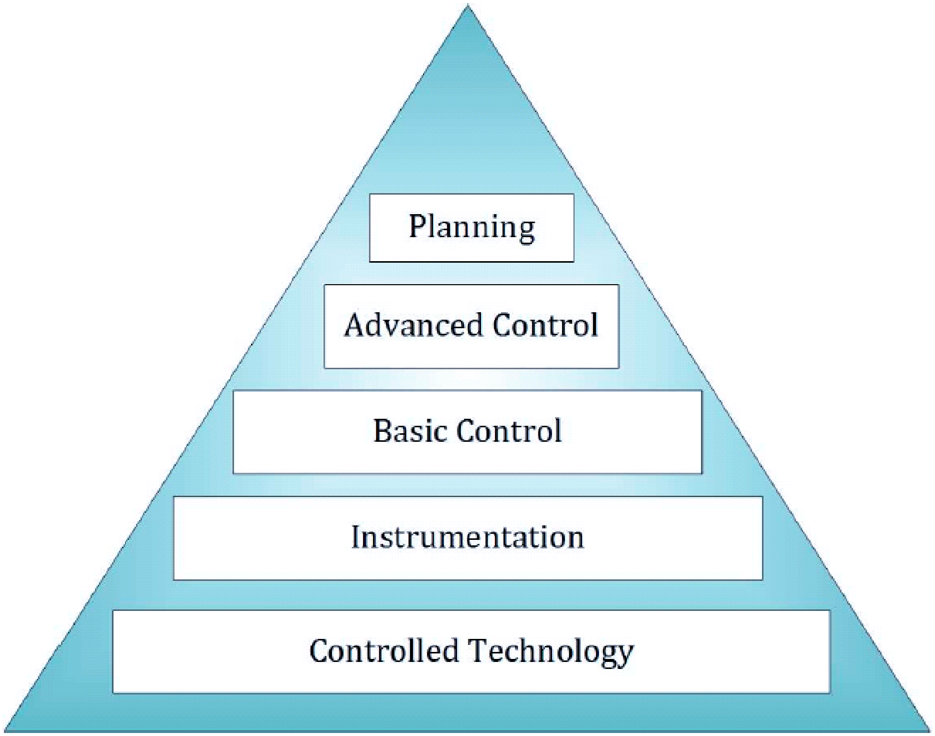
Agenda II

- **Nonlinear Model Predictive Control**
  - Formulation of nonlinear MPC
  - Analysis of nonlinear MPC
  - Numerical methods for nonlinear MPC

- **Practical Model Predictive Control**
  - Practical formulations of MPC
  - Development cycle of industrial MPC
  - Demonstration of an application

Motivation for Advanced Control

- **Advanced control technologies in process control**



Motivation for Advanced Control

- **Instrumentation layer**
  - Interface to the controlled technology
  - Actuators and sensors
  - Number of I/O points may be large – more than several thousands
  - Periodical reading\writing, the measured values are marked by time stamp and are organized in a process history database

Motivation for Advanced Control

- **Basic control layer**
  - Ensures basic functionality and safe operation
  - Provides backup solution for the advanced control
  - Basic monitoring and visualization tools
  - Provides basic control modes for the operators – man/aut/cas
  - Used as a gate for the advanced control technologies
    - System prestabilization
    - Nonlinearity reduction

Motivation for Advanced Control

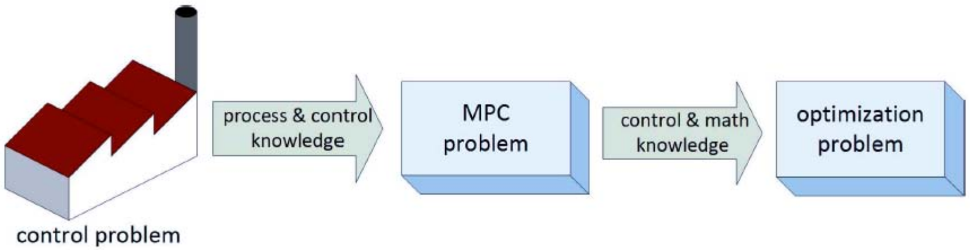
- **Advanced control layer**
  - Advanced optimization and coordination
  - Interaction with the basic control layer by coordination of basic control loops by manipulating their set-points
  - Slower sampling periods than in the basic control layer
  - Static/dynamic optimization
  - Basic requirements on advanced control layer
    - Multivariable control
    - Various constraints handling
    - Optimal solution

MPC is a candidate...

Motivation for Advanced Control

- **Planning layer**
  - Top supervisory layer
  - Entry points for the plant technologists and managers
  - Based on economic-related information and provides complex overview of the plant performance
  - Main tools are namely databases, visualization tools and specialized computation routines
  - Specification of goals for the advanced control layer, i.e. set-point, constraints, optimality conditions, resource availability, etc.

Motivation for Advanced Control



- **Model Predictive Control is ...**
  - A control technology that enables to delivery the decided goals specified for the controlled process
  - NOT a technology that could replace all the control techniques.

Motivation for Advanced Control

- **Model Predictive Control**
  - Success of applications depends namely on the skills of the application engineers responsible for MPC implementation to particular process
  - Transformation of MPC problem to an optimization problem is relatively simple
  - The difficult part is formulation of the control problem as a MPC problem

Agenda

- **Introduction**
  - Motivation for advanced control techniques
  - Classical approach
  - Brief history of model predictive control
- **Linear Model Predictive Control**
  - Formulation of linear MPC
  - Analysis of linear MPC
  - Hybrid systems and linear MPC
  - Optimization Algorithms for linear MPC

Classical approach

- **Three general optimization methods**
  - Mathematical programming
  - Discrete maximum principle
  - Dynamic programming

Mathematical programming

- **Assume a discrete time dynamic system**

$$x(t+1) = f\big(x(t), u(t), t\big)\ , \quad t = t_0, \ldots, t_1 - 1 \qquad x(t_0) = x_0$$

- **The optimal control problem is defined as a criterion minimization problem**

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

Mathematical programming

- **This is a problem of mathematical programming**
  - Minimization of criterion
  - Limiting conditions

- **To solve the optimization problem, let us define Lagrangian**

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

- **The Hamiltonian is defined as**

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

- **and the Lagrangian can be written in the form**

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

Mathematical programming

- **Simplified notation**

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

- **If the function J is differentiable with respect to x(t) and u(t), the increment of the criterion along the trajectories x(t) and u(t) equals to**

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

Mathematical programming

- **The necessary condition for the optimum:** $\partial \bar{J}/\partial x(t) = 0$
- **And therefore**

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

$$\frac{\partial h(t_1)}{\partial x(t_1)}^T - \lambda(t_1) = 0$$

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

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

Mathematical programming

- **For the solution of discrete optimal control problem it is necessary to find the solution to system of difference equation**

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

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

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

Maximum Principle

- **Assume state space equations of discrete time system**

$$x(t+1) = f(x(t), u(t))$$

- **The optimal control minimizes the criterion**

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

Maximum Principle

- **The Hamiltonian is equal to**

$$H(x(t), u(t), p(t+1)) = -g(x(t), u(t)) + p^T(t+1) f(x(t), u(t))$$

- **The maximum principle states that the optimal control maximizes the Hamiltonian, therefore**

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

Maximum Principle

- **The system equation is given by**

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

initial condition

$$x(t_0)$$

- **The conjugate system is given by**

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

final condition

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

Maximum Principle

- **Summary**
  - Discrete maximum principle changes the problem of the optimal control to two point boundary value problem of two set of difference equations and maximization of Hamiltonian with respect to control u(t)

Dynamic programming

- **Principle of optimality**
  - From arbitrary state x(t) our next decission must be optimal, without respect how the state is reached by previous decisions. 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.

Dynamic programming

- **Principle of imbedding**
  - 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.

Dynamic programming

- **Assume an optimality criterion in the form**

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

- **The Bellman function is defined as**

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

- **which can be reformulated as**

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

Dynamic programming

- **Resulting Bellman equation**

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

- **The solution to the control problem is then given by**

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

boundary condition

$$V(s, k_1) = h\left(s(k_1)\right)$$

Agenda

- **Introduction**
  - Motivation for advanced control techniques
  - Classical approach
  - Brief history of model predictive control
- **Linear Model Predictive Control**
  - Formulation of linear MPC
  - Analysis of linear MPC
  - Hybrid systems and linear MPC
  - Optimization Algorithms for linear MPC

MPC – brief history

- **Used by the industrial practitioners before having solid theoretical base**
- **First known formulation of a moving horizon controller using linear programming**

[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.

[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.

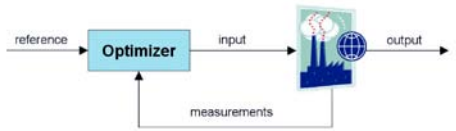
MPC – brief history

- **At present, MPC is a standard advanced control technology for the process industry**
  - The practical applications have been limited to the linear models
  - Sometimes, it is beneficial to define and implement the nonlinear MPC

Agenda

- **Introduction**
  - Motivation for advanced control techniques
  - Classical approach
  - Brief history of model predictive control
- **Linear Model Predictive Control**
  - Formulation of linear MPC
  - Analysis of linear MPC
  - Hybrid systems and linear MPC
  - Optimization Algorithms for linear MPC

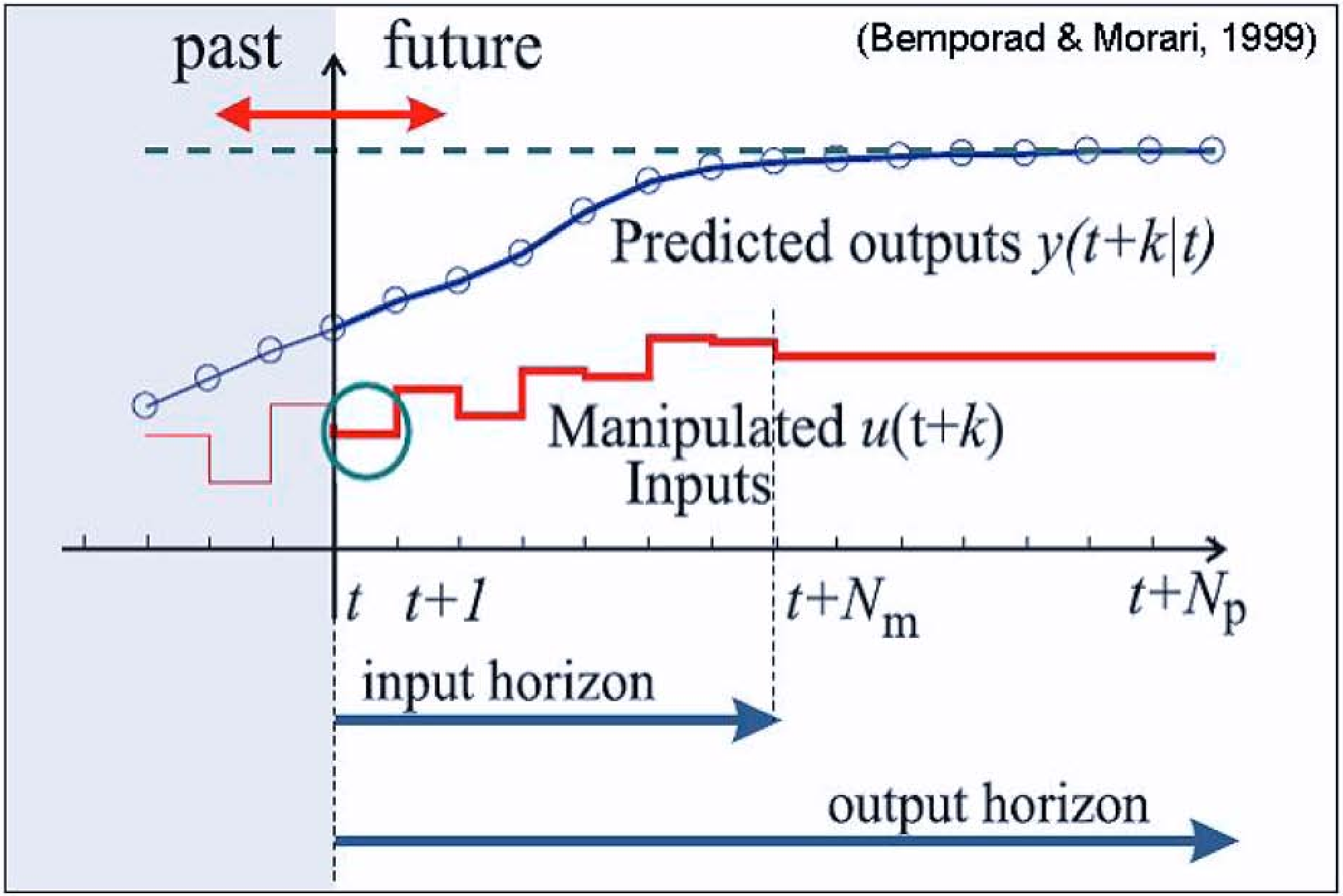
Formulation of linear MPC



- **Why MPC?**
  - It provides a systematic approach to control the multivariable dynamical systems with constraints

- **What is MPC**
  - MPC refers to class of computer algorithms that use a system model to predict the future response of the controlled plant. The prediction is the used for computation of the optimal control action.

Formulation of linear MPC



Formulation of linear MPC

- **Main parts of the MPC controller**
  - System model and predictions
  - Control problem and MPC formulation
  - Real time optimization problem

Formulation of linear MPC

- **Main parts of the MPC controller (i/iii)**
  - System model and predictions
    - The modeling stage in MPC design is one of the most important activities. The quality of the resulting controller is proportional to the model quality and therefore, the model should describe the system as accurate as possible.
  - Control problem and MPC formulation
  - Real time optimization problem

Formulation of linear MPC

- **Main parts of the MPC controller (ii/iii)**
  - System model and predictions
  - Control problem and MPC formulation
    - This is an important part which usually requires practical experiences. Sometimes, it is difficult even to identify what should be controlled and optimized. We have to know all the basic properties and limitations of MPC at this stage.
  - Real time optimization problem

Formulation of linear MPC

- **Main parts of the MPC controller (iii/iii)**
  - System model and predictions
  - Control problem and MPC formulation
  - Real time optimization problem
    - The last step is translation of the MPC control problem to a numerical optimization problem. Usually this is relatively easy part

System models and predictions

- **In the linear MPC, we can utilize any linear model**
  - Impulse response
  - Step response
  - ARX model
  - State space models

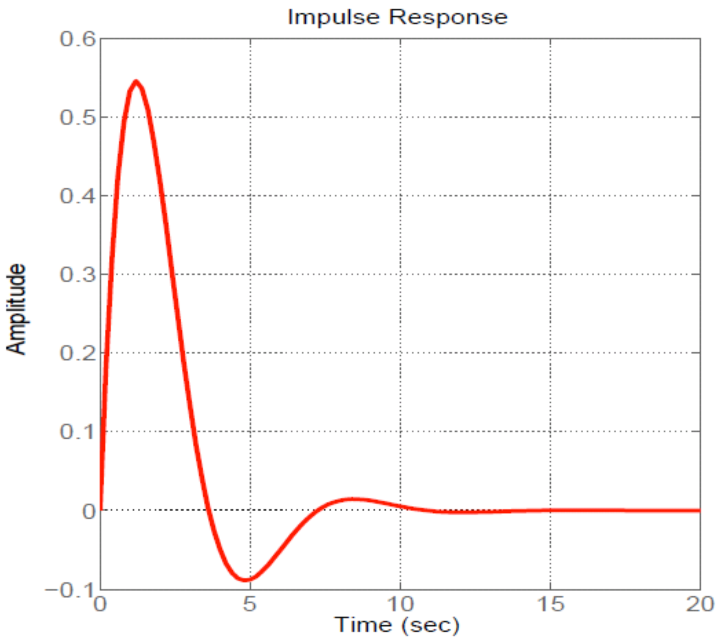
System models and predictions

- **Impulse response**

$$y(t) = \sum_{i=1}^{\infty} h_i u(t-i)$$

- **Prediction model**

$$\hat{y}(t+k \mid t) = \sum_{i=1}^{N} h_i u(t+k-i \mid t)$$



$$G(s) = \frac{1}{s^2 + s + 1}$$

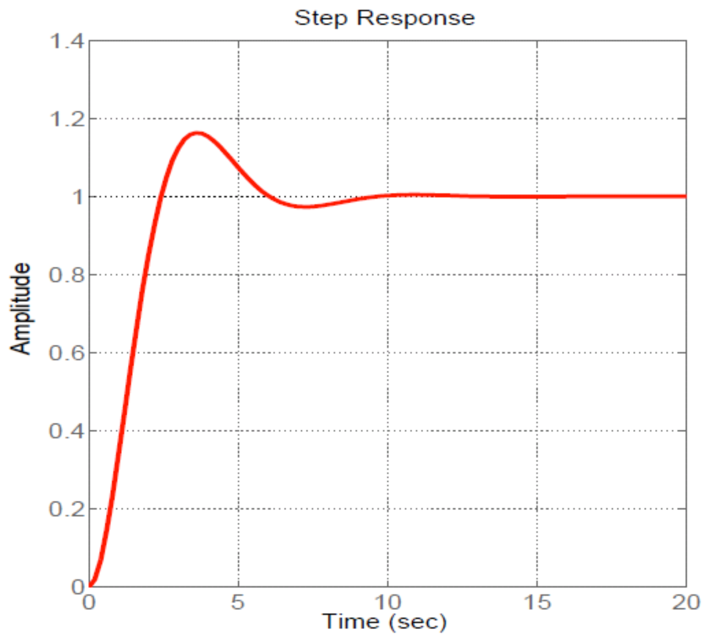
System models and predictions

- **Step response**

$$y(t) = y_0 + \sum_{i=1}^{\infty} g_i \Delta u(t-i)$$

- **Prediction model**

$$\hat{y}(t+k \mid t) = \sum_{i=1}^{N} g_i \Delta u(t+k-i \mid t)$$



$$G(s) = \frac{1}{s^2 + s + 1}$$

System models and predictions

- **ARX "based" models**

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

- **Prediction is given by**

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

$$[A_t|A_p] = \left[\begin{array}{ccccc|ccccccc} a_n & . & . & . & a_1 & 1 & 0 & . & . & . & . & 0 \\ 0 & a_n & . & . & . & a_1 & 1 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & 0 & a_n & . & . & a_1 & 1 & 0 & . & 0 \\ 0 & . & . & . & 0 & a_n & . & . & a_1 & 1 & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & 0 & . & . & a_n & . & . & a_1 & 1 \end{array}\right] \quad [B_t|B_p] = \left[\begin{array}{ccccc|ccccccc} b_n & . & . & . & b_1 & b_0 & 0 & . & . & . & . & 0 \\ 0 & b_n & . & . & . & b_1 & b_0 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & 0 & b_n & . & . & b_1 & b_0 & 0 & . & 0 \\ 0 & . & . & . & 0 & b_n & . & . & b_1 & b_0 & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & 0 & . & . & b_n & . & . & b_1 & b_0 \end{array}\right] \quad [D_t|D_p] = \left[\begin{array}{ccccc|ccccccc} d_n & . & . & . & d_1 & d_0 & 0 & . & . & . & . & 0 \\ 0 & d_n & . & . & . & d_1 & d_0 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & 0 & d_n & . & . & d_1 & d_0 & 0 & . & 0 \\ 0 & . & . & . & 0 & d_n & . & . & d_1 & b_0 & . & 0 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ 0 & . & . & . & 0 & . & . & d_n & . & . & d_1 & d_0 \end{array}\right]$$

System models and predictions

- **State space model (ideal candidate)**

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

- **Prediction**

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

$$
\begin{aligned}
\vec{y} &= \left[\, y(k)^T \;\; y(k+1)^T \;\; \cdots \;\; y(k+N-1)^T \,\right]^T , \\
\vec{u} &= \left[\, u(k)^T \;\; u(k+1)^T \;\; \cdots \;\; u(k+N-1)^T \,\right]^T ,
\end{aligned}
$$

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

System models and predictions

- **Summary:** In the linear MPC, any linear model can be utilized.

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

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

Cost function

- **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.

Cost function

- **The basic form is**

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

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

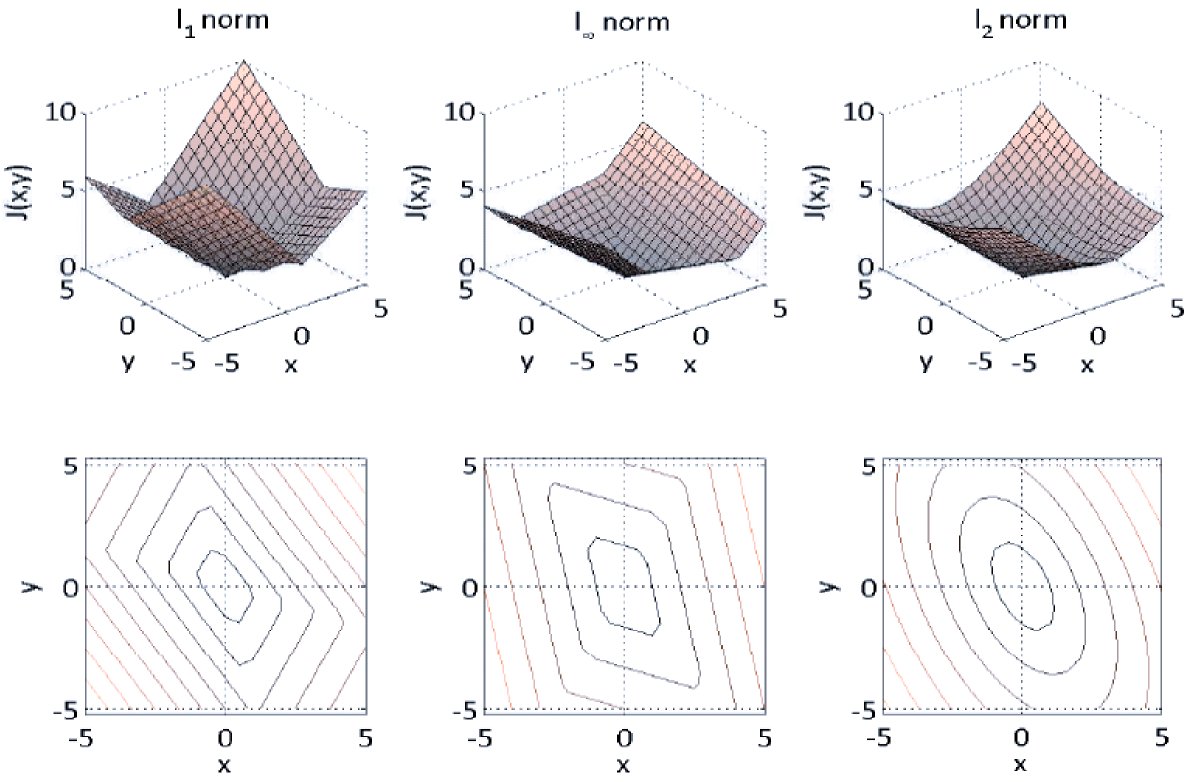
- **Another form**

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

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

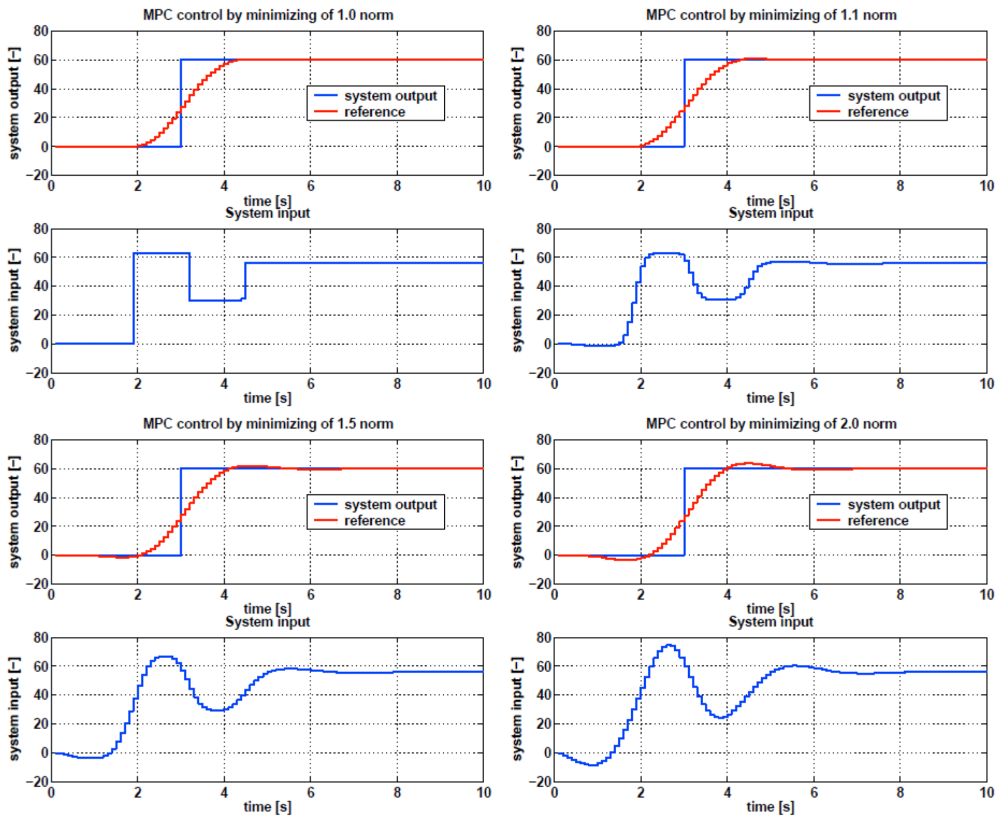
Cost function

- **Examples of different norms**



Cost function

- **Examples of different norms (p <1, 2>)**



Constraints

- **Role of constraints in MPC**
  - A real differentiator for the MPC controllers is the fact that they can handle the system constraints in 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.

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

$$\Delta u_{\min}(t) \le \Delta u(t) \le \Delta u_{\max}(t) \qquad x_{\min}(t) \le x(t) \le x_{\max}(t)$$

Constraints

- **Hard constraints**
  - Physical limitations of real process, e.g. actuator extreme positions and this type of constraints must not be violated.

- **Soft constraints**
  - These can be violated though at some penalty, for example a loss of product quality, constraints for the system “inner” variables.

Constraints

- **HARD constraints**

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

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

$$y_{\min}(t) \leq y(t) \leq y_{\max}(t)$$

$$x_{\min}(t) \leq x(t) \leq x_{\max}(t)$$

Constraints

- **SOFT constraints**

$$\cancel{u_{\min}(t) \leq u(t) \leq u_{\max}(t)\ ,}$$
$$\cancel{\Delta u_{\min}(t) \leq \Delta u(t) \leq \Delta u_{\max}(t)}$$

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

Constraints

- **SOFT constraints – implementation**
  - 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

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

  - we have to include a term to be minimized into the cost function

$$||\varepsilon||_2^2$$

Constraints

- **SOFT constraints – implementation**
  - Alternative option is to penalize the constraints violation directly in the cost function

$$\|y(t) - \varepsilon\|_2^2$$

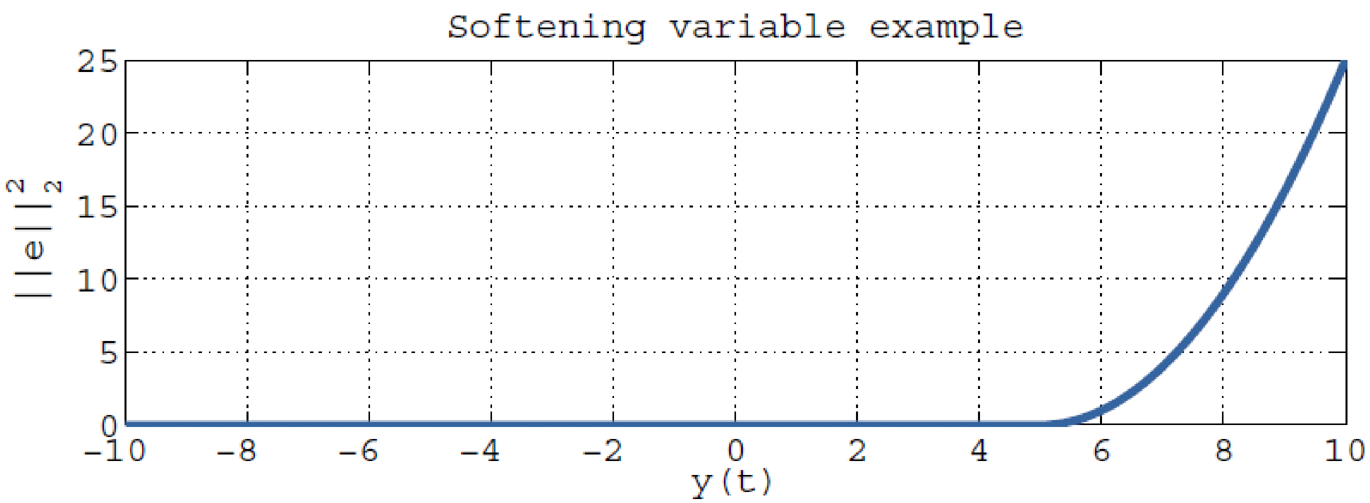
  - and we have to introduce a “box” constraint

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

Constraints

- **SOFT constraints - example**

$$y(t) \leq 5 + \varepsilon$$



MPC optimization problem

- **Basic MPC control problem can be formulated as an optimization problem**

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

- **Subject to**

- input constraints

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

- system state constraints (usually softened)

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

- output constraints (usually softened)

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

- system model equations

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

MPC optimization problem

- **The MPC problem for a linear system with the linear constraints can be transformed to mathematical programming problem of the form**

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

- **This is well known Quadratic Programming problem**

```
>> help quadprog
 QUADPROG Quadratic programming.
    X=QUADPROG(H,f,A,b) attempts to solve the quadratic programming problem:

             min 0.5*x'*H*x + f'*x   subject to:  A*x <= b
              x

    X=QUADPROG(H,f,A,b,Aeq,beq) solves the problem above while additionally
    satisfying the equality constraints Aeq*x = beq.
```

Receding Horizon Control

- **So far …**
  - The MPC control problem can be transformed to an optimization problem, which is parameterized by ‘p’ parameter vector.
  - The resulting optimization problem at time t0 is the optimal future trajectory of the system input
  - An immediate idea would be to apply the whole sequence and to compute the new trajectories at the end of prediction horizon. Such MPC control strategy corresponds to the **open loop control…**

… not a good idea

Receding Horizon Control

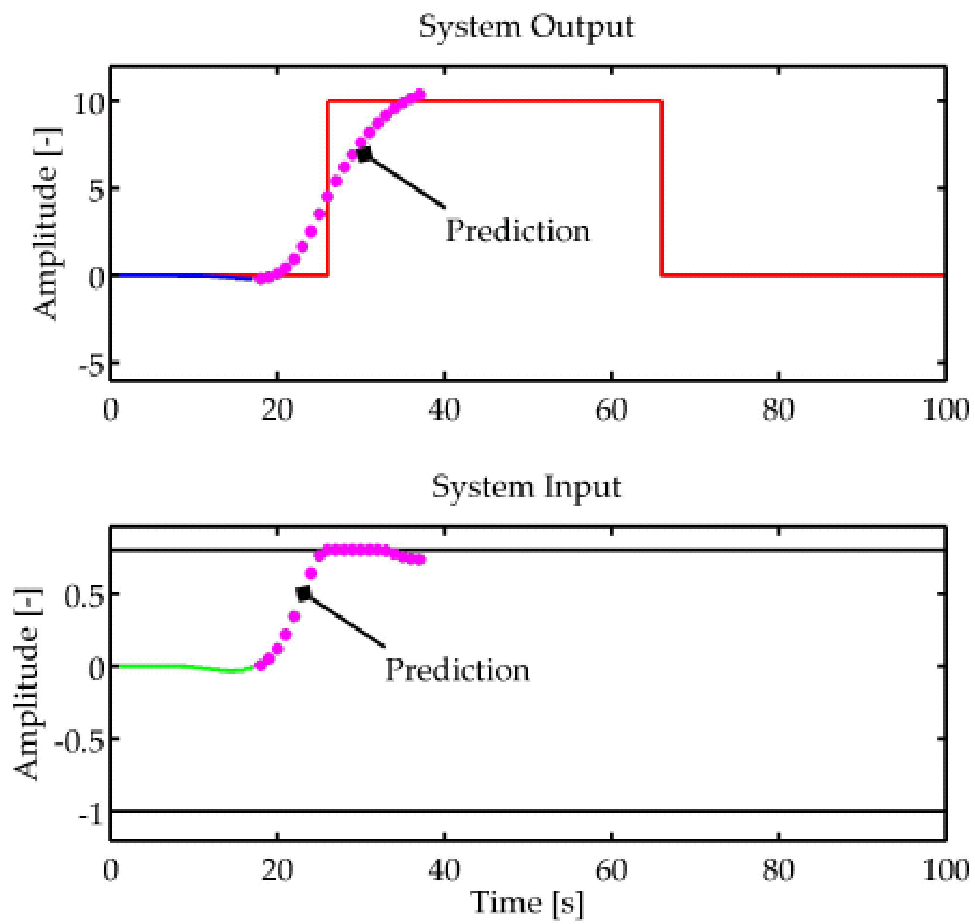
- The standard feedback, as we know it for the classical control methods, is introduced by using so called *Receding Horizon Control*.

- In the *Receding Horizon Control*, the optimization problem is computed at each sampling period after having new system measurements or estimates and we apply only the first control action from the optimal input vector. This strategy ensures the standard feedback control in the MPC.

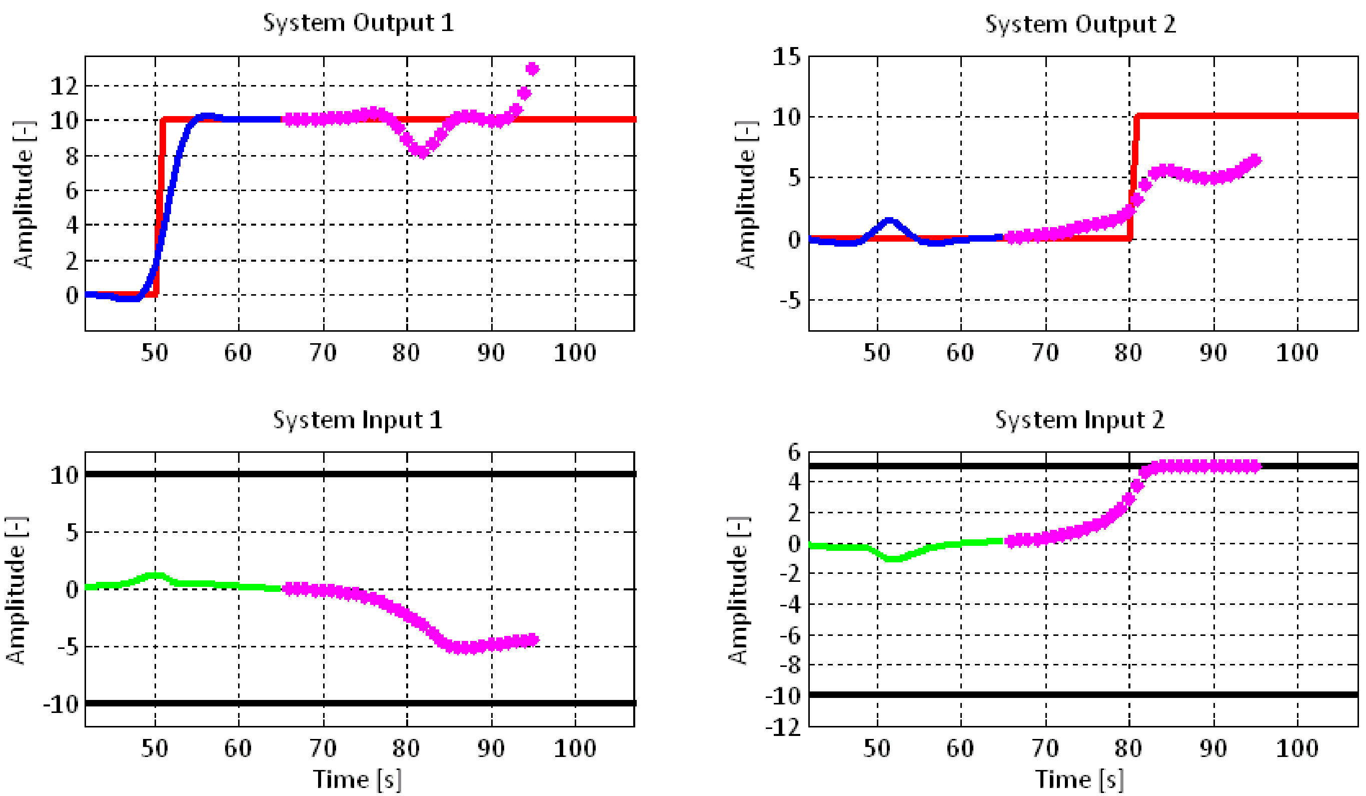
… much better

Receding Horizon Control

- **Example - SISO**



- **Example - MIMO**



Offset-free tracking

- **Offset-free tracking in classical control**
  - 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.
  - 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**?

Offset-free tracking

- **Offset-free tracking for MPC – first option**
  - 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.

Offset-free tracking

- **Offset-free tracking for MPC – second option**
  - 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. The disturbances can be estimated by using the augmented system state observer, These techniques are known as **Unknown Input Observer**.

Offset-free tracking

- **The virtual disturbances can be connected to the system in a number of ways. Consider a linear model of a controlled process**

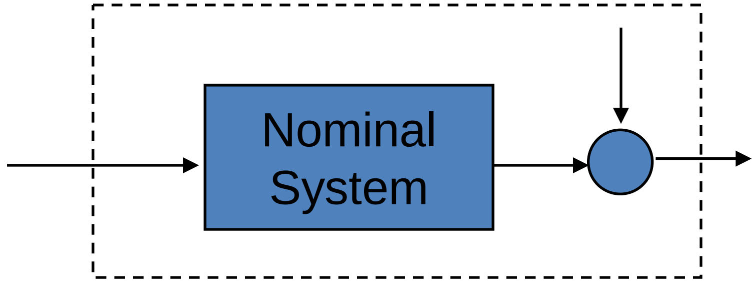
$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) \\
\widehat{y}(k) &= Cx(k) + Du(k)
\end{aligned}
$$

- **And assume a disturbance model in form**

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

Offset-free tracking

- **Disturbance acting on the system output**

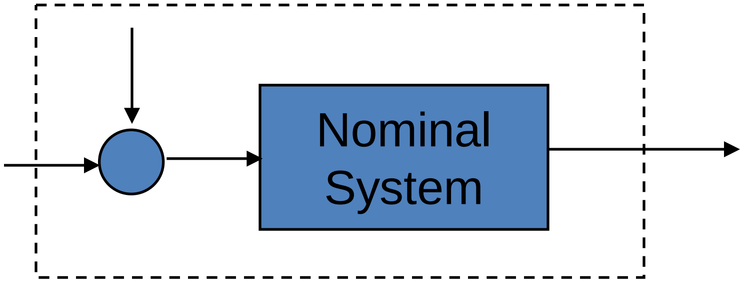


$$\begin{bmatrix} x(k+1) \\ x_d(k+1) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A_d \end{bmatrix} \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix} + Du(k)$$

Offset-free tracking

- **Disturbance acting on the system input**



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

$$y(k) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix} + Du(k)$$

Offset-free tracking

- **Summary:**
  - Can by achieved by using several formulation
  - Preferred one is the method using the exogenous disturbance model and its estimate

Agenda

- **Introduction**
  - Motivation for advanced control techniques
  - Classical approach
  - Brief history of model predictive control

- **Linear Model Predictive Control**
  - Formulation of linear MPC
  - Analysis of linear MPC
  - Hybrid systems and linear MPC
  - Optimization Algorithms for linear MPC

Analysis of linear MPC

- **Analysis of classical feedback controllers**
  - The classical feedback controllers (PID) can be analyzed in a number of ways. The most important properties are the nominal **performance**, **stability** and **robustness**.

- **What is different?**
  - 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.

Analysis of linear MPC

- **MPC control law**
  - 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**.
  - It can be shown, that we can find a control law for each combination of feasible active constraints in the form

$$u(k) = K_i x(k) + g_i$$

Analysis of linear MPC - unconstrained

- **Assume that the system can be described by a state space model, then the prediction is given by**

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

- **Assume a quadratic cost function defining the tracking MPC problem**

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

Analysis of linear MPC - unconstrained

- **Then 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**

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

- **By applying the receding horizon strategy, we can define control law in the form** $u(k)^* = -K^x x(k) + K^r \vec{r}$

Analysis of linear MPC - unconstrained

- **Example:** $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix},\ C = \begin{bmatrix} 1 & 1 \end{bmatrix},\ D = 0$

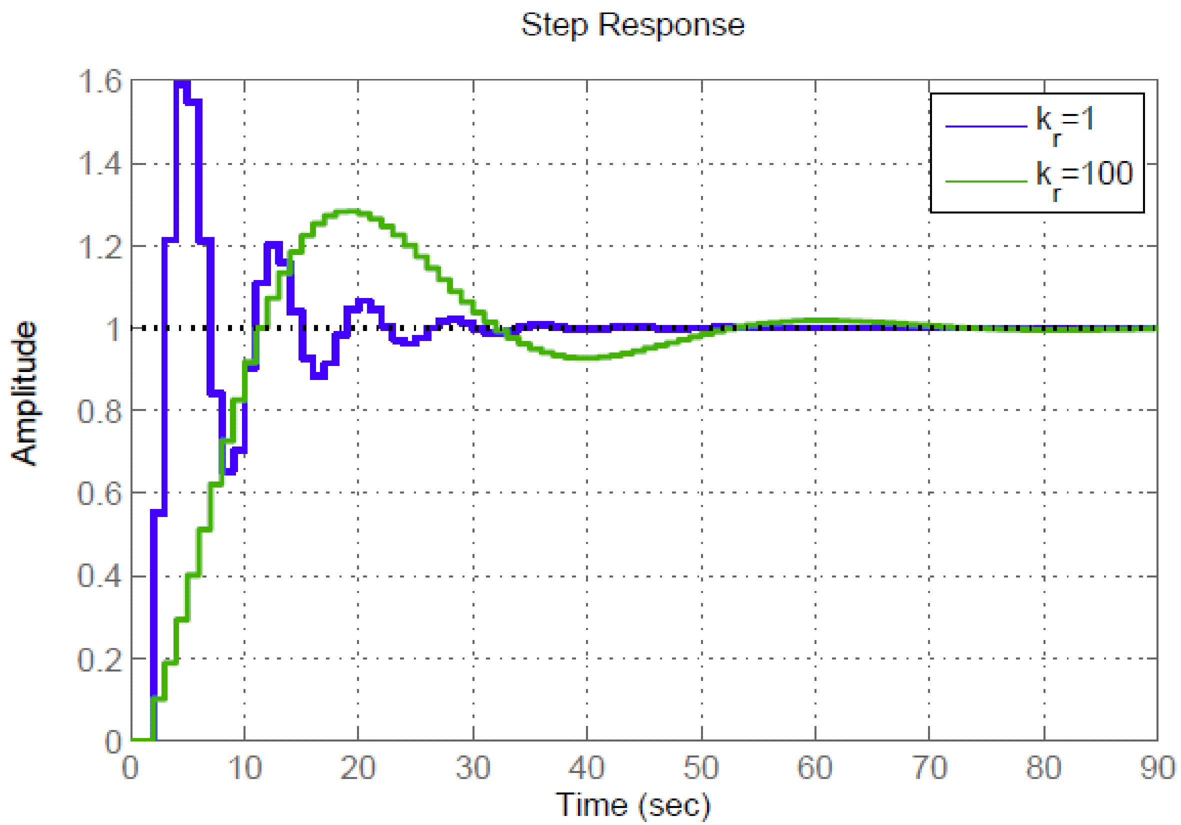
- **MPC cost function**

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

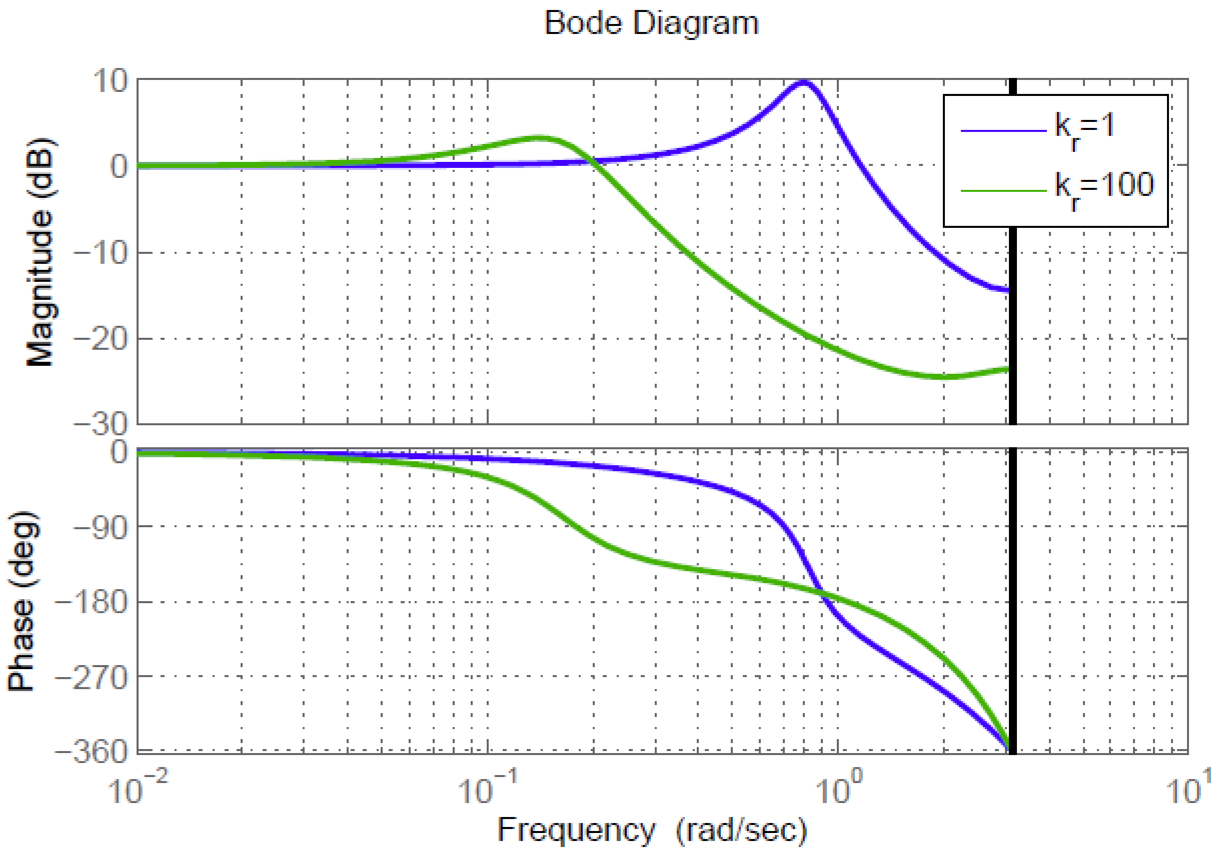
- **Parameters** $N = 10 \quad Q = I \quad R = k_r I$

- **Control law** $u(k) = -K^x x(k) + K^r \vec{r} + K^u u(k-1)$

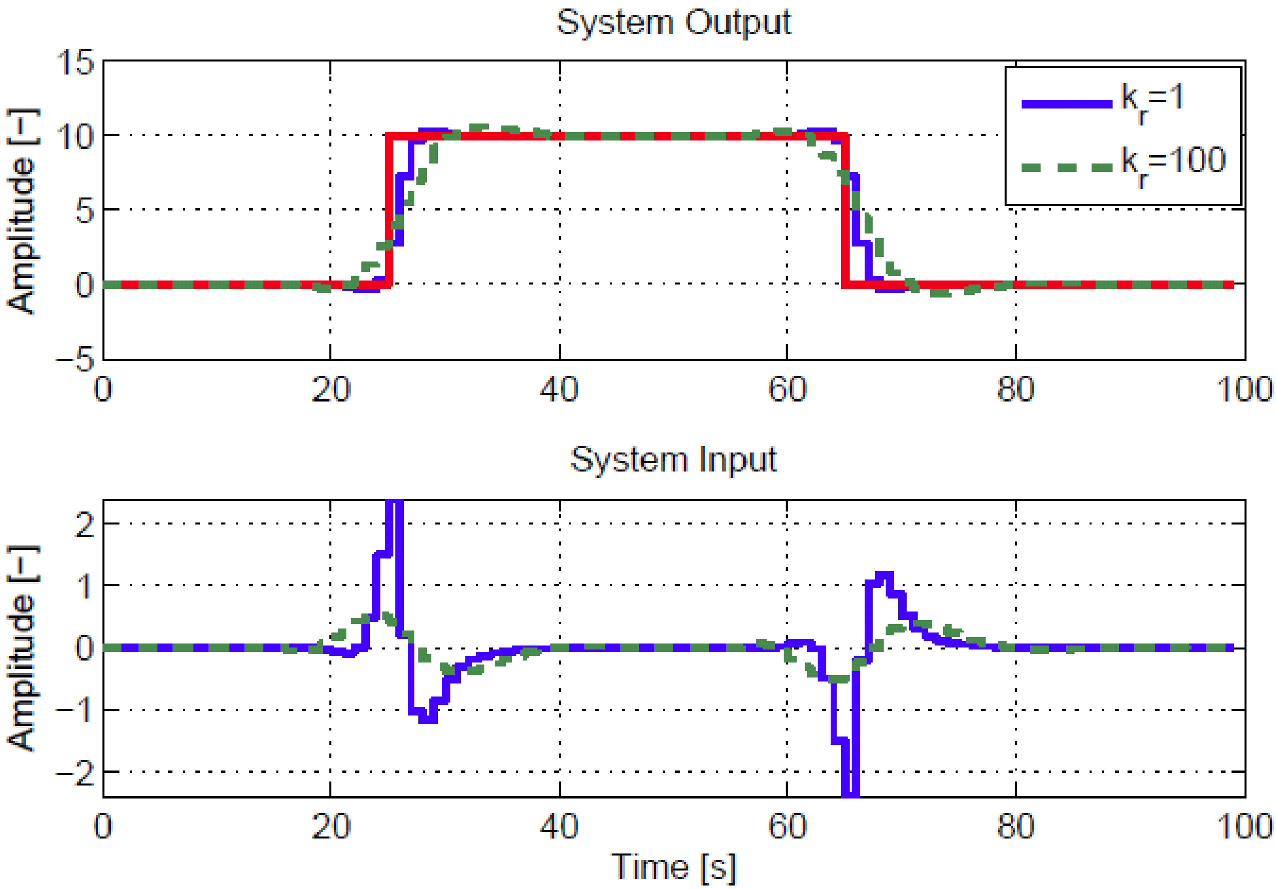
Analysis of linear MPC - unconstrained



Analysis of linear MPC - unconstrained



Analysis of linear MPC - unconstrained



Analysis of linear MPC - stability

- **Introduction**

  – 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.

Analysis of linear MPC - stability

- **Extension of the MPC prediction horizon to infinity (LQR control)**

$$J\left(u(0),\dots,u(\infty)\right)=\sum_{k=0}^{\infty}\left(x(k)^TQx(k)+u(k)^TRu(k)\right)$$

- **We can split the infinite prediction horizon into the two parts**
  - Finite: time 0 to N-1
  - Infinite: time N to inf

Analysis of linear MPC - stability

- **The quasi-infinite cost function can be written as**

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

- **where the terminal penalty term is defined as**

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

Analysis of linear MPC - stability

- **We can show that** $\Psi(x) = x^T \Psi x \qquad \Psi \geq 0$

$$
\begin{aligned}
\Psi &= A^T \Psi A - A^T \Psi B \left(R + B^T \Psi B\right)^{-1} B^T \Psi A + Q \\
K &= \left(R + B^T \Psi B\right)^{-1} B^T \Psi A\,.
\end{aligned}
$$

Analysis of linear MPC - stability

- **Dual mode control**
  - MODE 1:
    - 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:
    - the subsequent infinite horizon over which the control inputs are determined by control law: $u(k) = -K\,x(k)$. The gain matrix K is the feedback gain that ensures the unconstrained closed-loop stability

Analysis of linear MPC - stability

- **MPC stability**
  - 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.

Analysis of linear MPC - stability

- **The basic tools are:**
  - Terminal equality constraints
  - Terminal cost function
  - Terminal constraints set
- **The most practical is the combination of terminal cost function in combination with the terminal constraints set**

Analysis of linear MPC - stability

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.*

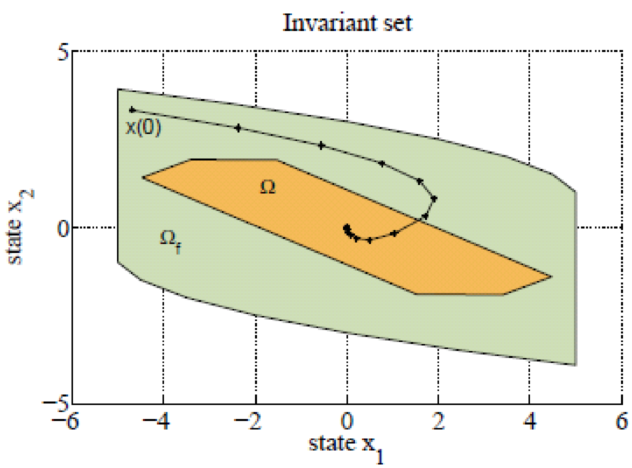
Definition 2 *An admissible positively invariant set Ω is a region of state space with the property that all state trajectories starting from an initial condition within the set remain within the set at all future instants and all considered constraints will be satisfied.*

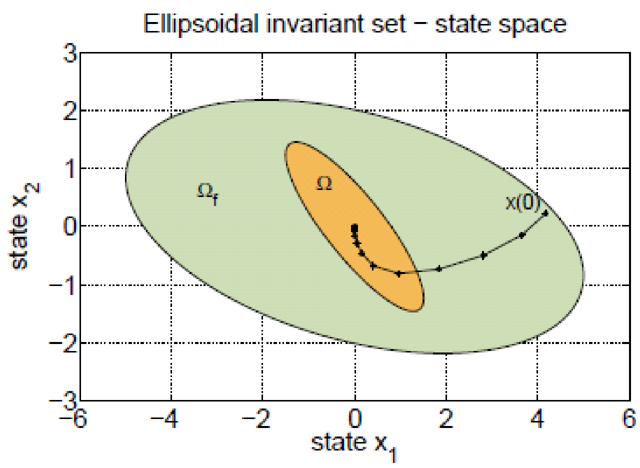
Definition 3 *The maximal admissible positively invariant set (MAS) is a region of state space of all possible initial states so that all state trajectories starting from an initial condition within the set remain within the set at all future instants and all considered constraints will be satisfied.*

Analysis of linear MPC - stability

- **Formally, the basic property of a positively invariant set can be summarized as**

$$(A - BK)\,x(k) \in \Omega \quad \forall x(k) \in \Omega$$
$$m_{\min} \leq Mx(k) \leq m_{\max} \quad \forall x(k) \in \Omega\ .$$





Analysis of linear MPC - stability

- **Definition of MPC problem ensuring the stability (stable control synthesis)**

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

$$G\vec{u} \leq W + Sx(k) \qquad x(k+N) \in \Omega$$

Analysis of linear MPC – stability proof

- **…If there exist a Lyapunov function with given properties for the given control law, then the nominal closed loop will be stable…**

- **A candidate for the Lyapunov function is the cost function, i.e.**

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

Analysis of linear MPC – stability proof

- **Assume three different control sequences**

  – Optimal control sequence at time k

$$\vec{u}_k^* = \left[\; u^*(k|k) \;\; u^*(k+1|k) \;\; \ldots \;\; u^*(k+N-1|k) \;\right]$$

  – Suboptimal control sequence at time k+1

$$\vec{u}_{k+1}^{shift} = \left[\; u(k+1|k) \;\; u(k+2|k) \;\; \ldots \;\; u(k+N|k) \;\right]$$

  – Optimal control sequence at time k+1

$$\vec{u}_{k+1}^* = \left[\; u^*(k+1|k+1) \;\; u^*(k+2|k+1) \;\; \ldots \;\; u^*(k+N|k+1) \;\right]$$

Analysis of linear MPC – stability proof

- **From definition, we have**

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

- **Then**

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

Analysis of linear MPC – stability proof

- **It holds that** $V(k) = J(\vec{u}_k^*|x(k), k)$
- **Therefore**

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

- **Finally, the Lyapunov function must satisfy**

$$V(k+1) - V(k) \leq 0$$

Analysis of linear MPC – stability proof

- **It is clear that the non-increasing condition will be satisfied if**

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

- **If the Mode 2 control law satisfies the above condition, then V(k) is a Lyapunov function and the receding horizon control sequence will stabilize the system.**

Analysis of linear MPC – stability proof

- **Two basic options for Mode 2:**

  - $u(k+i) = 0$, $i \geq N$: 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.

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

    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.

Analysis of linear MPC – stability

- **Summary:**
  - The MPC control stability can be ensured by defining the MPC control problem as Dual mode control
  - To complete the MPC analysis, we have to show that the optimization control problem in the MPC Mode 1 is feasible at any time – this is due to presence of constraints. It can be shown that if the optimization problem is feasible at time t=0, then it remains feasible for all times.

Analysis of linear MPC – robustness

- **Robust MPC control 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.

Analysis of linear MPC – robustness

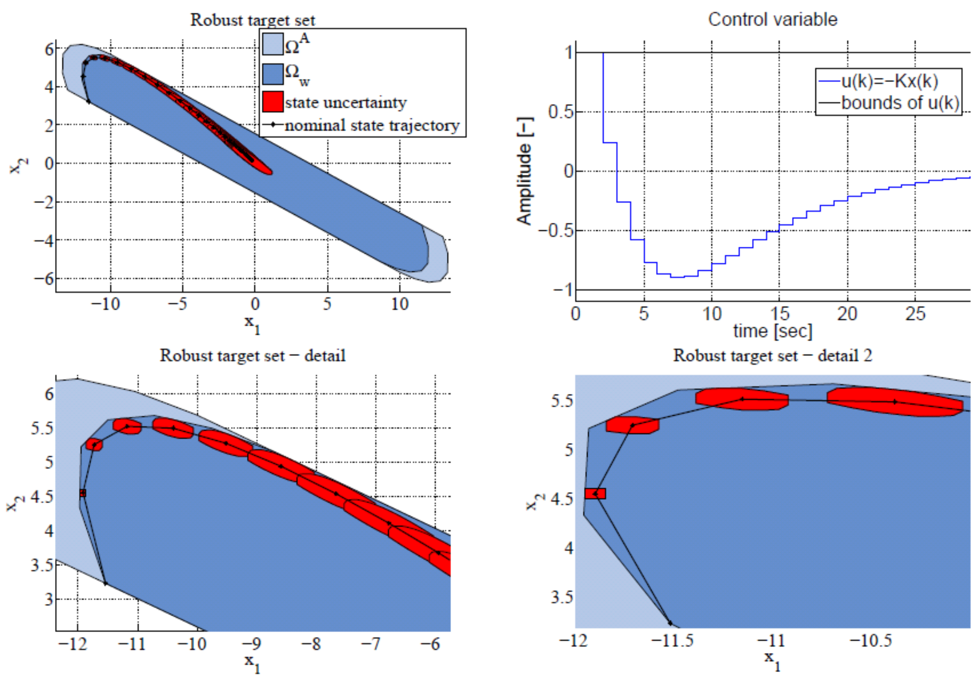
- **Robust admissible positively invariant set**

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.*

Analysis of linear MPC – robustness

- **Robust admissible positively invariant set - example**

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



Analysis of linear MPC – robustness

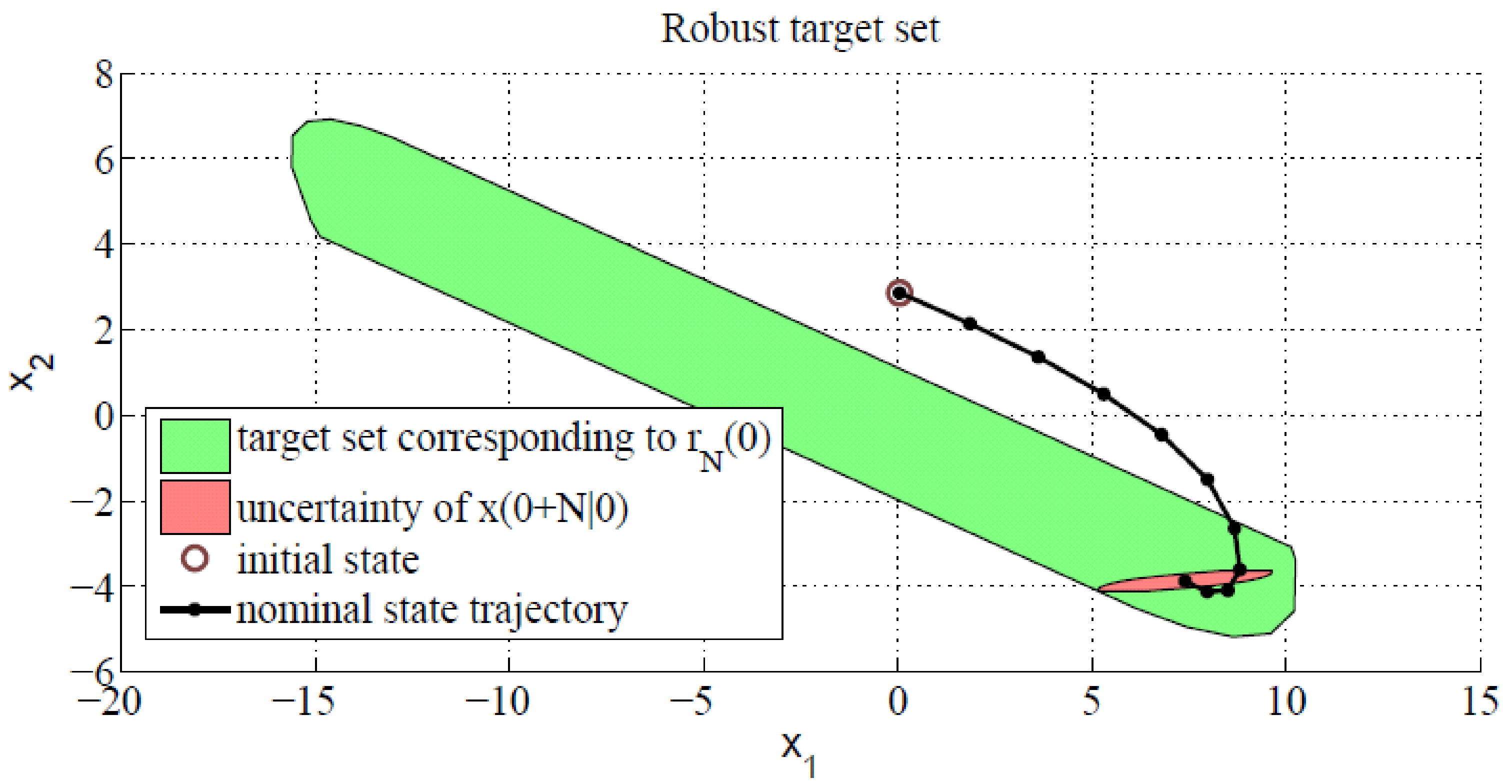
- **Basic robust MPC optimization problem can be defined as min-max optimization**

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

$$\vec{u}_k \in \mathcal{U}_k \qquad \mathcal{X}_k^N \in \Omega_{robust} \qquad x(k+N) \in \mathcal{X}_k^N$$

Analysis of linear MPC – robustness

- **Robust MPC - example**



Agenda

- **Introduction**
  - Motivation for advanced control techniques
  - Classical approach
  - Brief history of model predictive control

- **Linear Model Predictive Control**
  - Formulation of linear MPC
  - Analysis of linear MPC
  - Hybrid systems and linear MPC
  - Optimization Algorithms for linear MPC

Hybrid systems and linear MPC

- **Definition**
  - 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

Hybrid systems and linear MPC

- **An example of a hybrid system is a piece-wise affine (PWA) system, which can be 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

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

Hybrid systems and linear MPC

- **Properties**
  - 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. These models transform the logical part of a hybrid system into the mixed-integer linear inequalities by using Boolean variables.

Hybrid systems and linear MPC

- **The basic form of the MLD system is given by**

$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) + B_2\delta(k) + B_3 z(k)\ , \\
y(k) &= Cx(k) + Du(k) + D_2\delta(k) + D_3 z(k)\ ,
\end{aligned}
$$

subject to

$$
E_2\delta(k) + E_3 z(k) \leq Eu(k) + E_4 x(k) + E_5\ ,
$$

  – where x(k) is a combined continuous and binary state, u(k) and y(k) are the system input and outputs (continuous and binary), delta(k) are auxiliary binary variables and z(k) are auxiliary continuous variables.

Hybrid systems and linear MPC

- **We can define the optimal control problem for the PWA system as**

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

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

- **Summary:**
  - 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
    - if the the cost function is based on l1 or linfty norm, the optimization problem leads to Mixed-Integer Linear Programming.

Agenda

- **Introduction**
  - Motivation for advanced control techniques
  - Classical approach
  - Brief history of model predictive control

- **Linear Model Predictive Control**
  - Formulation of linear MPC
  - Analysis of linear MPC
  - Hybrid systems and linear MPC
  - Optimization Algorithms for linear MPC

Optimization algorithms for linear MPC

- **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**

Optimization algorithms for linear MPC

- **Assume the resulting QP optimization problem for the linear MPC in form**

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

- **Modern QP solvers are based on**
  - Active set approach
  - Interior point approach

Optimization algorithms for linear MPC

- **Active set solvers**
  - Iterative algorithms where 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.

Optimization algorithms for linear MPC

- **Interior point solvers**
  - 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.

Optimization algorithms for linear MPC

- **Multi-parametric explicit solution**
  - If we need extremely fast sampling periods in the MPC, we can use multi-parametric explicit solution. 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 multi-parametric explicit solution is applicable for small systems only due to storage demands.

Optimization algorithms for linear MPC

- **Tailoring for MPC**
  - 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 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.

Agenda

- **Nonlinear Model Predictive Control**
  - Formulation of nonlinear MPC
  - Analysis of nonlinear MPC
  - Numerical methods for nonlinear MPC

- **Practical Model Predictive Control**
  - Practical formulations of MPC
  - Development cycle of industrial MPC
  - Demonstration of an application

Formulation of nonlinear MPC

- **Why nonlinear MPC?**
  - 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.

Formulation of nonlinear MPC

- **The modeling phase in the NMPC is the most important part. Assume a model described by a set of nonlinear equations**

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

Formulation of nonlinear MPC

- **The basic cost function has the integral form**

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

- **where the function L(.) defines the control objectives. For example it can be defined as**

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

Formulation of nonlinear MPC

- **The cost function on the infinite prediction horizon can be divided into two parts**
  - finite part

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

  - infinite part

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

Formulation of nonlinear MPC

- **The general nonlinear MPC can be formulated as a nonlinear optimization problem**

$$\min_{u(t)} J\left(u(t) \middle| x(t_0)\right)$$

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

Agenda

- **Nonlinear Model Predictive Control**
  - Formulation of nonlinear MPC
  - Analysis of nonlinear MPC
  - Numerical methods for nonlinear MPC

- **Practical Model Predictive Control**
  - Practical formulations of MPC
  - Development cycle of industrial MPC
  - Demonstration of an application

Analysis of nonlinear MPC - stability

- **Stability**
  - The dual mode control strategy, as it was presented in the section about linear MPC, can be extended also for the nonlinear systems.
  - Assume a cost function of the form and assume that it is a candidate for the Lyapunov function

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

Analysis of nonlinear MPC - stability

- **Stability condition**

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:*

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

Agenda

- **Nonlinear Model Predictive Control**
  - Formulation of nonlinear MPC
  - Analysis of nonlinear MPC
  - Numerical methods for nonlinear MPC

- **Practical Model Predictive Control**
  - Practical formulations of MPC
  - Development cycle of industrial MPC
  - Demonstration of an application

Numerical methods for nonlinear MPC

- **Basic approach**
  - A commonly used approach to solve the problem 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. 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 .

Numerical methods for nonlinear MPC

- **Two important direct approaches (i/ii)**

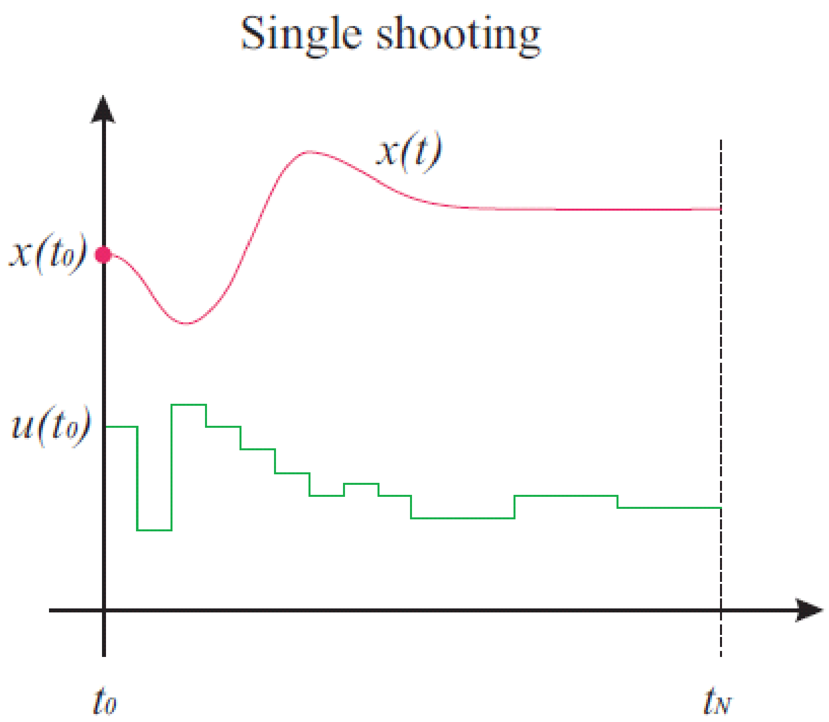
- **Direct Single Shooting**
  - Basic approach and is similar to the approach used by the standard linear model predictive control.
  - At the initial time, the numerical integration is used to obtain the predicted trajectories as a function of manipulated variable for the prediction horizon.
  - Having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

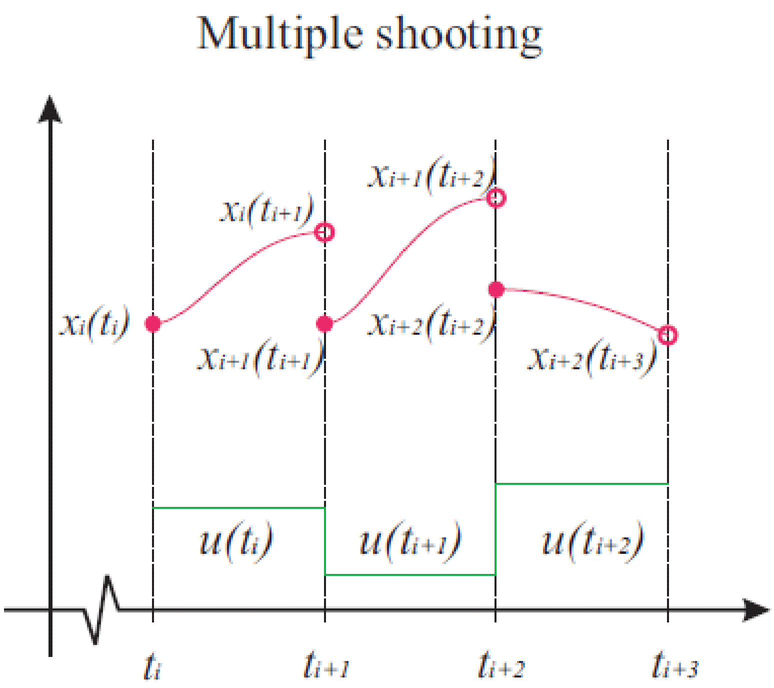
Numerical methods for nonlinear MPC

- **Two important direct approaches (ii/ii)**

- **Direct Multiple Shooting**
  - 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. The optimization problem is then augmented by auxiliary constraints - continuity conditions.

Numerical methods for nonlinear MPC





Numerical methods for nonlinear MPC

- **Multiple shooting**
  – 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 sub-problem 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.

Numerical methods for nonlinear MPC

- **The key idea of parameterization is to find the sensitivity matrices**

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

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

Numerical methods for nonlinear MPC

- **The sensitivity matrices can be found as**

$$\dot{\Phi}(t,t_0)=\frac{\partial f\left(x(t),u(t)\right)}{\partial x(t)}\Phi(t,t_0)\ ,\qquad \Phi(t,t_0)=\frac{\partial x(t_0)}{\partial x(t_0)}=I$$

- **and**

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

$$1(t-t_0)=\begin{cases}0 & t<t_0\\ 1 & \text{otherwise}\end{cases}\qquad \Gamma(t,t_0)=\frac{\partial x(t_0)}{\partial u(t_0)}=0$$

Numerical methods for nonlinear MPC

- **The final optimization problem is formulated as**

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

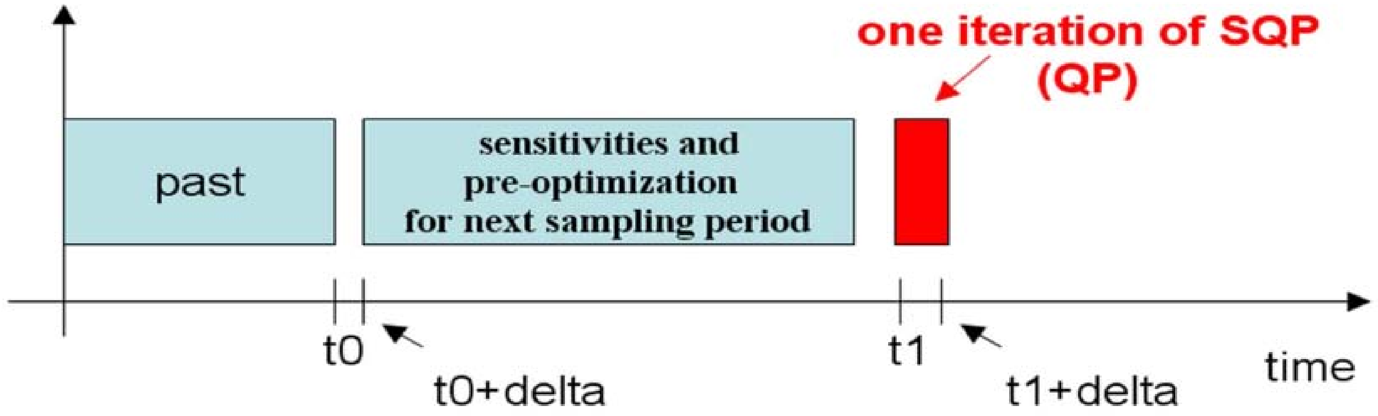
$$L_i\left(x_i(t_i), u(t_i), t_i\right) = \int_{t_i}^{t_{i+1}} L\left(x_i(\tau), u(\tau), \tau\right) d\tau$$

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

Numerical methods for nonlinear MPC

- **Iterative scheme**
  - 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.
  - 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.
    - feedback phase takes new measurement and calculates the control action that can be immediately sent to the system.

Numerical methods for nonlinear MPC



Agenda

- **Nonlinear Model Predictive Control**
  - Formulation of nonlinear MPC
  - Analysis of nonlinear MPC
  - Numerical methods for nonlinear MPC

- **Practical Model Predictive Control**
  - Practical formulations of MPC
  - Development cycle of industrial MPC
  - Demonstration of an application

Discussion

References (1/8)

[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.

References (2/8)

[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.

References (3/8)

[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.

References (4/8)

[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.

References (5/8)

[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.

References (6/8)

[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.

References (7/8)

[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.

References (8/8)

[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.