DISC Course, Lecture Notes

Model Predictive Control

Ton J.J. van den Boom & Ton C.P.M. Backx

October 1, 2003

Model Predictive Control

dr.ir. Ton J.J. van den Boom
Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, NL-2628 CD Delft, The Netherlands
Tel. (+31) 15 2784052   Fax. (+31) 15 2786679
Email: T.J.J.vandenBoom@dcsc.tudelft.nl

prof.dr.ir. Ton C.P.M. Backx
Measurement and Control Section
Measurement and Control Systems Group
Faculty of Electrical Engineering
Eindhoven University of Technology
P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands
Tel. (+31) 40 2473577   Fax. (+31) 40 2434582
Email: A.C.P.M.Backx@ele.tue.nl

Contents

Abbreviations

CARIMA model	controlled autoregressive integrated moving average
CARMA model	controlled autoregressive moving average
DMC	Dynamic Matrix Control
EPSAC	Extended Prediction Self-Adaptive Control
GPC	Generalized Predictive Control
IDCOM	Identification Command
IIO model	Increment-Input Output model
IO model	Input Output model
LQ control	Linear-Quadratic control
LQPC	Linear-Quadratic predictive control
MAC	Model Algorithmic Control
MBPC	Model Based Predictive Control
MHC	Moving Horizon Control
MIMO	Multiple-Input Multiple-Output
MPC	Model Predictive Control
PFC	Predictive Functional Control
QDMC	Quadratic Dynamic Matrix Control
RHC	Receding Horizon Control
SISO	Single-Input Single-Output
SPC	Standard Predictive Control
UPC	Unified Predictive Control

Chapter 1

Motivation of Model Predictive Control for industry

1.1 Main drivers for MPC application in the processing industries

Processing Industries are facing an enormous challenge: World wide competition, production capacities that exceed market demand and tightening legislation on the consumption of natural resources and on eco-sphere load enforce industries to produce products at demand. Products that are produced without having been ordered by customers block capital. These stored products have a huge negative impact on the financial performance of the company. The money earned with the money invested in the company approaches critically low limits. The main reason for this is stemming from the mechanism in a free economic market that product prices result from the balance between demand and availability in the market. In a saturated market competition between suppliers of products is heavy. If products meet imposed specifications, price will become the main discriminating factor to get a customer to decide to buy the product from a given supplier. After optimization of production, costs will level off at some minimum level that can hardly be further reduced without a further breakthrough in the applied production processes. Market saturation will continue to put pressure on prices and as a result the margins that can be made on the product will become small. The consequence of all this is that the amount of money earned with the capital invested in the production facilities and production, the capital productivity, goes down. The gradual decline of the productivity of invested capital over the past three decades is a major problem of a significant part of the processing industries. The average financial performance of many companies belonging to the processing industries has approached dangerously low levels, which may make it hard for these companies to compete with industries that do well like for example the Information and Communication Technology oriented industries. An interesting parallel can be observed between this situation that the processing industries are facing now and the situation of the Consumer Electronics and Automotive Industries around 1980. The

Consumer Electronics Industries and the Automotive Industries also saw their performance rapidly going down despite a sequence of successive re-organizations oriented to reduction of costs. The industries that survived this critical situation and that regained momentum, are the ones that completely turned around their way of working from a supply driven production organization with large buffers between production and customer supply to a market driven organization with production largely on orders received from customers. The ones that did not survive are the ones that did not follow in this major turnaround. The solutions created by the Consumer Electronics and Automotive Industries can of course not be copied directly by the processing industries. Market driven operation is far more complex to be realized in a processing industry -with its process inertia that spans several decades on a time scale- than in industries that primarily have to focus on logistics and supply chain only. Processing industries are in addition confronted with further complicating factors related to tightening operating constraints imposed upon production sites in terms of required reduction of consumption of energy, raw materials and natural resources at one hand and in terms of required reduction of eco-sphere load at the other hand. The constraints imposed upon production result in increasing complexity of processes and of their operations. More sophisticated operation support systems will therefore be required to exploit freedom available in process operation [8], [47], [81]. Many of the processing industries are still operating their production facilities in a supply driven mode of operation. This implies that no direct connection exists in these companies between actual market demand and actual production. Products are often produced cyclically in fixed sequences. Delivery of orders is handled from stock of finished products or from intermediates that only require finishing. Ongoing developments in the processing industries at the turn of the millennium show that many of the smaller operating companies have been taken over by larger ones. Two tendencies can be observed:

- Companies that try to further reduce operating costs by minimizing the number of different product types produced at a production site and by extending the production of well selected sites using the *'economy of scale'* principle. These companies focus on minimization of the fixed cost component of the product price and try to drive towards minimization of total costs of the operation. Flexibility is realized by ensuring that sufficient production plants are available to cover the variety in market demand. Due to the limited number of different products per plant relatively short production cycles can be applied, which reduces the amount of finished products that have to be kept in stock to supply the market. Each production site can only produce a limited and a small range of products. This implies that flexibility to respond to actual market demand and especially to changes in market demand is very limited.

- Companies that try to significantly increase their flexibility in producing and processing a wide range of products at their sites and that attempt to move to production at demand. These companies improve their financial performance by minimization of stock, by increase of their flexibility to adapt to market demand, even if new product specifications are requested, by maximization of margins and by reduction of the capital turnaround cycle time related to capital invested in products and intermediates.

The first category of companies produce products clearly at lowest costs initially as they can realize a lean operation with minimum overhead costs and no significant investments in upgrade of their operation support technologies. Longer term it will appear however that the average residence time of products in warehouses will be long in comparison with the average residence time of products in warehouses for the second category of companies due to the remaining lack of flexibility to directly link production to market demand. Also margins will continuously be under extreme pressure for a significant part of the volume produced due to market saturation effects and due to mismatch between market demand and supply from stored products. The average capital turnaround cycle time, although improved due to the limitation of the number of grades produced per plant, will remain poor. This will continue to put pressure on the ultimate business results of these companies. Companies belonging to the second category are the ones that are setting the scene for turning around the way of working in the processing industries. These companies are doing exactly the same thing as the ultimately successful companies in the Consumer Electronics and Automotive Industries did: Operate production directly driven by market demand to the extent feasible. These companies are facing tough times however as their total production costs, due to their focus on flexibility, initially appear to be higher. They have to make significant investments in adapting their production equipment and instrumentation to enable the flexible operation. Ultimately, these companies will see their overall performance rapidly improve. These improvements are due to the increase of capital turnaround, the better margins they can realize related to improved flexibility, their ability to better adapt to changing market conditions and their capability to timely deliver at (changing) specifications and varying volumes of product demand. The influence of reduction of the capital turnaround cycle on business performance is showing interesting characteristics. Defining capital productivity (C_{pr}) by

$$C_{pr} = \frac{M}{K \cdot T_{cycle}} \tag{1.1}$$

with:

$$\begin{aligned}
C_{pr} &= \text{Capital productivity per year} \\
M &= \text{Total margin realized per production cycle} \\
K &= \text{Capital \textit{'consumed'} during the production cycle for production} \\
&\quad\ \text{and for enabling production} \\
T_{cycle} &= \text{Length of the production cycle in years}
\end{aligned}$$

The capital productivity can now be calculated as follows:

$$C_{pr} = \frac{n\,T_h\left((V_h - C_{in})\,P - F - F_{eq}\right)}{n\,(T_h + T_w)(P\,C_{in} + F + F_{eq})\,T_{cycle}} + \frac{n\,T_w\left((V_w - C_{in})\,P - F - F_{eq}\right)}{n\,(T_h + T_w)(P\,C_{in} + F + F_{eq})\,T_{cycle}} \tag{1.2}$$

with

$$P = \text{average production rate [kg/hr]}$$

Variable	**Value**	**Dimension**
P	11500	[kg/hr]
F	1400	[EUR/hr]
F_{eq}	3000	[EUR/hr]
V_h	0.72	[EUR/kg]
V_w	0.46	[EUR/kg]
C_{in}	0.29	[EUR/kg]
T_h	40	[hr]
T_w	8	[hr]
n	1-30	[-]
T_{cycle}	$n(T_h + T_w)$	[hr]

Table 1.1: Overview of the assumed values for calculation of the capital productivity

F = fixed costs related to operation of the equipment (e.g. salaries, maintenance, overhead,) [EUR/hr]

F_{eq} = fixed costs related to depreciation of equipment and interest paid on capital invested in equipment [EUR/hr]

V_h = market value of high spec product [EUR/kg]

V_w = market value of wide spec product [EUR/kg]

C_{in} = variable costs related to input materials, energy costs, etc. [EUR/kg]

T_h = average run time of a specific product [hr]

T_w = average transition time between subsequent products. During this transition time wide spec product is produced [hr]

n = number of product types in a total production cycle [-]

Taking as an example an average size polyethylene production plant this capital productivity can be calculated as a function of the number of product grades produced within a production cycle. Assuming the conditions given in table 1.1 the capital productivity as a function of the number of product types in a total production cycle is given in fig. 1.1.
The underlying assumption made in this calculation is that, on the average, production of a total production cycle is stored before it is sold. The capital invested in products is released and the margin is made after completion of a full production cycle. In case flexibility is maximally increased to enable production directly on demand, the capital productivity realized is equal to the capital productivity corresponding with a production cycle of one grade only as a limit. The increase in overall capital productivity results from the faster turnaround of capital invested in intermediates and finished products. Although perhaps small the margin made on the capital invested in production can be made more times per

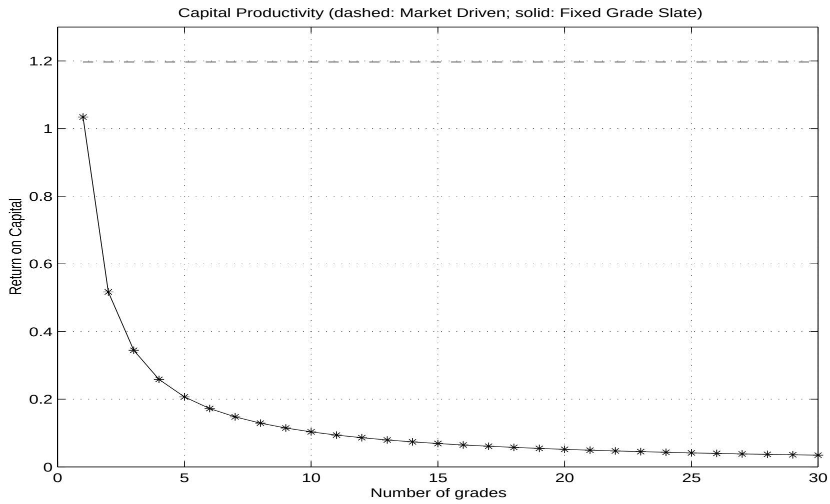

Figure 1.1: Capital productivity as a function of the number of product grades in a production cycle

year thus resulting in the better business performance. In general, capital productivity can be further increased, if flexibility in operation of the plant supports production at market demand due to better margins that can be obtained for the products produced. This further potential for improvement stems from the market mechanism that products that have high demand can be sold at better prices than products that are abound. The average margin improvement due to this market mechanism will easily be a few percent of the market price of the products. The red line in fig. 1.1 shows the capital productivity for an assumed additional average margin improvement of only 0.1 percent of the market value of the product due to this mechanism. As can be seen from this figure, a significant improvement of performance results when the overall cycle time is significantly reduced. Focusing operations on enabling market driven operation of processes opens up this opportunity amongst others. It ultimately ensures that products are almost directly delivered to customers after production so enabling shortest possible capital turnaround and significantly improved capital productivity. Market driven process operation puts extremely high requirements on predictability and reproducibility in process operation. One needs to be able to produce products at adjustable specifications in predefined, tight time slots and in changing volumes. Flexibility and timing are key parameters that drive performance. Technologies that support such process operation have to provide the functionality to operate processes this way. The following problem statement summarizes the problems faced by process industries to turnover production control from supply driven process operation to market driven process operation:

Given an industrial scale production plant that forms one link in a supply chain, provide the process operation support technologies for this plant that:

- *Enable operation of the plant in such a way that imposed operating constraints related to safety, ecology, plant lifetime and plant economics are always satisfied*
- *Continuously drive the plant towards operating conditions that comply with supply chain optimum operation within a pre-defined, feasible operating envelope for the plant*
- *Operate the plant in accordance with process operating conditions that push for maximization of capital productivity of the company the plant belongs to.*
- *Exploit remaining freedom in plant operation to maximize capital productivity of the plant over plant lifetime*

This problem definition clearly reveals that a set of sub-problems needs to be resolved that interfere with one another:

- Optimization of supply chain operation
- Optimization of overall capital productivity of the company the plant is part of
- Window over which the optimization is done together with the weighting applied over this window
- Restricted operating envelope and its resulting limitations to business performance

As a result of market saturation a trend is observed towards increasing demands with respect to tailor-made products that have to be produced with existing installations and process equipment. At the same time products often have to meet tighter specifications. Furthermore product life-time (the time between first introduction of the product on the market and the moment it is no longer produced, while it is outdated and replaced by next generation products) is getting shorter. This innovation requires flexibility in the use of existing production facilities for the production of current and new products with minor adjustments in the process hardware only. Due to the high investments involved and the long lead times for the construction and startup of new plants, plants will have to be designed to enable production of a wider range of products. The operation and control of these plants has to enable production of the mix of tightly specified products in accordance with actual market demand. In the saturated market situation with global competition it still is possible to make good profit. Necessary conditions to keep capital productivity out of the danger zones are:

- Production needs to be completely predictable and reproducible to ensure that orders can be produced and delivered at specifications, with minimum production losses, at minimum costs and just in time to minimize blocking of capital in intermediates and finished products

- Production needs to have the flexibility to produce exactly those products, that have high market demand, at the right time and in the right volume in order to maximize margins

Both conditions require tight control of production processes over broad operating ranges. The control systems applied need to exploit freedom in process operation to meet quality requirements on products and processes, to drive processes optimally to desired operating conditions along given trajectories and to minimize overall cost of operation. All freedom available in plant operation must be used for driving the plant continuously to the operating conditions that best comply with a selected balance of mutually conflicting objectives. Industries are frequently forced to operate with low margins and nevertheless remain profitable in order to survive. Reliable and profitable operation of production environments with low margins and under mentioned quality and flexibility constraints implies a need for operation of processes in a very predictable way. Detailed knowledge of dynamic behavior of critical units, depicted in the form of a mathematical model, and extensive use of this knowledge for operation and control is crucial to meet the indicated requirements. Model Predictive Control is a control technology that supports most of the above mentioned requirements as will become clear in this text. It uses known characteristics of the dynamic behavior of processes to feedforwardly drive the processes to the desired conditions along given trajectories. Unknown characteristics of process behavior are compensated by traditional feedback control (cf. Fig. 1.2).

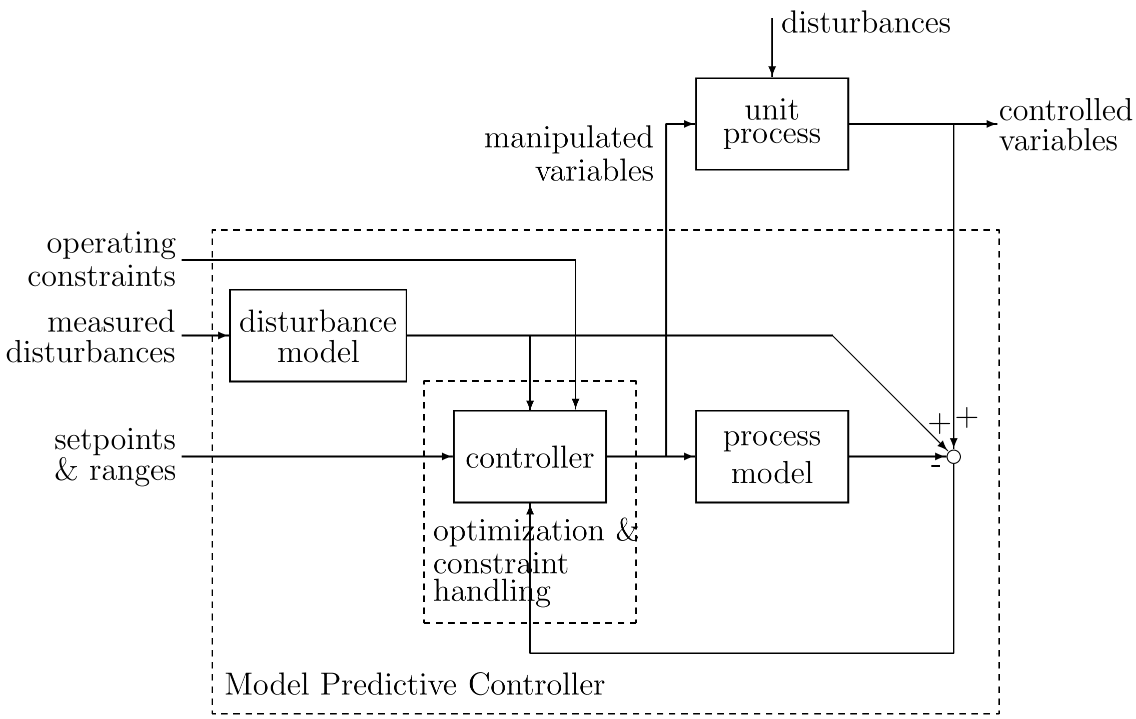

Figure 1.2: Model Predictive Control System

1.2 History of Model Predictive Control

Since 1970 various techniques have been developed for the design of model based control systems for robust multivariable control of industrial unit processes ([15], [24], [27], [33],[34],[35],[36],[37], [51], [74], [83], [99], [108]). Predictive control was pioneered simultaneously by Richalet *et al.* [97], [99] and Cutler & Ramaker [27]. The first implemented algorithms and successful applications were reported in the referenced papers. Model Predictive Control technology has evolved from a basic multivariable process control technology to a technology that enables operation of processes within well defined operating constraints [2], [12]), [94]. The main reasons for increasing acceptance of MPC technology by the process industry since 1985 are clear:

- MPC is a model based controller design procedure, which can easily handle processes with large time-delays, non-minimum phase and unstable processes.
- It is an easy-to-tune method, in principle there are only three basic parameters to be tuned.
- Industrial processes have their limitations in valve capacity, technological requirements and are supposed to deliver output products with some pre- specified quality specifications. MPC can handle these constraints in a systematic way during the design and implementation of the controller.
- Finally MPC can handle structural changes, such as sensor and actuator failures, changes in system parameters and system structure by adapting the control strategy on a sample-by-sample basis.

However, the main reasons for its popularity are the constraint-handling capabilities, the easy extension to multivariable processes and most of all increased profit as discussed in the first section. From academic side the interest in MPC mainly came from the field of self-tuning control. The problem of Minimum Variance control (Åström & Wittenmark [4]) was studied while minimizing the performance index $J(u,k) = \mathrm{E}\{(\,r(k+d) - y(k+d)\,)^2\}$ at time k, where $y(k)$ is the process output signal, $u(k)$ is the control signal, $r(k)$ is the reference signal, $\mathrm{E}(\cdot)$ stands for expectation and d is the process dead-time. To overcome stability problems with non-minimum phase plants, the performance index was modified by adding a penalty on the control signal $u(k)$. Later this $u(k)$ in the performance index was replaced by the increment of the control signal $\Delta u(k) = u(k) - u(k-1)$ to guarantee a zero steady-state error. To handle a wider class of unstable and non-minimum phase systems and systems with poorly known delay the Generalized Predictive Control (GPC) scheme (Clarke *et al.* [13],[14]) was introduced with a quadratic performance index.
In GPC mostly polynomial based models, such as Controlled AutoRegressive Moving Average (CARMA) models or Controlled AutoRegressive Integrated Moving Average (CARIMA) models, are used. These models describe the process using a minimum number of parameters and therefore lead to effective and compact algorithms. Most GPC-literature in this area is based on Single-Input Single-Output (SISO) models. However, the extension to

Multiple-Input Multiple-Output (MIMO) systems is straightforward as was shown by De Vries & Verbruggen [31] using a MIMO polynomial model, and by Kinnaert [62] using a state-space model.
This text covers state-of-the-art technologies for model predictive process control that are good candidates for future generations of industrial model predictive control systems. As all controller design methodologies, MPC also has its drawbacks:

- A detailed process model is required. This means that either one must have a good insight in the physical behavior of the plant or system identification methods have to be applied to obtain a good model.

- The methodology is open, and many variations have led to a large number of MPC-methods. We mention IDCOM (Richalet *et al.* [99]), DMC (Cutler & Ramaker [26]), EPSAC (De Keyser and van Cauwenberghe [29]), MAC (Richalet *et al.* [99]), (Rouhani & Mehra [101]), QDMC (García & Morshedi [49]), GPC (Clarke *et al.* [21][22]), PFC (Richalet *et al.* [99]), IDCOM-M (Grossdidier *et al.* [54]), UPC (Soeterboek [108]), SPC (De Vries & Van den Boom [126]),(Van den Boom & De Vries [118]). UPC and SPC can be seen as a unification of most other methods.

- Although, in practice, stability and robustness are easily obtained by accurate tuning, theoretical analysis of stability and robustness properties are difficult to derive.

Still, in industry for supervisory optimizing control of multivariable unit processes MPC is often preferred over other controller design methods, such as PID, LQ and H_∞. A PID controller is also easily tuned but can only be straightforward applied to SISO systems. LQ and H_∞ can be applied to MIMO systems, but cannot handle signal constraints in an adequate way. These techniques also show difficulties in realizing robust performance for varying operating conditions. Essential in model predictive control is the explicit use of a model that can simulate dynamic behavior of the process at a certain operating point. In this respect model predictive control differs from most of the model based control technologies that have been studied in the Academia in the sixties, seventies and eighties. Academic research has been focusing on the use of models for controller design and robustness analysis of control systems only for quite a while ([1], [10], [15], [17], [32], [33], [34], [35], [36], [41], [61], [65], [74], [134], [135], With their initial work on internal model based control Garcia and Morari [48] made a first step towards bridging academic research in the area of process control and industrial developments in this area. Significant progress has been made in understanding stability and performance of model predictive control systems since the end of the eighties ([63], [69], [78], [96], [95], [104], [141], [139]). A lot of results have been obtained on stability, robustness and performance of model predictive control systems since the start of academic research on model predictive control ([5], [14], [16], [19], [21], [22], [23], [125], [31], [38], [39], [40], [42], [45], [52], [56], [64], [68], [69], [70], [67], [77], [80], [82], [83], [85], [84], [87], [88], [89], [90], [91], [102], [105], [108], [110], [114], [117], [118]. [129],) Since the pioneering work at the end of the seventies and early eighties, MPC has become the most widely applied supervisory control technique in the process industry. In the last decades Model Predictive Control (MPC), also referred to as Model-Based Predictive Control (MBPC), Receding Horizon Control (RHC) and Moving Horizon Control (MHC), has shown to respond effectively to the demands of process industries in many practical process control in especially oil refining and petrochemical process industry. Many papers report successful applications. Some examples of these papers are a distillation column ([68],[98],[99],[115]), a hydrocracker ([25]), a fluidized bed catalytic cracker ([93]), a utility boiler ([79]), a chemical reactor ([48],[66],[90]), a transonic wind tunnel ([109]), a pulp and paper plant ([76]). Applications of MPC to faster systems were also reported, such as a mechatronic servo system ([98]), a servo mechanism ([28]) and a robot arm ([19],[86]). This list is far from complete, but it gives an impression of the wide range of MPC- applications.

1.3 Basics of model predictive control

Model predictive control techniques give flexibility in the operation of unit processes by (automatic) adjustment of the control structure on the basis of given controller objectives, specified operating constraints and actual operating conditions. Model predictive control techniques allow for adjustment of controlled process characteristics in accordance with actual demands. The model predictive control techniques together with in-line model based optimization techniques enable operation of unit processes so that undesired dynamic properties of the process are compensated for and that the behavior observed at the process outputs approximates the desired behavior. Of course the compensation of non-desired dynamic process properties is restricted by the internal mechanisms of the process and by limitations stemming from the controller. Process limitations affect the compensation of non-desired dynamic process properties as this compensation has to be done through appropriate variations of manipulated variables within operating constraints. Controller limitations are related to the prediction capabilities of the applied models for prediction and control inside the model predictive control system. Examples of such limitations are restrictions in the applied models to cover wide dynamic ranges and limited accuracy due to linear approximation of non-linear process behavior. Currently applied model predictive control techniques in industry are still based on the use of linear, often non-parametric type, dynamic models. They apply LP based or QP based optimizers for solving finite horizon, constrained control problems. Application of these controllers for realizing robust, high performance process operation requires an extension of these technologies to the use of more accurate (non-linear) models and control systems. In combination with plant wide optimizers processes tend to be operated over larger operating ranges to continuously drive processes towards most profitable operating conditions. Extension of the model predictive control systems to the non-linear domain is needed in future to enable tight control over these broad ranges and for control of the process during startup, shutdown and transition. MPC is rather a methodology than a single technique. The difference in the various methods is mainly the way the problem is translated into a mathematical formulation, so that the problem becomes solvable in the limited time interval available for calculation of adequate process manipulations in response to external influences on the process behavior (Disturbances). However, in all methods five important items are part of the design procedure:

1. Process model and disturbance model
2. Performance index
3. Constraints
4. Optimization
5. Receding horizon principle

In the following sections these five ingredients of MPC will be discussed.

1.3.1 Process model and disturbance model

The process model is mostly chosen linear. An extension to nonlinear models can be made, but will not be considered in this text. On the basis of the model a prediction of the process signals over a specified horizon is made. In this text we will use the state space description, for it is the most general system description, it is well suited for multivariable systems, while still providing a compact model description. The computations are usually well conditioned and the algorithms easy to implement. The following state space description will be adopted:

$$\boxed{\begin{aligned} x(k+1) &= Ax(k) + B_1 e(k) + B_2 w(k) + B_3 v(k) && (1.3)\\ y(k) &= Cx(k) + D_{11} e(k) + D_{12} w(k) && (1.4) \end{aligned}}$$

where $x(k)$ is the state of the system, $e(k)$ is zero-mean white noise, $w(k)$ denotes of all known signals, for example all known and measurable disturbances, $v(k)$ are the manipulated values or input signals, and $y(k)$ is the output of the system. In chapter 2 we will elaborate on the modeling issues, such as choice of model description (step/impulse response models, polynomial models, state space models), noise and disturbance characterization and model identification. The model will be used to make an estimation of the future behaviour of the system, and as such make predictions of state and output signals.

1.3.2 Performance index

A performance-index or cost-criterion is formulated, reflecting the reference tracking error and the control action. In the *Generalized Predictive Control* (GPC) method (Clarke *et al.*, [21][22]) the performance index is based on control and output signals:

$$J(u,k) = \sum_{j=N_m}^{N} \Big(\hat{y}_p(k+j|k) - r(k+j)\Big)^T \Big(\hat{y}_p(k+j|k) - r(k+j)\Big) +$$

$$+\lambda^2 \sum_{j=1}^{N} \Delta u^T(k+j-1|k)\Delta u(k+j-1|k) \tag{1.5}$$

$y_p(k)$	$= P(q)y(k)$ is the weighted process output signal
$r(k)$	is the reference trajectory
$y(k)$	is the process output signal
$\Delta u(k)$	is the process control increment signal
N_m	is the minimum cost- horizon
N	is the prediction horizon
N_c	is the control horizon
λ	is the weighting on the control signal
$P(q)$	$= 1+p_1 q^{-1}+\ldots+p_{n_p} q^{-n_p}$ is a polynomial with desired closed-loop poles

where $\hat{y}_p(k+j|k)$ is the prediction of $y_p(k+j)$, based on knowledge up to time k, the increment input signal is $\Delta u(k) = u(k) - u(k-1)$ and $\Delta u(k+j) = 0$ for $j \geq N_c$. λ determines the trade-off between tracking accuracy (first term) and control effort (second term). The polynomial $P(q)$ can be chosen by the designer and broaden the class of control objectives. n_p of the closed-loop poles will be placed at the location of the roots of polynomial $P(q)$.

In the *Linear Quadratic Predictive Control* (LQPC) method (García *et al.*, [50]) the performance index is based on control and state signals:

$$J(u,k) = \sum_{j=N_m}^{N} \hat{x}^T(k+j|k)Q\hat{x}(k+j|k) + \sum_{j=1}^{N} u^T(k+j-1|k)Ru(k+j-1|k) \tag{1.6}$$

$x(k)$	is the state signal vector
$u(k)$	is the process control signal
Q	is the state weighting matrix
R	is the control weighting matrix

where $\hat{x}(k+j|k)$ is the prediction of $x(k+j)$, based on knowledge until time k. In some papers the control increment signal $\Delta u(k)$ is used instead of the control signal $u(k)$.

A performance index that is popular in industry is the *Zone performance index*:

$$J(u,k) = \sum_{j=N_m}^{N} \Big(\hat{\epsilon}(k+j|k)\Big)^T \Big(\hat{\epsilon}(k+j|k)\Big) + \lambda^2 \sum_{j=1}^{N} u^T(k+j-1|k)u(k+j-1|k) \tag{1.7}$$

where ϵ_i, $i = 1, \ldots, m$ contribute to the performance index only if $|\hat{y}_i(k) - r_i(k)| > \delta_{max,i}$:

$$\epsilon_i(k) = \begin{cases} 0 & \text{for } \; |\hat{y}_i(k) - r_i(k)| \leq \delta_{max,i} \\ \hat{y}_i(k) - r_i(k) - \delta_{max,i} & \text{for } \; \hat{y}_i(k) - r_i(k) \geq \delta_{max,i} \\ \hat{y}_i(k) - r_i(k) + \delta_{max,i} & \text{for } \; \hat{y}_i(k) - r_i(k) \leq -\delta_{max,i} \end{cases}$$

so

$$|\epsilon_i(k)| = \min_{|\delta_i(k)| \leq \delta_{max,i}} |\hat{y}_i(k) - r_i(k) + \delta_i(k)|$$

The relation between tracking error $\Big(y(k) - r(k)\Big)$, the zone-bounds and zone performance signal $\epsilon(k)$ are visualized in figure 1.3.

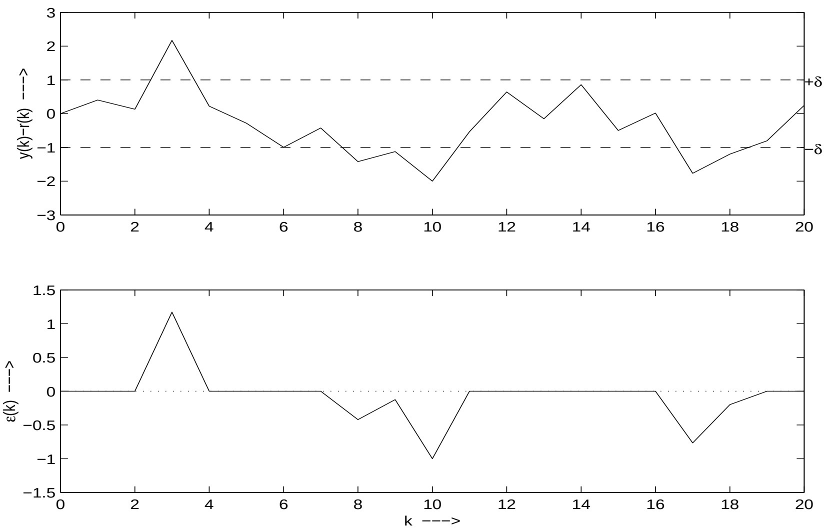

Figure 1.3: Tracking error $\Big(y(k) - r(k)\Big)$ and zone performance signal $\epsilon(k)$

Most papers on predictive control deal with either the GPC or the LQPC performance indices which are clearly weighted squared 2-norms. Also other performance indices can be used, based on other norms, for example the 1-norm or the ∞-norm (Genceli & Nikolaou [52], Zheng & Morari [141]).
In this text only the 2-norm performance index is considered, and a standard form is introduced:

$$\boxed{J(v,k) = \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \qquad (1.8)}$$

$z(k)$	is the performance measure signal vector
$\Gamma(j)$	is a diagonal selection matrix

where $\hat{z}(k+j|k)$ is the prediction of $z(k+j)$ at time k and $\Gamma(j)$ is a diagonal selection matrix with ones and zeros on the diagonal.

The mentioned performance indices (GPC, LQPC and zone performance index) can be translated into the general form of equation (1.8).

GPC:
Choose

$$z(k) = \begin{bmatrix} y_p(k+1) - r(k+1) \\ \lambda \Delta u(k) \end{bmatrix}$$

and

$$\Gamma(j) = \begin{cases} \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} & \text{for} \quad 0 \leq j < N_m - 1 \\ \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} & \text{for} \quad N_m - 1 \leq j \leq N - 1 \end{cases} \tag{1.9}$$

then we find that for $N = N$, performance index (1.5) is equal to performance index (1.8):

$$\begin{aligned} J(v,k) &= \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \\ &= \sum_{j=0}^{N-1} \Big[\, (\hat{y}_p(k+j+1|k) - r(k+j+1))^T \;\; \lambda \Delta u^T(k+j|k) \, \Big] \Gamma(j) \times \\ &\qquad\qquad \begin{bmatrix} \hat{y}_p(k+j+1|k) - r(k+j+1) \\ \lambda \Delta u(k+j|k) \end{bmatrix} \\ &= \sum_{j=N_m-1}^{N-1} \Big(\hat{y}_p(k+j+1|k) - r(k+j+1) \Big)^T \Big(\hat{y}_p(k+j+1|k) - r(k+j+1) \Big) + \\ &\quad + \sum_{j=0}^{N-1} \lambda^2 \Delta u^T(k+j|k) \Delta u(k+j|k) \\ &= \sum_{j=N_m}^{N} \Big(\hat{y}_p(k+j|k) - r(k+j) \Big)^T \Big(\hat{y}_p(k+j|k) - r(k+j) \Big) + \\ &\quad + \sum_{j=1}^{N} \lambda^2 \Delta u^T(k+j-1|k) \Delta u(k+j-1|k) \end{aligned}$$

LQPC:
Choose

$$z(k) = \begin{bmatrix} Q^{1/2} x(k+1) \\ R^{1/2} u(k) \end{bmatrix}$$

and Γ as in (1.9). Now, performance index (1.6) is equal to performance index (1.8) for $N = N$:

$$J(v,k) = \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k)$$

$$
\begin{aligned}
&= \sum_{j=0}^{N-1} \left[\begin{array}{cc} \hat{x}^T(k+j+1|k)Q^{1/2} & u^T(k+j|k)R^{1/2} \end{array} \right] \Gamma(j) \left[\begin{array}{c} Q^{1/2}\hat{x}(k+j+1|k) \\ R^{1/2}u(k+j|k) \end{array} \right] \\
&= \sum_{j=N_m-1}^{N-1} \hat{x}^T(k+j+1|k)Q\hat{x}(k+j+1|k) + \\
&\quad + \sum_{j=0}^{N-1} u^T(k+j|k)Ru(k+j|k) \\
&= \sum_{j=N_m}^{N} \hat{x}^T(k+j|k)Q\hat{x}(k+j|k) + \\
&\quad + \sum_{j=1}^{N} u^T(k+j-1|k)Ru(k+j-1|k)
\end{aligned}
$$

Zone performance index:
Choose

$$
z(k) = \left[\begin{array}{c} r(k+1) - y(k+1) + \delta(k+1) \\ \lambda u(k) \end{array} \right]
$$

and Γ as in (1.9). Further we introduce an additional constraint that $|\delta(k+j)| \leq \delta_{max}$, then we find that for $N = N$, performance index (1.7) is equal to performance index (1.8):

$$
\begin{aligned}
J(v,k) &= \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \\
&= \min_{|\delta|\leq\delta_{max}} \sum_{j=0}^{N-1} \left[\begin{array}{cc} (\hat{y}(k+j+1|k)-r(k+j+1)+\delta(k+j+1))^T & \lambda u^T(k+j|k) \end{array} \right] \\
&\qquad \times \Gamma(j) \left[\begin{array}{c} \hat{y}(k+j+1|k) - r(k+j+1) + \delta(k+j+1) \\ \lambda u(k+j|k) \end{array} \right] \\
&= \min_{|\delta|\leq\delta_{max}} \sum_{j=N_m-1}^{N-1} \Big(\hat{y}(k+j+1|k) - r(k+j+1) + \delta(k+j+1)\Big)^T \times \\
&\qquad \Big(\hat{y}(k+j+1|k) - r(k+j+1) + \delta(k+j+1)\Big) + \\
&\quad + \sum_{j=0}^{N-1} \lambda^2 u^T(k+j|k)u(k+j|k) \\
&= \min_{|\delta|\leq\delta_{max}} \sum_{j=N_m}^{N} \Big(\hat{y}(k+j|k) - r(k+j)\Big)^T \Big(\hat{y}(k+j|k) - r(k+j)\Big) + \\
&\quad + \sum_{j=1}^{N} \lambda^2 u^T(k+j-1|k)u(k+j-1|k) \\
&= \min_{|\delta|\leq\delta_{max}} \sum_{j=N_m}^{N} \hat{\epsilon}^T(k+j|k)\hat{\epsilon}(k+j|k) + \sum_{j=1}^{N} \lambda^2 u^T(k+j-1|k)u(k+j-1|k)
\end{aligned}
$$

In chapter 4 the transformation of the GPC, LQPC and zone performance index will be discussed in more detail.

1.3.3 Constraints

In practice industrial processes are subject to constraints. Specific signals must not violate specified bounds due to safety limitations, environmental regulations, consumer specifications and physical restrictions such as minimum and/or maximum temperature, pressure, level limits in reactor tanks, flows in pipes and slew rates of valves.
Careful tuning of the controller parameters may keep these values away from the bounds. However, because of economical motives, the control system should drive the process towards the constraints as close as possible, without violating them: Closer to limits in general often means closer to maximum profit.
Therefore, predictive control employs a more direct approach by modifying the optimal unconstrained solution in such a way that constraints are not violated. This can be done using optimization techniques such as linear programming (LP) or quadratic programming (QP) techniques.
In most cases the constraints can be translated in bounds on control, state or output signals:

$$\begin{array}{llll} u_{min} \leq u(k) \leq u_{max} \ , & \forall\, k & \Delta u_{min} \leq \Delta u(k) \leq \Delta u_{max} \ , & \forall\, k \\ y_{min} \leq y(k) \leq y_{max} \ , & \forall\, k & x_{min} \leq x(k) \leq x_{max} \ , & \forall\, k \end{array}$$

Under no circumstances the constraints may be violated, and the control action has to be chosen such that the constraints are satisfied.

Except from inequality constraints we can also use equality constraints. Equality constraints are usually motivated by the control algorithm itself. An example is the control horizon, which forces the control signal to become constant:

$$\Delta u(k+j|k) = 0 \ \text{ for } \ j \geq N_c$$

This makes the control signal to be smooth and the controller more robust. A second example is the state end-point constraint

$$\hat{x}(k+N|k) = x_{ss}$$

which is related to stability and forces the state at the end of the prediction horizon to its steady state value x_{ss}.

Resuming, we see that there are two types of constraints (inequality and equality constraints). In this course we look at the two types in a generalized framework:

Inequality constraints:

$$\tilde{\psi}(k) \leq \tilde{\Psi}(k) \tag{1.10}$$

Equality constraints:

$$\tilde{\phi}(k) = 0 \tag{1.11}$$

where $\tilde{\Psi}(k)$ does not depend on future inputs of the system. In the above formulas $\tilde{\phi}(k)$ and $\tilde{\psi}(k)$ are vectors and the equalities and inequalities are meant element-wise. If there are multiple constraints, for example $\tilde{\phi}_1(k) = 0$ and $\tilde{\phi}_2(k) = 0$, $\tilde{\psi}_1(k) \leq \tilde{\Psi}_1(k), \ldots, \tilde{\psi}_4(k) \leq \tilde{\Psi}_4(k)$ we can combine them by both stacking them into one equality constraint and one inequality constraint.

$$\tilde{\phi}(k) = \begin{bmatrix} \tilde{\phi}_1(k) \\ \tilde{\phi}_2(k) \end{bmatrix} = 0 \qquad \tilde{\psi}(k) = \begin{bmatrix} \tilde{\psi}_1(k) \\ \vdots \\ \tilde{\psi}_4(k) \end{bmatrix} \leq \begin{bmatrix} \tilde{\Psi}_1(k) \\ \vdots \\ \tilde{\Psi}_4(k) \end{bmatrix} = \tilde{\Psi}(k)$$

Note that if a variable is bounded two-sided

$$\tilde{\Psi}_{1,min} \leq \tilde{\psi}_1(k) \leq \tilde{\Psi}_{1,max}$$

we can always translate that into two one-sided constraints by setting:

$$\tilde{\psi}(k) = \begin{bmatrix} \tilde{\psi}_1(k) \\ -\tilde{\psi}_1(k) \end{bmatrix} \leq \begin{bmatrix} \tilde{\Psi}_{1,max} \\ -\tilde{\Psi}_{1,min} \end{bmatrix} = \tilde{\Psi}(k)$$

which corresponds to the general form of eqaution (1.10).
The capability of predictive control to cope with signal constraints is probably the main reason for its popularity. It allows operation of the process within well defined operating limits. As constraints are respected, better exploitation of the permitted operating range is feasible. In many applications of control, signal constraints are present, caused by limited capacity of liquid buffers, valves, saturation of actuators and the more. By minimizing performance index (1.8) subject to these constraints we obtain the best possible control signal within the set of admissible control signals.

Structuring the input signal

To obtain a tractable optimization problem, the input signal should be structured. The degrees of freedom in the future input signal $[u(k|k), u(k+1|k), \ldots, u(k+N-1|k)]$ is decreased. This can be done in various ways. The most common and most frequently used method is by using a control horizon, introduce input blocking or parametrize the input with (orthogonal) basis functions.

Control horizon:
When we apply a control horizon, the input signal is assumed to become constant from a certain moment in the future, denoted as control horizon N_c, so

$$u(k+j|k) = u(k+N_c-1|k) \quad \text{for} \quad j \geq N_c$$

Blocking:
Instead of making the input to be constant beyond the control horizon, we can force the input to remain constant during some predefined (non-uniform) intervals. In that way, there is some freedom left beyond the control horizon. Define n_{bl} intervals with range $(m_\ell, m_{\ell+1})$. The first m_1 control variables $(u(k), \dots, u(k+m_1-1))$ are still free. The input beyond m_1 can be described by:

$$u(k+j|k) = u(k+m_\ell-1|k) \quad \text{for} \quad m_\ell \leq j \leq m_{\ell+1},\;\; \ell = 1, \dots, n_{bl}$$

Basis functions:
Parametrization of the input signal can also be done using a set of basis functions:

$$u(k+j|k) = \sum_{i=0}^{M} S_i(j)\, \alpha_i(k)$$

where $S_i(j)$ are the basis functions, M are the degrees of freedom left, and the scalars $\alpha_i(k)$, $i = 1, \dots, M$ are the parameters, to be optimized at time k.
The already mentioned structuring of the input signal, with either a control horizon or blocking, can be seen as a parametrization of the input signal with a set of basis functions. For example, by introducing a control horizon, we choose $M = N_c - 1$ and the basis functions become:

$$\begin{aligned} S_i(j) &= \delta(j-i) \;\; \text{for} \quad i = 0, \dots, N_c - 2 \\ S_{N_c-1}(j) &= E(j - N_c + 1) \end{aligned}$$

where $\delta(k-i)$ is the discrete-time impulse function, being 1 for $j = i$ and zero elsewhere, and $E(j - N_c + 1)$ is the step function at time $j = N_c - 1$. The optimization parameters are now equal to

$$\alpha_i(k) = u(k+i|k) \quad \text{for } i = 0, \dots, N_c - 1$$

which indeed corresponds to the N_c free parameters.
In the case of blocking, we choose $M = m_1 + n_{bl}$ and the basis functions become:

$$\begin{aligned} S_i(j) &= \delta(j-i) \;\; \text{for} \quad i = 0, \dots, m_1 - 1 \\ S_{m_1+\ell}(j) &= E(j - m_\ell) - E(j - m_{\ell+1}) \;\; \text{for} \quad \ell = 1, \dots, n_{bl} \end{aligned}$$

The optimization parameters are now equal to

$$\alpha_i(k) = u(k+i|k) \quad \text{for } i = 0, \ldots, m_1 - 1$$

and

$$\alpha_{m_1+\ell}(k) = u(k+m_\ell|k) \quad \text{for } \ell = 1, \ldots, n_{bl}$$

which corresponds to the $N_c - 1 + n_{bl}$ free parameters.

Figure 1.4 gives the basis functions in case of a control horizon approach and a blocking

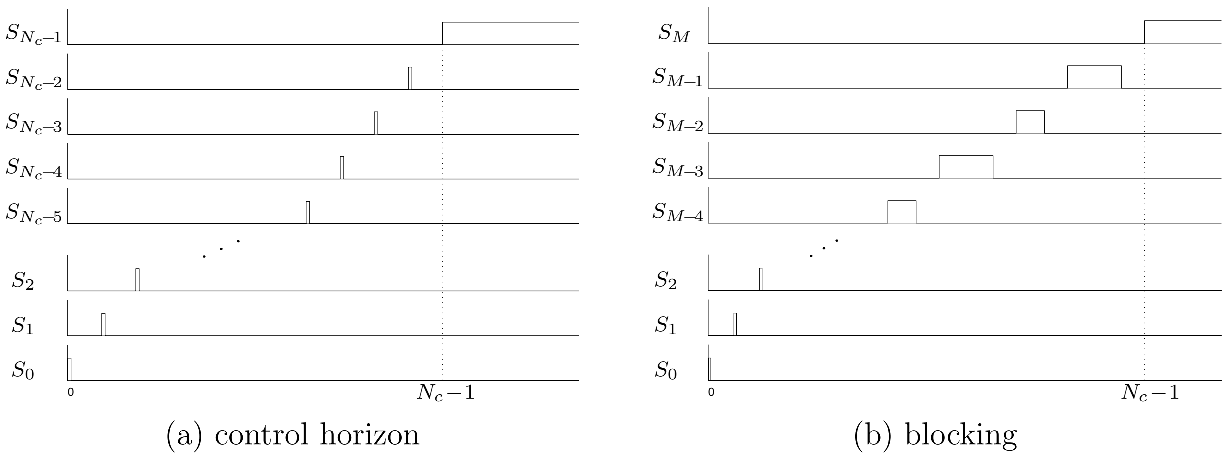

Figure 1.4: Basis functions in case of a control horizon approach (a) and a blocking approach (b).

approach.

To obtain a tractable optimization problem, we can choose a set of basis functions $S_i(k)$ that are orthogonal. These orthogonal basis functions will be discussed in section 4.3.

1.3.4 Optimization

An optimization algorithm will be applied to compute a sequence of future control signals that minimizes the performance index subject to the given constraints. For linear models with linear constraints and a quadratic (2-norm) performance index the solution can be found using quadratic programming algorithms. If a 1-norm or ∞- norm performance index is used, linear programming algorithms will provide the solution (Genceli & Nikolaou [52], Zheng & Morari [141]). Both types of algorithms are convex and show fast convergence.
In some cases, the optimization problem will have an empty solution set, so the problem is not feasible. In that case we will have to relax one or more of the constraints to find a solution leading to an acceptable control signal. A prioritization among the constraints is an elegant way to tackle this problem.
Procedures for different types of MPC problems will be presented in chapter 5.

1.3.5 Receding horizon principle

Predictive control uses the receding horizon principle. This means that after computation of the optimal control sequence, only the first control sample will be implemented, subsequently the horizon is shifted one sample and the optimization is restarted with new information of the measurements. Figure 1.5 explains the idea of receding horizon. At

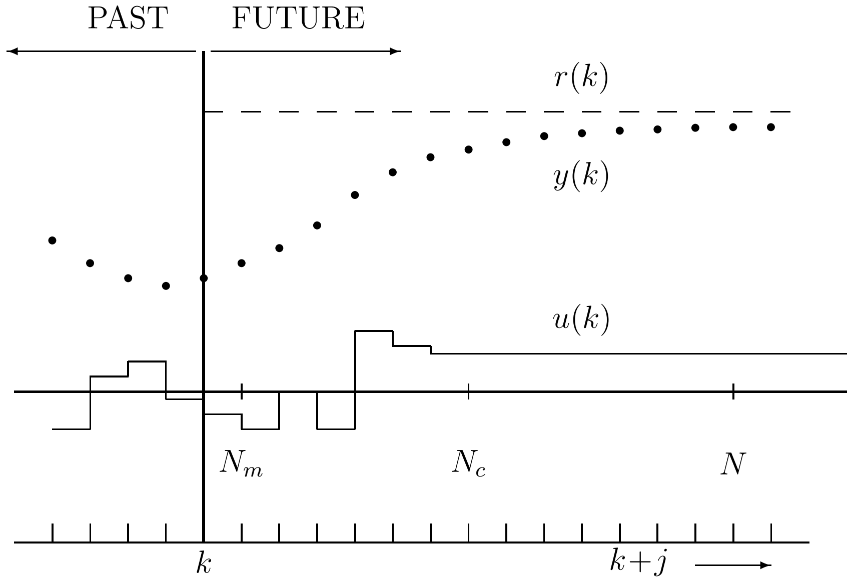

Figure 1.5: The 'Moving horizon' in predictive control

time k the future control sequence $\{u(k|k),\ \ldots,\ u(k+N_c-1|k)\}$ is optimized such that the the performance-index $J(u,k)$ is minimized subject to constraints.
At time k the first element of the optimal sequence $(u(k)=u(k|k))$ is applied to the real process. At the next time instant the horizon is shifted and a new optimization at time $k+1$ is solved.
The prediction $\hat{y}_p(k+j|k)$ in (1.5) and $\hat{x}(k+j|k)$ in (1.6) are based on dynamical models. Common model representations in MPC are polynomial models, step response models impulse response models or state space models. In this course we will consider a state space representation of the model. In chapter 2 we will show how other realizations can be transformed in a state space model.

1.4 Example: MPC with an impulse response model

At the end of this introductory chapter we will present a simple example to illustrate the use of the different ingredients.

Model:

Consider a discrete-time linear time invariant system with input signal $u(k)$, output signal $y(k)$, $k \in \mathbb{Z}$, and impulse response $h(k)$ (see figure 1.6), given by:

$$h(k) = \begin{cases} 2^{-k} & \text{for } 1 \leq k \leq 10, \quad k \in \mathbb{Z} \\ 0 & \text{otherwise} \end{cases}$$

and initial conditions $u(k) = 0$ and $y(k) = 0$ for $k < 0$.
On the basis of the model we can predict the output signal $y(k+j)$ at time k for a given

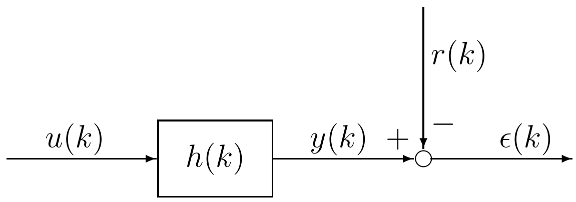

Figure 1.6: Control problem

input signal u:

$$y(k+j|k) = \sum_{m=1}^{10} h(m)\, u(k+j-m|k) \qquad \text{for } j \geq 0\ , \quad k \in \mathbb{Z}$$

Performance index:

We like the output $y(k)$ to follow a given trajectory $r(k) = 1$ for $k > 0$, so the error $\epsilon(k) = y(k) - r(k)$ should be as small as possible. Therefore the performance index at time k is chosen as the sum of the squared errors:

$$\min_u J(u,k) = \min_u \sum_{j=1}^{N} \epsilon^2(k+j|k)$$

The parameter N is the denoted as the prediction horizon and determines how far in the future the error is taken into account. Using the expression for the prediction we derive:

$$\epsilon(k+j|k) = \sum_{m=1}^{10} h(m)\, u(k+j-m|k) - r(k+j)$$

and the performance index becomes:

$$J(u,k) = \sum_{j=1}^{N} \Big(\sum_{m=1}^{10} h(m)\, u(k+j-m|k) - r(k+j) \Big)^2$$

Constraints:

In this example we will introduce two constraints. The first is called the control horizon constraint:

$$u(k+j|k) = u(k+N_c-1|k) \quad \text{for} \quad j \geq N_c$$

This implies that $u(k+j|k)$ is kept constant for $j \geq N_c$. This forces the system to go to its steady-state value, which has a great stabilizing effect. The second constraint in this example is that we want the input to remain bounded in the interval

$$|\, u(k+j|k) \,| \leq u_{max} \quad \text{for} \quad j \geq 0$$

where the bound is chosen $u_{max} = 1.2$.

Optimization:

Combining the above performance index $J(k)$ and the constraints we obtain a minimization problem at time k:

$$\min_u J(u,k) = \min_{u(k|k),\ldots,u(k+N_c-1|k)} \sum_{j=1}^{N} \Big(\sum_{m=1}^{10} h(m)\, u(k+j-m|k) - r(k+j) \Big)^2$$

subject to

$$u(k+j|k) = u(k+N_c-1|k) \quad \text{for} \quad j \geq N_c$$

$$|\, u(k+j|k) \,| \leq 1.2 \quad \text{for} \quad j \geq 0$$

The above optimization problem is a quadratic programming problem and fast converging algorithms exist for this kind of problems. In chapter 3, chapter 5 and appendix A we will come back on this topic.

Receding horizon principle:

Predictive control is a receding horizon method. This means that at time instant k the above performance index $J(k)$ for the interval $(k+1, k+N)$ is minimized subject to the constraints and the first element $u(k) = u(k|k)$ of the computed optimal input sequence $\{u(k|k), u(k+1|k), \ldots, u(k+N_c-1|k)\}$ is applied to the system. Then the horizon is shifted one sample and the optimization is restarted for the interval $(k+2, k+N+1)$.

Simulation results:

The system with controller is simulated for $k = 4, \ldots, 16$ where we have chosen the prediction horizon $N = 15$ and the control horizon $N_c = 5$. Figure 1.8 shows the input signal $u(k)$ and figure 1.7 shows the output signal $y(k)$. Clearly the input signal remains bounded within its range $-1.2 \leq u(k) \leq 1.2$.

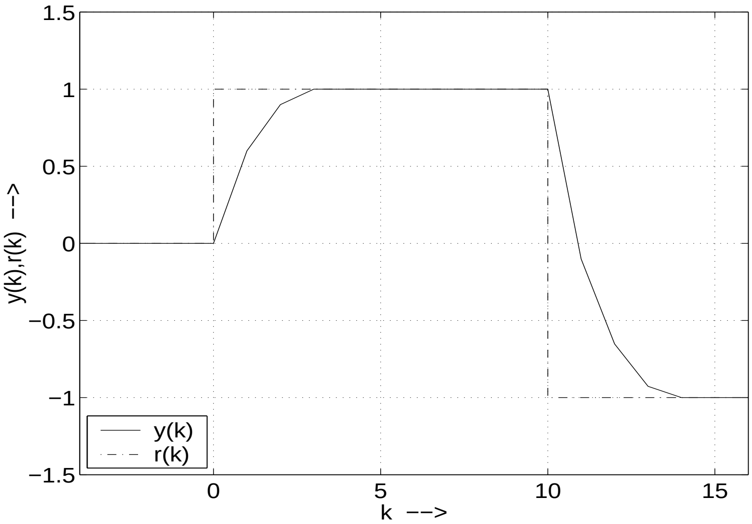

Figure 1.7: Reference signal $r(k)$ and output signal $y(k)$

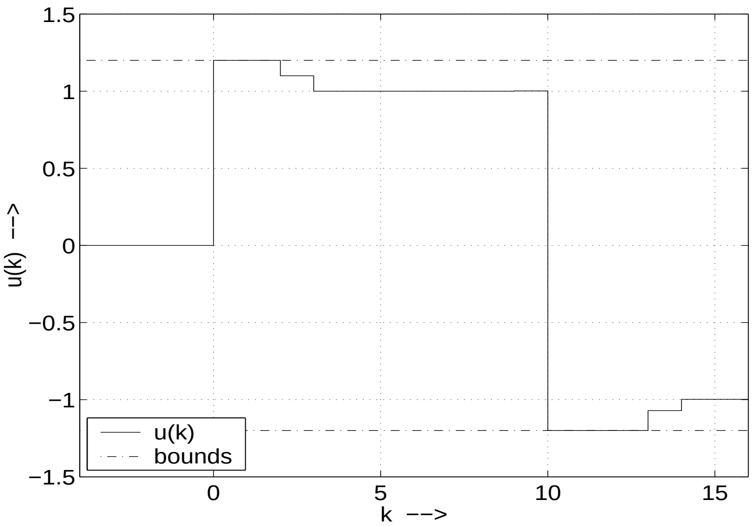

Figure 1.8: Input signal $u(k)$

Chapter 2

The model

2.1 Introduction

An important difference between Model Predictive control (MPC) and PID-kind design-methods is the explicit use of a model. This aspect is both the advantage and the disadvantage of MPC. The advantage is that the behaviour of our controller can be studied in detail, simulations can be made and possible failures in plant or controller can be well-detected. The disadvantage is that a detailed study of the plant behaviour has to be done before the actual MPC-design can be started. About 80 % of the work that has to be done, is in modelling and identification of the plant (Richalet [98]). However, in the final result this effort and the investment to obtain a good model nearly always pay back in a short time.

The models applied in MPC serve two purposes:

- Prediction of expected future process output behavior on the basis of inputs and known disturbances applied to the process in the past
- Calculation of the next process input signal that minimizes the controller objective function

The models required for these tasks do not necessarily have to be the same. The model applied for prediction may differ from the model applied for calculation of the next control action. In practice though both models are almost always chosen to be the same.
As the models play such an important role in model predictive control the models are discussed in this chapter. The models applied are so called Input-Output (IO) models. These models describe the input-output behavior of the process. The MPC controller discussed in the sequel explicitly assumes that the superposition theorem holds. The models applied in the controller are chosen to be linear models therefore. Two types of IO models are applied:

- Direct Input-Output models (IO models) in which the input signal is directly applied to the models.

- Increment Input-Output models (IIO models) in which the increments of the input signal are applied to the models instead of the input directly.

The following assumptions are made with respect to the models that are applied:

1. Linear
2. Time invariant
3. Discrete time
4. Causal
5. Finite order

Linearity is assumed to allow the use of the superposition theorem. The second assumption -time invariance- enables the use of a model made on the basis of observations of process behavior at a certain time for simulation of process behavior at another arbitrary time instant. The discrete time representation of process dynamics supports the sampling mechanisms required for calculating control actions with an MPC algorithm implemented in a sequential operating system like a computer. Causality implies that the model does not anticipate future changes of its inputs. This aligns quite well with the behavior of physical, chemical, biological systems. The assumption regarding the order of the model to be finite implies that models and model predictions can always be described in a set of explicit equations. These assumptions have to be validated of course against the actual process behavior as part of the process modeling or process identification phase.
In this chapter only discrete time models will be considered. The control algorithm will always be implemented on a digital computer. The design of a discrete time controller is obvious. Further a linear time-invariant continuous time model can always be transformed into a linear time-invariant discrete time model using a zero-order hold z-tranformation.

The most general description of linear time invariant systems is the state space description. Models, given in a impulse/step response or transfer function structure, can easily be converted into state space models. Computations can be done numerically more reliable if based on (balanced) state space models instead of impulse/step response or transfer function models, especially for multivariable systems (Kailath [60]). Also system identification, using the input-output data of a system, can be done using reliable and fast algorithms (Verhaegen & Dewilde [124]; Backx *et al.* [6],[9],[8],[7]; Van Overschee [123],[122]; Falkus [44]).

Input-output (IO) models

In this course we consider causal, discrete time, linear, finite-dimensional, time-invariant systems given by

$$y(k) = G_o(q)\, u(k) + F_o(q) d_o(k) + H_o(q) e_o(k) \tag{2.1}$$

in which $G_o(q)$ is the process model, $F_o(q)$ is the disturbance model, $H_o(q)$ is the noise model, $y(k)$ is the output signal, $u(k)$ is the input signal, $d_o(k)$ is a known disturbance signal, $e_o(k)$ is zero-mean white noise (ZMWN) and q is the shift-operator $q^{-1}y(k) = y(k-1)$. We assume $G_o(q)$ to be stricly proper, which means that $y(k)$ does not depend on the present value of $u(k)$, but only on past values $u(k-j)$, $j>0$.

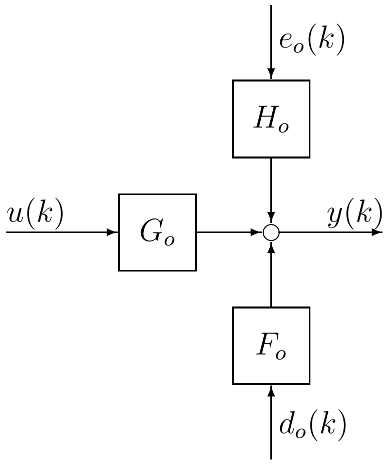

Figure 2.1: Input-Output (IO) model

A state space representation for this system can be given as

$$\begin{aligned} x_o(k+1) &= A_o x_o(k) + K_o e_o(k) + L_o d_o(k) + B_o u_o(k) && (2.2)\\ y(k) &= C_o x_o(k) + D_H e_o(k) + D_F d_o(k) && (2.3) \end{aligned}$$

and so the transfer functions $G_o(q)$, $F_o(q)$ and $H_o(q)$ can be given as

$$\begin{aligned} G_o(q) &= C_o\,(\,qI - A_o\,)^{-1}B_o && (2.4)\\ F_o(q) &= C_o\,(\,qI - A_o\,)^{-1}L_o + D_F && (2.5)\\ H_o(q) &= C_o\,(\,qI - A_o\,)^{-1}K_o + D_H && (2.6) \end{aligned}$$

Increment-input-output (IIO) models

Sometimes it is useful not to work with the input signal $u(k)$ itself, but with the input increment instead, defined as

$$\Delta u(k) = u(k) - u(k-1) = (1-q^{-1})u(k) = \Delta(q)u(k)$$

where $\Delta(q) = (1-q^{-1})$ is denoted as the increment operator. Using the increment of the input signal implies that the model keeps track of the actual value of the input signal. The model needs to integrate the increments of the inputs to calculate the output corresponding

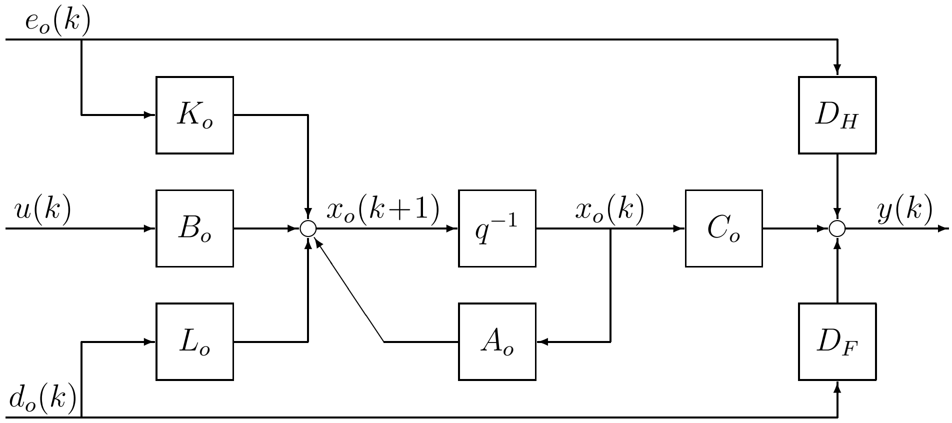

Figure 2.2: State space representation of the model

with the input signals actually applied to the process. We obtain an increment-input-output (IIO) model:

$$\text{IIO-model:} \quad y(k) = G_i(q)\,\Delta u(k) + F_i(q)\,d_i(k) + H_i(q)\,e_i(k) \tag{2.7}$$

where $d_i(k)$ is a known disturbance signal and $e_i(k)$ is ZMWN. A state space representation for this system is given by

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + K_i e_i(k) + L_i d_i(k) + B_i \Delta u(k) && (2.8)\\ y(k) &= C_i x_i(k) + D_H e_i(k) + D_F d_i(k) && (2.9) \end{aligned}$$

The transfer functions $G_i(q)$ and $H_i(q)$ become

$$\begin{aligned} G_i(q) &= C_i\,(\,q\,I - A_i\,)^{-1} B_i && (2.10)\\ F_i(q) &= C_i\,(\,q\,I - A_i\,)^{-1} L_i + D_F && (2.11)\\ H_i(q) &= C_i\,(\,q\,I - A_i\,)^{-1} K_i + D_H && (2.12) \end{aligned}$$

Relation between IO and IIO model

Given an IO-model with state space realization

$$\begin{aligned} x_o(k+1) &= A_o\,x_o(k) + K_o\,e_o(k) + L_o\,d_o(k) + B_o\,u(k)\\ y(k) &= C_o\,x_o(k) + D_H\,e_o(k) + D_F\,d_o(k) \end{aligned}$$

Define the system matrices

$$A_i = \begin{bmatrix} I & C_o \\ 0 & A_o \end{bmatrix} \quad B_i = \begin{bmatrix} 0 \\ B_o \end{bmatrix} \quad K_i = \begin{bmatrix} D_H \\ K_o \end{bmatrix} \quad C_i = \begin{bmatrix} I & C_o \end{bmatrix} \quad L_i = \begin{bmatrix} D_F \\ L_o \end{bmatrix}$$

the disturbance and noise signals

$$\begin{aligned} d_i(k) &= \Delta d_o(k) &= d_o(k) - d_o(k-1) \\ e_i(k) &= \Delta e_o(k) &= e_o(k) - e_o(k-1) \end{aligned}$$

and new state

$$x_i(k) = \begin{bmatrix} y(k-1) \\ \Delta x_o(k) \end{bmatrix}$$

where $\Delta x_o(k) = x_o(k) - x_o(k-1)$ is the increment of the original state. Then the state space realization, given by

$$\begin{aligned} x_i(k+1) &= A_i\, x_i(k) + K_i\, e_i(k) + L_i\, d_i(k) + B_i\, \Delta u(k) \\ y(k) &= C_i\, x_i(k) + D_H\, e_i(k) + D_F\, d_i(k) \end{aligned}$$

is a IIO-model of the original IO model.

An interesting observation in comparing the IO model with the corresponding IIO model is the increase of the number of states with the number of outputs of the system. As can been seen from the state matrix A_i of the IIO model this increase in the number of states compared to the state matrix A_o of the IO model is related to the integrators required for calculating the actual process outputs on the basis of input increments. The additional eigenvalues of A_i are all integrators: eigenvalues $\lambda_i = 1$.

Proof:
From (2.2) we derive

$$\begin{aligned} \Delta x_o(k+1) &= A_o\, \Delta x_o(k) + K_o\, \Delta e_o(k) + L_o\, \Delta d_o(k) + B_o\, \Delta u(k) \\ \Delta y(k) &= C_o\, \Delta x_o(k) + D_H\, \Delta e_o(k) + D_F\, \Delta d_o(k) \end{aligned}$$

and so for the output signal we derive

$$\begin{aligned} y(k) &= y(k-1) + \Delta y(k) = \\ &= y(k-1) + C_o\, \Delta x_o(k) + D_H\, \Delta e_o(k) + D_F\, \Delta d_o(k) \end{aligned}$$

so based on the above equations we get:

$$\begin{aligned} \begin{bmatrix} y(k) \\ \Delta x_o(k+1) \end{bmatrix} &= \begin{bmatrix} I & C_o \\ 0 & A_o \end{bmatrix} \begin{bmatrix} y(k-1) \\ \Delta x_o(k) \end{bmatrix} + \begin{bmatrix} D_H \\ K_o \end{bmatrix} \Delta e_o(k) + \\ &\quad + \begin{bmatrix} D_F \\ L_o \end{bmatrix} \Delta d_o(k) + \begin{bmatrix} 0 \\ B_o \end{bmatrix} \Delta u(k) \\ y(k) &= \begin{bmatrix} I & C_o \end{bmatrix} \begin{bmatrix} y(k-1) \\ \Delta x_o(k) \end{bmatrix} + D_H \Delta e_o(k) + D_F \Delta d_o(k) \end{aligned}$$

by substitution of the variables $x_i(k)$, $d_i(k)$ and $e_i(k)$ and the matrices A_i, B_i, L_i, K_i and C_i the state space IIO model is found. □ End Proof

An important observation is that in the IO-model $e_o(k)$ is ZMWN, where as in the IIO-model the disturbance $e_i(k) = \Delta e_o(k)$ is assumed to be ZMWN, and thus in the IIO-model $e_o(k)$ becomes integrated ZMWN.

The transfer functions of the IO-model and IIO-model are related by

$$\begin{aligned} G_i(q) &= G_o(q)\Delta(q)^{-1} \\ F_i(q) &= F_o(q)\Delta(q)^{-1} \\ H_i(q) &= H_o(q)\Delta(q)^{-1} \end{aligned} \tag{2.13}$$

where $\Delta(q) = (1 - q^{-1})$. In the next sections we will show how the other models, like impulse response models, step response models and polynomial models relate to state space models.

The advantage of using an IIO model

The main reason for using an IIO model is the good steady-state behaviour of the controller designed on the basis of this IIO model. As indicated above the IIO model simulates process outputs on the basis of input increments. This implies that the model must have integrating action to describe the behavior of processes that have steady state gains unequal to zero. The integrating behavior of the model also implies that the model output can take any value with inputs being equal to zero. This property of the IIO model appears to be very attractive for good steady state behavior of the control system as we will see.

Steady-state behaviour of an IO model: Consider a IO model $G_o(q)$ without a pole in $q = 1$, so $\|G_o(1)\| < \infty$. Further assume $e_o(k) = 0$ and $d_o(k) = 0$. If $u \longrightarrow u_{ss}$ for $k \longrightarrow \infty$ then $y \longrightarrow y_{ss} = G_o(1)u_{ss}$ for $k \longrightarrow \infty$. This process is controlled by a predictive controller, minimizing the IO performance index

$$J(k) = \sum_{j=1}^{N_2} \|y(k+j) - r(k+j)\|^2 + \lambda^2 \|u(k+j-1)\|^2$$

Let the reference be constant $r(k) = r_{ss} \neq 0$ for large k, and let $k \longrightarrow \infty$. Then u and y will reach their steady-state and so $J(k) = N_2 J_{ss}$ for $k \longrightarrow \infty$, where

$$\begin{aligned} J_{ss} &= \|y_{ss} - r_{ss}\|^2 + \lambda^2 \|u_{ss}\|^2 \\ &= \|G_o(1)u_{ss} - r_{ss}\|^2 + \lambda^2 \|u_{ss}\|^2 \\ &= u_{ss}^T \Big(G_o^T(1) G_o(1) + \lambda^2 I \Big) u_{ss} - 2u_{ss}^T G_o^T(1) r_{ss} + r_{ss}^T r_{ss} \end{aligned}$$

Minimizing J_{ss} over u_{ss} means that

$$\partial J_{ss} / \partial u_{ss} = 2\Big(G_o^T(1) G_o(1) + \lambda^2 I \Big) u_{ss} - 2G_o^T(1) r_{ss} \; = \; 0$$

so

$$u_{ss} = \Big(G_o^T(1) G_o(1) + \lambda^2 I \Big)^{-1} G_o^T(1) r_{ss}$$

The steady-state output becomes

$$y_{ss} = G_o(1) \Big(G_o^T(1) G_o(1) + \lambda^2 I \Big)^{-1} G_o^T(1) r_{ss}$$

It is clear that $y_{ss} \neq r_{ss}$ for $\lambda > 0$ and for this IO model there will always be a steady-state error for $r_{ss} \neq 0$ and $\lambda > 0$.

Steady-state behaviour of an IIO model: Consider the above model in IIO form, so $G_i(q) = \Delta^{-1} G_o(q)$, and let it be controlled by a predictive controller, minimizing the IIO performance index

$$J(k) = \sum_{j=1}^{N_2} \|y(k+j) - r(k+j)\|^2 + \lambda^2 \|\Delta u(k+j-1)\|^2$$

In steady state the output is given by $y_{ss} = G_o(1) u_{ss}$ and the increment input becomes $\Delta u_{ss} = 0$, because the input signal $u = u_{ss}$ has become constant. In steady-state we will reach the situation that the performance index becomes $J(k) = N_2 J_{ss}$ for $k \longrightarrow \infty$, where

$$J_{ss} = \|y_{ss} - r_{ss}\|^2 + \lambda^2 \|\Delta u_{ss}\|^2$$

The optimum $J_{ss} = 0$ is obtained for $y_{ss} = r_{ss}$ and $\Delta u_{ss} = 0$, which means that no steady-state error occures. It is clear that for a good steady-state behaviour, it is necessary to design the predictive controller on the basis of an IIO model.

2.2 Impulse and step response models

MPC has part of its roots in the process industry, where the use of detailed dynamical models is less common. To get reliable dynamical models of these processes on the basis of physical laws is difficult and it is therefore not suprising that the first models used in MPC were impulse and step response models (Richalet [99]), (Cutler & Ramaker [26]). The models are easily obtained by rather simple experiments and give good models for sufficiently large length of the impulse/step response.

Let g_m and s_m denote the impulse response parameters and the step response parameters of the stable system $G_o(q)$, respectively. Then

$$s_m = \sum_{j=1}^{m} g_j \quad \text{and} \quad g_m = s_m - s_{m-1} \quad , m \in \{1, 2, \ldots\}$$

The transfer function $G_o(q)$ is found by

$$G_o(q) \;=\; \sum_{m=1}^{\infty} g_m \, q^{-m} \;=\; \sum_{m=1}^{\infty} s_m \, q^{-m} (1 - q^{-1}) \tag{2.14}$$

In the same way, let f_m and t_m denote the impulse response parameters and the step response parameters of the stable disturbance model $F_o(q)$, respectively. Then

$$t_m = \sum_{j=0}^{m} f_j \quad \text{and} \quad f_m = t_m - t_{m-1} \quad , m \in \{0, 1, 2, \ldots\}$$

The transfer function $F_o(q)$ is found by

$$F_o(q) \;=\; \sum_{m=0}^{\infty} f_m\, q^{-m} \;=\; \sum_{m=0}^{\infty} t_m\, q^{-m}(1 - q^{-1}) \tag{2.15}$$

The truncated impulse response model is defined by

$$y(k) = \sum_{m=1}^{n} g_m\, u(k-m) + \sum_{m=0}^{n} f_m\, d_o(k-m) + e_o(k)$$

where n is an integer number such that $g_m \approx 0$ and $f_m \approx 0$ for all $m \geq n$.
This is an IO-model in which the disturbance $d_o(k)$ is known and noise signal $e_o(k)$ is chosen as ZMWN.

The truncated step response model is defined by

$$y(k) = \sum_{m=1}^{n-1} s_m\, \Delta u(k-m) + s_n\, u(k-n) + \sum_{m=0}^{n-1} t_m\, d_i(k-m) + t_n\, d_o(k-n) + e_o(k)$$

This is a IIO-model in which $d_i(k) = \Delta d_o(k) = d_o(k) - d_o(k-1)$ is the disturbance increment signal and the noise increment signal $e_i(k) = \Delta e_o(k) = e_o(k) - e_o(k-1)$ is chosen as ZMWN to account for an offset at the process output, so

$$e_o(k) = \Delta^{-1}(q) e_i(k)$$

From impulse response model to state space model

(Lee *et al.* [69]) Let

$$\begin{aligned} G_o(q) &= g_1 q^{-1} + g_2 q^{-2} + \ldots + g_n q^{-n} \\ F_o(q) &= f_0 + f_1 q^{-1} + f_2 q^{-2} + \ldots + f_n q^{-n} \\ H_o(q) &= 1 \end{aligned}$$

where g_i are the impulse response parameters of the system, then the state space matrices are given by:

$$A_o = \begin{bmatrix} 0 & I & 0 & \ldots & 0 \\ 0 & 0 & I & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & I \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix} \quad K_o = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad L_o = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \quad B_o = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}$$

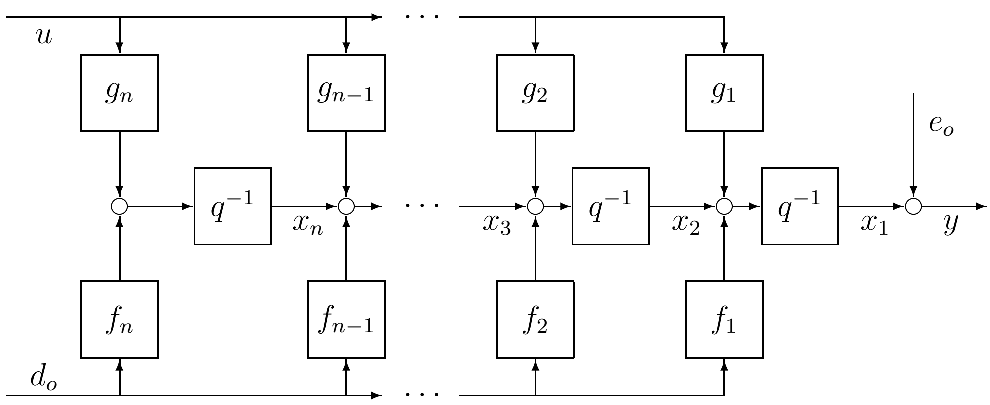

Figure 2.3: IO state space representation of impulse response model

$$C_o = \begin{bmatrix} I & 0 & 0 & \dots & 0 \end{bmatrix} \quad D_H = I \quad D_F = f_0$$

This IO system can be translated into an IIO model using the formulas from section 2.1:

$$A_i = \begin{bmatrix} I & I & 0 & \dots & 0 & 0 \\ 0 & 0 & I & & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & & I & 0 \\ 0 & 0 & 0 & \dots & 0 & I \\ 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix} \quad K_i = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad L_i = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \quad B_i = \begin{bmatrix} 0 \\ g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}$$

$$C_i = \begin{bmatrix} I & I & 0 & \dots & 0 \end{bmatrix} \quad D_H = I \quad D_F = f_0$$

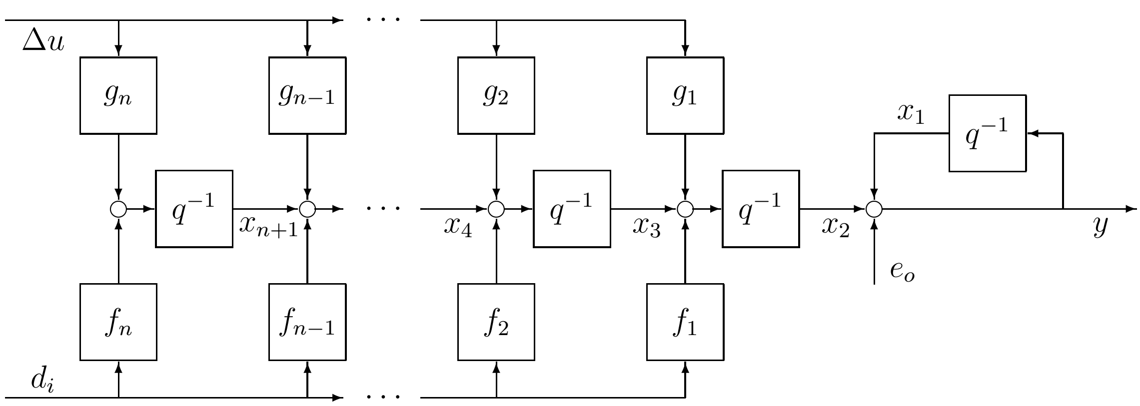

Figure 2.4: IIO state space representation of impulse response model

From step response model to state space model

(Lee *et al.* [69]) Let

$$
\begin{aligned}
G_i(q) &= s_1 q^{-1} + s_2 q^{-2} + \ldots + s_{n-1} q^{-n+1} + s_n q^{-n} + s_n q^{-n-1} + s_n q^{-n-2} + \ldots \\
&= s_1 q^{-1} + s_2 q^{-2} + \ldots + s_{n-1} q^{-n+1} + s_n \frac{q^{-n}}{1-q^{-1}} = \\
&= g_1 \frac{q^{-1}}{1-q^{-1}} + g_2 \frac{q^{-2}}{1-q^{-1}} + \ldots + g_n \frac{q^{-n}}{1-q^{-1}} \\
F_i(q) &= t_0 + t_1 q^{-1} + t_2 q^{-2} + \ldots + t_{n-1} q^{-n+1} + t_n q^{-n} + t_n q^{-n-1} + t_n q^{-n-2} + \ldots \\
&= t_0 + t_1 q^{-1} + t_2 q^{-2} + \ldots + t_{n-1} q^{-n+1} + t_n \frac{q^{-n}}{1-q^{-1}} \\
&= f_0 \frac{1}{1-q^{-1}} + f_1 \frac{q^{-1}}{1-q^{-1}} + f_2 \frac{q^{-2}}{1-q^{-1}} + \ldots + f_n \frac{q^{-n}}{1-q^{-1}} \\
H_i(q) &= \frac{I}{(1-q^{-1})}
\end{aligned}
$$

where s_i are the step response parameters of the system, then the state space matrices are given by:

$$
A_i = \begin{bmatrix} 0 & I & 0 & \ldots & 0 & 0 \\ 0 & 0 & I & & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & & I & 0 \\ 0 & 0 & 0 & \ldots & 0 & I \\ 0 & 0 & 0 & \ldots & 0 & I \end{bmatrix} \qquad K_i = \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} \qquad L_i = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix} \qquad B_i = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}
$$

$$
C_i = \begin{bmatrix} I & 0 & 0 & \ldots & 0 \end{bmatrix} \qquad D_H = I \qquad D_F = t_0 = f_0
$$

2.3 Polynomial models

Sometimes good models of the process can be obtained on the basis of physical laws or by parametric system identification. In that case a polynomial description of the model is preferred (Clarke *et al.*[21],[22]). Less parameters are used than in impulse or step response models and in the case of parametric system identification, the parameters can be estimated more reliably (Ljung [72]).

Consider the SISO IO polynomial model:

$$
G_o(q) = \frac{b_o(q)}{a_o(q)} \qquad F_o(q) = \frac{f_o(q)}{a_o(q)} \qquad H_o(q) = \frac{c_o(q)}{a_o(q)}
$$

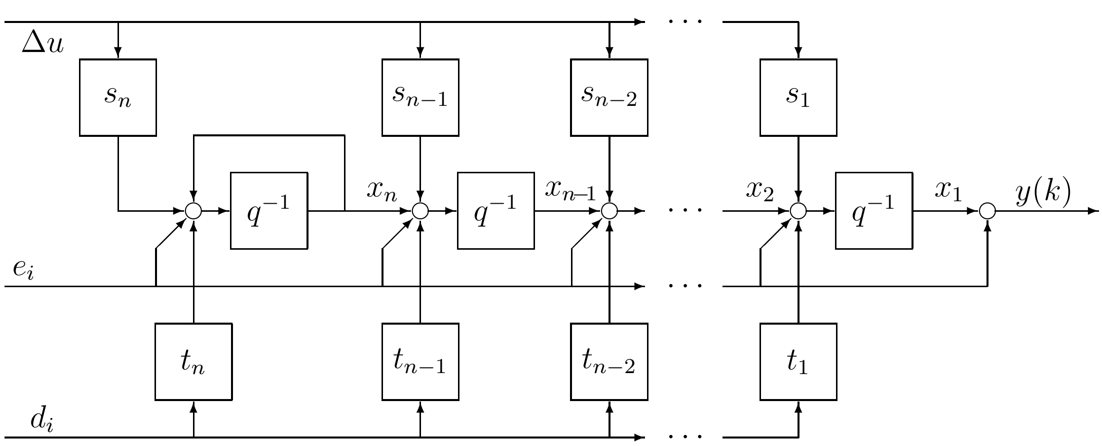

Figure 2.5: State space representation of step response model

where $a_o(q)$, $b_o(q)$ and $c_o(q)$ are polynomials in the operator q^{-1}

$$\begin{aligned}
a_o(q) &= 1 + a_{o,1}\, q^{-1} + \ldots + a_{o,n_a}\, q^{-n_a} \\
b_o(q) &= b_{o,1}\, q^{-1} + \ldots + b_{o,n_b}\, q^{-n_b} \\
c_o(q) &= 1 + c_{o,1}\, q^{-1} + \ldots + c_{o,n_c}\, q^{-n_c} \\
f_o(q) &= f_{o,0} + f_{o,1}\, q^{-1} + \ldots + f_{o,n_c}\, q^{-n_f}
\end{aligned} \tag{2.16}$$

The difference equation corresponding to the IO-model is given by the controlled autoregressive moving average (CARMA) model:

$$a_o(q)\, y(k) = b_o(q)\, u(k) + f_o(q)\, d_o(k) + c_o(q)\, e_o(k)$$

in which $e_o(k)$ is chosen as ZMWN.

Now consider the SISO IIO polynomial model (compare eq. 2.13):

$$G_i(q) = \frac{b_i(q)}{a_i(q)} \qquad F_i(q) = \frac{f_i(q)}{a_i(q)} \qquad H_i(q) = \frac{c_i(q)}{a_i(q)}$$

where $a_i(q) = a_o(q)\Delta(q) = a_o(q)(1 - q^{-1})$, $b_i(q) = b_o(q)$, $c_i(q) = c_o(q)$, and $f_i(q) = f_o(q)$, so

$$\begin{aligned}
a_i(q) &= 1 + a_{i,1}\, q^{-1} + \ldots + a_{i,n+1}\, q^{-n-1} \\
&= (1 - q^{-1})(1 + a_{o,1}\, q^{-1} + \ldots + a_{o,n}\, q^{-n}) \\
b_i(q) &= b_{i,1}\, q^{-1} + \ldots + b_{i,n}\, q^{-n} \\
&= b_{o,1}\, q^{-1} + \ldots + b_{o,n}\, q^{-n} \\
c_i(q) &= 1 + c_{i,1}\, q^{-1} + \ldots + c_{i,n}\, q^{-n} \\
&= 1 + c_{o,1}\, q^{-1} + \ldots + c_{o,n}\, q^{-n} \\
f_i(q) &= f_{i,0} + f_{i,1}\, q^{-1} + \ldots + f_{i,n}\, q^{-n} \\
&= f_{o,0} + f_{o,1}\, q^{-1} + \ldots + f_{o,n}\, q^{-n}
\end{aligned} \tag{2.17}$$

The difference equation corresponding to the IIO-model is given by the controlled autoregressive integrated moving average (CARIMA) model:

$$a_i(q)\, y(k) = b_i(q)\, \Delta u(k) + f_i(q)\, d_i(k) + c_i(q)\, e_i(k)$$

in which $d_i(k) = \Delta(q) d_o(k)$ and $e_i(k)$ is chosen as ZMWN.

From IO polynomial model to state space model

(Kailath [60]) Let

$$G_o(q) = \frac{b_o(q)}{a_o(q)} \qquad F_o(q) = \frac{f_o(q)}{a_o(q)} \qquad H_o(q) = \frac{c_o(q)}{a_o(q)}$$

where $a_o(q)$, $b_o(q)$, $f_o(q)$ and $c_o(q)$ are given in (2.16). Then the state space matrices are given by:

$$A_o = \begin{bmatrix} -a_{o,1} & 1 & 0 & \dots & 0 \\ -a_{o,2} & 0 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ -a_{o,n-1} & 0 & 0 & \ddots & 1 \\ -a_{o,n} & 0 & 0 & \dots & 0 \end{bmatrix} \qquad B_o = \begin{bmatrix} b_{o,1} \\ b_{o,2} \\ \vdots \\ b_{o,n-1} \\ b_{o,n} \end{bmatrix} \tag{2.18}$$

$$C_o = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix} \quad D_H = I \quad D_F = f_{o,0} \tag{2.19}$$

$$L_o = \begin{bmatrix} f_{o,1} - f_{0,o} a_{o,1} \\ f_{o,2} - f_{0,o} a_{o,2} \\ \vdots \\ f_{o,n-1} - f_{0,o} a_{o,n-1} \\ f_{o,n} - f_{0,o} a_{o,n} \end{bmatrix} \qquad K_o = \begin{bmatrix} c_{o,1} - a_{o,1} \\ c_{o,2} - a_{o,2} \\ \vdots \\ c_{o,n-1} - a_{o,n-1} \\ c_{o,n} - a_{o,n} \end{bmatrix} \tag{2.20}$$

A state space realization of a polynomial model is not unique. Any (linear nonsingular) state transformation gives a valid alternative. The above realization is called the observer canonical form (Kailath [60]).
We can also translate the IO polynomial model into an IIO state space model using the transformation formulas. We obtain the state space matrices:

$$A_i = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & -a_{o,1} & 1 & 0 & \dots & 0 \\ 0 & -a_{o,2} & 0 & 1 & & 0 \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & -a_{o,n-1} & 0 & 0 & \ddots & 1 \\ 0 & -a_{o,n} & 0 & 0 & \dots & 0 \end{bmatrix} \qquad B_i = \begin{bmatrix} 0 \\ b_{0,1} \\ b_{o,2} \\ \vdots \\ b_{o,n-1} \\ b_{o,n} \end{bmatrix} \tag{2.21}$$

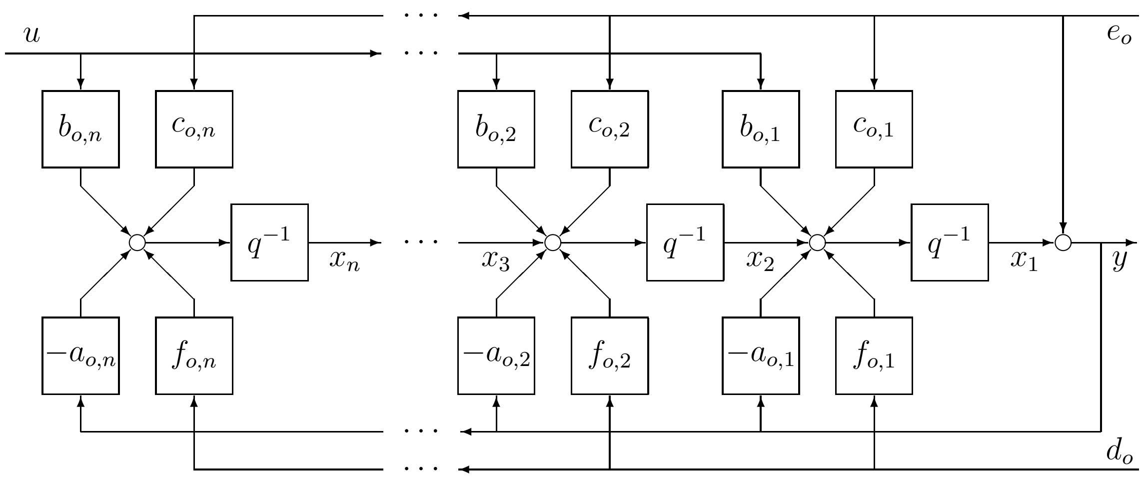

Figure 2.6: State space representation of IO polynomial model

$$C_i = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 \end{bmatrix} \quad D_H = I \quad D_F = f_{o,0} \tag{2.22}$$

$$L_i = \begin{bmatrix} 0 \\ f_{o,1} - f_{0,o}a_{o,1} \\ f_{o,2} - f_{0,o}a_{o,2} \\ \vdots \\ f_{o,n-1} - f_{0,o}a_{o,n-1} \\ f_{o,n} - f_{0,o}a_{o,n} \end{bmatrix} \qquad K_i = \begin{bmatrix} 1 \\ c_{o,1} - a_{o,1} \\ c_{o,2} - a_{o,2} \\ \vdots \\ c_{o,n-1} - a_{o,n-1} \\ c_{o,n} - a_{o,n} \end{bmatrix} \tag{2.23}$$

Example 1 : IO State space representation of IO polynomial model

In this example we show how to find the state space realization of an IO polynomial model. Consider the second order system

$$y(k) = G_o(q)\,u(k) + F_o(q)\,d_o(k) + H_o(q)\,e_o(k)$$

where $y(k)$ is the proces output, $u(k)$ is the proces input, $d_o(k)$ is a known disturbance signal and $e_o(k)$ is assumed to be ZMWN and

$$G_o(q) = \frac{q^{-1}}{(1-0.5q^{-1})(1-0.2q^{-1})} \qquad F_o(q) = \frac{0.1q^{-1}}{1-0.2q^{-1}} \qquad H_o(q) = \frac{1+0.5q^{-1}}{1-0.5q^{-1}}$$

To use the derived formulas, we have to give $G_o(q)$, $H_o(q)$ and $F_o(q)$ the common denominator $(1-0.5q^{-1})(1-0.2q^{-1}) = (1-0.7q^{-1}+0.1q^{-2})$:

$$F_o(q) = \frac{(0.1q^{-1})(1-0.5q^{-1})}{(1-0.2q^{-1})(1-0.5q^{-1})} = \frac{0.1q^{-1}-0.05q^{-2}}{(1-0.7q^{-1}+0.1q^{-2})}$$

$$H_o(q) = \frac{(1+0.5q^{-1})(1-0.2q^{-1})}{(1-0.5q^{-1})(1-0.2q^{-1})} = \frac{1+0.3q^{-1}-0.1q^{-2}}{(1-0.7q^{-1}+0.1q^{-2})}$$

A state-space representation for this example is now given by (2.18)-(2.20):

$$A_o = \begin{bmatrix} 0.7 & 1 \\ -0.1 & 0 \end{bmatrix} \qquad B_o = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad C_o = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$L_o = \begin{bmatrix} 0.1 \\ -0.05 \end{bmatrix} \qquad K_o = \begin{bmatrix} 1 \\ -0.2 \end{bmatrix} \qquad D_H = 1 \qquad D_F = 0$$

and so

$$\begin{aligned} x_{o1}(k+1) &= 0.7\, x_{o1}(k) + x_{o2}(k) + e_o(k) + 0.1\, d_o(k) + u(k) \\ x_{o2}(k+1) &= -0.1\, x_{o1}(k) - 0.2\, e_o(k) - 0.05\, d_o(k) \\ y(k) &= x_{o1}(k) + e_o(k) \end{aligned}$$

Example 2 : IIO State space representation of IO polynomial model

In this example we show how to find the IIO state space realization of the IO polynomial system of example 1 using (2.21)-(2.23):

$$A_i = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0.7 & 1 \\ 0 & -0.1 & 0 \end{bmatrix} \qquad B_i = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad C_i = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$$

$$L_i = \begin{bmatrix} 0 \\ 0.1 \\ -0.05 \end{bmatrix} \qquad K_i = \begin{bmatrix} 1 \\ 1 \\ -0.2 \end{bmatrix} \qquad D_H = 1 \qquad D_F = 0$$

and so

$$\begin{aligned} x_{i1}(k+1) &= x_{i1}(k) + x_{i2}(k) + e_i(k) \\ x_{i2}(k+1) &= 0.7\, x_{i2}(k) + x_{i3}(k) + e_i(k) + 0.1\, d_i(k) + \Delta u(k) \\ x_{i3}(k+1) &= -0.1\, x_{i2}(k) - 0.2\, e_i(k) - 0.05\, d_i(k) \\ y(k) &= x_{i1}(k) + x_{i2}(k) + e_i(k) \end{aligned}$$

From IIO polynomial model to state space model

(Kailath [60]) Let

$$G_i(q) = \frac{b_i(q)}{a_i(q)} \qquad F_i(q) = \frac{f_i(q)}{a_i(q)} \qquad H_i(q) = \frac{c_i(q)}{a_i(q)}$$

where $a_i(q)$, $b_i(q)$ and $c_i(q)$ are given in (2.17). Then the state space matrices are given by:

$$A_i = \begin{bmatrix} -a_{i,1} & 1 & 0 & \dots & 0 \\ -a_{i,2} & 0 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ -a_{i,n} & 0 & 0 & \ddots & 1 \\ -a_{i,n+1} & 0 & 0 & \dots & 0 \end{bmatrix} \qquad B_i = \begin{bmatrix} b_{i,1} \\ b_{i,2} \\ \vdots \\ b_{i,n} \\ 0 \end{bmatrix} \tag{2.24}$$

$$C_i = \left[\begin{array}{ccccc} 1 & 0 & 0 & \dots & 0 \end{array}\right] \tag{2.25}$$

$$L_i = \left[\begin{array}{c} f_{i,1} \\ f_{i,2} \\ \vdots \\ f_{i,n} \\ 0 \end{array}\right] \qquad K_i = \left[\begin{array}{c} c_{i,1} - a_{i,1} \\ c_{i,2} - a_{i,2} \\ \vdots \\ c_{i,n} - a_{i,n} \\ -a_{i,n+1} \end{array}\right] \tag{2.26}$$

Example 3 : IIO State space representation of IIO polynomial model

In this example we show how to find the state space realization of the system of example 1, rewritten as an IIO polynomial model: We consider the system:

$$y(k) = G_i(q)\,\Delta u(k) + F_i(q)\,d_i(k) + H_i(q)\,e_i(k)$$

where $y(k)$ is the proces output, $\Delta u(k)$ is the proces increment input, $d_i(k)$ is the increment disturbance signal and $e(k)$ is assumed to be integrated ZMWN and

$$\begin{aligned}
G_i(q) &= \frac{q^{-1}}{(1-q^{-1})(1-0.5q^{-1})(1-0.2q^{-1})} \\
&= \frac{q^{-1}}{(1-1.7q^{-1}+0.8q^{-2}-0.1q^{-3})} \\
F_i(q) &= \frac{0.1q^{-1}(1-0.5q^{-1})}{(1-q^{-1})(1-0.5q^{-1})(1-0.2q^{-1})} \\
&= \frac{0.1q^{-1}-0.05q^{-2}}{(1-1.7q^{-1}+0.8q^{-2}-0.1q^{-3})} \\
H_i(q) &= \frac{(1+0.5q^{-1})(1-0.2q^{-1})}{(1-q^{-1})(1-0.5q^{-1})(1-0.2q^{-1})} \\
&= \frac{1+0.3q^{-1}-0.1q^{-2}}{(1-1.7q^{-1}+0.8q^{-2}-0.1q^{-3})}
\end{aligned}$$

A state-space representation for this example is now given by (2.25)-(2.26):

$$A_i = \left[\begin{array}{ccc} 1.7 & 1 & 0 \\ -0.8 & 0 & 1 \\ 0.1 & 0 & 0 \end{array}\right] \qquad B_i = \left[\begin{array}{c} 1 \\ 0 \\ 0 \end{array}\right] \qquad C_i = \left[\begin{array}{ccc} 1 & 0 & 0 \end{array}\right]$$

$$L_i = \left[\begin{array}{c} 0.1 \\ -0.05 \\ 0 \end{array}\right] \qquad K_i = \left[\begin{array}{c} 2.0 \\ -0.9 \\ 0.1 \end{array}\right] \qquad D_H = 1 \qquad D_F = 0$$

and so

$$\begin{aligned}
x_{i1}(k+1) &= 1.7x_{i1}(k) + x_{i2}(k) + 2.0e_i(k) + 0.1d_i(k) + \Delta u(k) \\
x_{i2}(k+1) &= -0.8x_{i1}(k) + x_{i3}(k) - 0.9e_i(k) - 0.05d_i(k) \\
x_{i3}(k+1) &= 0.1x_{i1}(k) + 0.1e_i(k) \\
y(k) &= x_{i1}(k) + e_i(k)
\end{aligned}$$

Clearly, the IIO state space representation in example 3 is different from the IIO state space representation in example 2. By denoting the state space matrices of example 2 with a hat (so $\hat{A}_i$, $\hat{B}_i$, $\hat{C}_i$, $\hat{L}_i$, $\hat{K}_i$,$\hat{D}_H$,$\hat{D}_F$), the relation between the two state space representations is given by:

$$\begin{array}{rclrclll} \hat{A}_i & = & T^{-1}A_iT & \hat{C}_i & = & C_iT & \hat{D}_H = D_H & \hat{D}_F = D_F \\ \hat{B}_i & = & T^{-1}B_i & \hat{L}_i & = & T^{-1}L_i & \hat{K}_i & = \; T^{-1}K_i \end{array}$$

where state transformation matrix T is given by:

$$T = \begin{bmatrix} 1 & 1 & 0 \\ -0.7 & 0 & 1 \\ 0.1 & 0 & 0 \end{bmatrix}$$

2.4 Modeling of relevant process dynamics

The classical approach for modeling of relevant unit process dynamics is rigorous modeling. Rigorous modeling implies the construction of a mathematical model of a system on the basis of fundamental physical, chemical, biological mechanisms. As a result, in general, a set of Differential Algebraic Equations (DAEs) is obtained. These rigorous modeling techniques mostly appear not to be very well suited for accurate modeling of all relevant process dynamics as required for the design of high performance model based control systems, although they provide a good understanding of the main process mechanisms. Approximate modeling using first or second order transfer function models with time delays have been and still are broadly applied to model dynamic behavior of processes for the purpose of control system design. Also these techniques have their restrictions however, if used for the purpose of high performance control system design. Such models can only describe a small part of the process dynamics that are available for actual control of the process. Preferably the approximate models applied for process control have to describe all relevant process dynamics. Process dynamics are considered relevant for control in this context, if they can be excited through feasible input manipulations and if the resulting process responses can be observed at the controlled process outputs.
Since the end of the fifties people started investigating techniques for the modeling of process dynamics on the basis of observed system input output behavior (e.g. [3], [43], [53], [58], [135], [137]). After this initial research many more people started working in this area of system identification. Two main streams may be discriminated in the process identification techniques developed. One stream refers to the developments on the basis of a stochastic framework using low order parametric type models applied mainly in research done at universities. The second stream is based on the use of high order non-parametric type models in combination with a deterministic modeling approach, which was originating from industrial applied research related to model predictive control developments.
Early seventies the process identification techniques for single input, single output system identification on the basis of a stochastic approach were rather well understood ([3], [43]). These process identification techniques have matured to a level that starts suiting the

requirements for accurate modeling of relevant dynamics of industrial processes in the period from 1980 to 1995 [6], [7], [30], [44], [46], [55], [57], [73], [72], [81], [92], [103], [107], [113], [116], [119], [123], [122], [127], [128], [130], [131], [132], [142], [143],). Some of these techniques enable true multi-input multi-output system identification (e.g. [6], [44], [55], [119], [123], [122], [142], [143],).
The advanced system identification techniques that are applied in industry for the design of high performance model based control systems consist of a multi-step approach. An example of this approach is described in [6], [7]. Essentially one to three subsequent process identification steps are applied (cf. fig. 2.7):

- Non-parametric model identification
- Semi-parametric model approximation
- Semi-parametric model identification

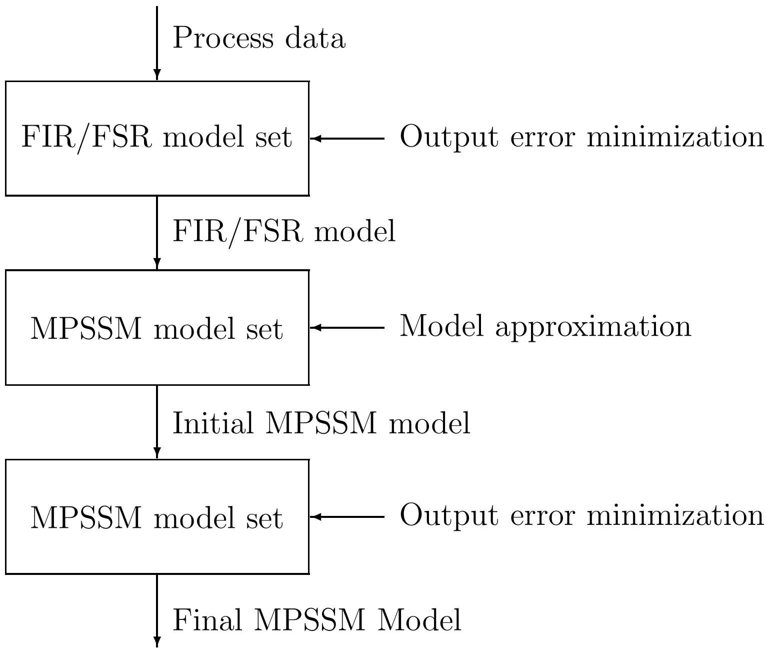

Figure 2.7: Multistep process identification procedure

The first step involves the estimation of the parameters of a Finite Step Response (FSR) or a Finite Impulse Response (FIR) model:

$$y(k) = \sum_{j=0}^{n} s(j)\,\Delta u(k-j) \qquad \text{(FSR)} \tag{2.27}$$

$$y(k) = \sum_{j=0}^{n} g(j)\,u(k-j) \qquad \text{(FIR)} \tag{2.28}$$

$s(j)$ denotes the step response parameters and $g(j)$ are the impulse response parameters in these expressions. A quadratic output error criterion function is minimized for the estimation of the model parameters. This output error criterion function is defined as:

$$V = \| Y_p - Y_m(g(j)|j \in I, 0 \leq j \leq n) \|_2^2 \tag{2.29}$$

Y_p denotes the matrix of measured process outputs during an identification test, whereas Y_m represents the matrix of outputs simulated by the model over the same time interval. Ym is a function of the impulse response or step response model parameters according to (2.27) or (2.28).

The criterion function V is a quadratic function of the model parameters for a FSR or a FIR model. As this is a convex function, the minimum is unique and can easily be found by solving a set of linear equations. It can easily be shown that the estimated model parameters converge to the true parameters of the model that exactly represents the linearized process behavior in the selected operating point, if the following conditions are satisfied:

1. The applied input test signal is not correlated with the process output noise ($\Psi_{un} = 0$)
2. The data set applied for parameter estimation is sufficiently long
3. The applied input test signal is sufficiently rich (e.g. $\Psi_{uu} = I$)

This non-parametric type of model is most widely applied in today's industrial applications of Model Predictive Control systems. It is not very well suited for the design of high performance Model Predictive Control systems however. Although the non-parametric model describes the process behavior well, this type of model has a major drawback: The number of parameters required for describing all relevant process dynamics can be extremely large. The complexity of the model is directly related to the range of dynamics that it has to cover. The sample rate required is linked to the highest frequencies that have to be described by the model, whereas the number of parameters is dictated by the time it takes for the process to reach steady state. This implies that the number n -the length of the FIR or the FSR- can become very large, if the fastest and the slowest relevant process dynamics are far apart. This is the case in most processes. Typical ranges of relevant process dynamics of industrial processes are 100 up to 10.000, which require n to be 1000 to 100.000 to cover such ranges. Semi-parametric and parametric models can overcome this problem. A model set that also directly describes impulse responses, but only requires a limited number of parameters is the Minimal Polynomial Start Sequence of Markov Parameter (MPSSM) model set:

$$y(k) = \sum_{j=0}^{\infty} F(j)\, u(k-j) \tag{2.30}$$

$$F(j) = \begin{cases} g(0) & j = 0 \\ g(j) & 1 \leq j \leq r \\ \sum_{i=1}^{r} a_i\, F(j-i) & j > r \end{cases} \tag{2.31}$$

This model set is called semi-parametric, while the order of the model is restricted already by the degree r of the minimal polynomial and therefore parametric, but the structure of the multi-input multi-output system is still treated in a non-parametric way by the start sequence of impulse response parameters $F(0)$, $F(1)$, ..., $F(r)$. This model reconstructs a complete infinite impulse response from the initial impulse response parameters using the minimal polynomial. This minimal polynomial describes the main dynamic modes of the process. The roots of this polynomial are the relevant poles of the process dynamics.
In the second step of the identification procedure an approximate MPSSM model is fitted to the FIR or FSR model obtained from the first step. Model approximation techniques like Hankel realization (e.g. Zeiger [137]) or subspace identification techniques (e.g. Van Overschee [123],[122]) can be applied for this purpose.
In the third step of the identification procedure the MPSSM model parameters are further optimized by minimization of the quadratic output error criterion function V_1:

$$V_1 = \| Y_p - Y_{MPSSM} \|_2^2 = \| Y_p - Y(F(0), F(j), a_j | j \in I, 1 \leq j \leq r) \|_2^2 \tag{2.32}$$

As this function is a high order polynomial function of the minimal polynomial coefficients, the function is no longer convex. The initial identification steps enable initialization of the model parameters sufficiently close to the global minimum of the function. This ensures that the model parameters obtained from minimization of V_1 result in the model looked for.
The MPSSM model obtained from this final optimization can be translated to a corresponding State Space model e.g. by means of infinite Hankel realization based model approximation techniques (e.g. [57]).

2.5 Choice of model description

Whatever model is used to describe the process behaviour, the resulting predictive control law will (at least approximately) give the same performance. Differences in the use of a model are in the effort of modeling, the model accuracy that can be obtained, and in the computational power that is needed to realize the predictive control algorithm.
Finite step response and finite impulse response models (non-parametric models) need more parameters than Minimal Polynomial Start Sequence Markov Parameter models (semi-parametric models), polynomial or state space models (parametric models). The larger the number of parameters of the model the lower the average number of datapoints per parameter that needs to be estimated during identification. As a consequence the variance of the estimated parameters will be larger for the parameters of non-parametric models than for semi-parametric and parametric models. At its turn also the variance of estimated parameters of semi-parametric models will be larger than the variance of the parameters of parametric models. This does not necessarily imply that the quality of the predictions with parametric models outperforms the predictions of non-parametric models! If the test signals applied for process identification have excited all relevant dynamics of the process persistently, the qualities of the predictions are generally the same despite the significantly

larger variance of the estimated model parameters. On the other hand step response models are intuitive, need less a priori information for identification and the predictions are constructed in a natural way. The models can be understood without any knowledge of dynamical systems. Step response models are often still preferred in industry, because advanced process knowledge is usually scarce and additional experiments are expensive.
The polynomial description is very compact, gives good (physical) insight in the systems properties and the resulting controller is compact as well. However, as we will see later, making predictions may be cumbersome and the models are far from practical for multivariable systems.
State space models are especially suited for multivariable systems, still providing a compact model description and controller. The computations are usually well conditioned and the algorithms easy to implement.

Chapter 3

Prediction

In this chapter we consider the concept of prediction. We consider the following model:

$$\begin{aligned} x(k+1) &= A\,x(k) + B_1\,e(k) + B_2\,w(k) + B_3\,v(k) \\ y(k) &= C_1\,x(k) + D_{11}\,e(k) + D_{12}\,w(k) \\ z(k) &= C_2\,x(k) + D_{21}\,e(k) + D_{22}\,w(k) + D_{23}\,v(k) \end{aligned} \tag{3.1}$$

where $x(k)$ is the state, $e(k)$ is zero-mean white noise (ZMWN), $v(k)$ is the control signal, $w(k)$ is a vector containing all known external signals, such as the reference signal and known (or measurable) disturbances, and $z(k)$ is a signal, to be predicted. The control signal $v(k)$ can be either $u(k)$ or $\Delta u(k)$. The prediction signal $z(k)$ can be any signal which can be formulated as in expression 3.1 (so the state $x(k)$, the output signal $y(k)$ or tracking error $r(k) - y(k)$). We will come back on the particular choice of the above model in the next chapter.

The control law in predictive control is, no surprise, based on prediction. At each time instant k, the performance index over the horizon N is considered by making a set of j-step ahead predictions of the signal $z(k+j)$. This prediction, denoted as $\hat{z}(k+j|k)$, is based on model (3.1), using information given at time k and future values of the control signal $v(k+j|k)$. As soon as a prediction algorithm is available, the optimal moves of the control signal can be selected such that the performance index J is minimized (chapter 5).
The predictions $\hat{z}(k+j|k)$ of the performance signal $z(k+j)$ are based on the knowledge at time k and the future control signals $v(k|k)$, $v(k{+}1|k)$, $\ldots$, $v(k{+}N{-}1|k)$.
At time instant k we define the signal vector $\tilde{z}(k)$ with the predicted performance signals $z(k+j|k)$, the signal vector $\tilde{v}(k)$ with the future control signals $v(k+j|k)$ and the signal vector $\tilde{w}(k)$ with the future reference signals $w(k+j|k)$ in the interval $0 \le j \le N{-}1$ as follows

$$\tilde{z}(k) = \begin{bmatrix} \hat{z}(k|k) \\ \hat{z}(k{+}1|k) \\ \ldots \\ \hat{z}(k{+}N{-}1|k) \end{bmatrix} \quad \tilde{v}(k) = \begin{bmatrix} v(k|k) \\ v(k{+}1|k) \\ \ldots \\ v(k{+}N{-}1|k) \end{bmatrix} \quad \tilde{w}(k) = \begin{bmatrix} w(k|k) \\ w(k{+}1|k) \\ \ldots \\ w(k{+}N{-}1|k) \end{bmatrix}$$

The goal of the prediction is to find an estimate for the prediction signal $\tilde{z}(k)$, composed of a free-response signal $\tilde{z}_0(k)$ and a matrix $\tilde{D}_{23}$ such that

$$\tilde{z}(k) = \tilde{z}_0(k) + \tilde{D}_{23}\,\tilde{v}(k)$$

where the term $\tilde{D}_{23}\,\tilde{v}(k)$ is the part of $\tilde{z}(k)$, contributed by the selected control signal $v(k+j|k)$ for $j \geq 0$ and $\tilde{z}_0(k)$ is the so called free-response. This free-response $\tilde{z}_0(k)$ is the predicted output signal when the future input signal is put to zero ($v(k+j|k) = 0$ for $j \geq 0$) and depends on the present state $x(k)$, the future values of the known signal $w(k)$ and the noise signal $e(k)$.

3.1 Noiseless case

The prediction mainly consists of two parts: Computation of the output signal as a result of the selected control signal $v(k+j|k)$ and the known signal $w(k)$ and a prediction of the response due to the noise signal $e(k)$. In this section we will assume the noiseless case (so $e(k) = 0$) and we only have to concentrate on the response due to the present and future values $v(k+j|k)$ of the input signal and known signal $w(k)$.
Consider the model from (3.1) for the noiseless case ($e(k) = 0$):

$$\begin{aligned} x(k+1) &= Ax(k) + B_2 w(k) + B_3 v(k) \\ z(k) &= C_2 x(k) + D_{22} w(k) + D_{23} v(k) \end{aligned}$$

For this state space model the prediction of the state is computed using successive substitution (Kinnaert [62]):

$$\begin{aligned} x(k+j|k) &= Ax(k+j-1|k) + B_2\, w(k+j-1|k) + B_3\, v(k+j-1|k) = \\ &\quad A^2 x(k+j-2|k) + A\, B_2\, w(k+j-2|k) + B_2\, w(k+j-1|k) + \\ &\quad + A\, B_3\, v(k+j-2|k) + B_3\, v(k+j-1|k) \\ &\ \ \vdots \\ &= A^j x(k) + \sum_{i=1}^{j} A^{i-1} B_2 w(k+j-i|k) + \sum_{i=1}^{j} A^{i-1} B_3 v(k+j-i|k) \\ z(k+j|k) &= C_2\, x(k+j|k) + D_{22} w(k+j|k) + D_{23} v(k+j|k) \\ &= C_2\, A^j x(k) + \sum_{i=1}^{j} C_2\, A^{i-1} B_2 w(k+j-i|k) + D_{22} w(k+j|k) + \\ &\quad + \sum_{i=1}^{j} C_2\, A^{i-1} B_3 v(k+j-i|k) + D_{23} v(k+j|k) \end{aligned}$$

Now define

$$\tilde{C}_2 = \begin{bmatrix} C_2 \\ C_2 A \\ C_2 A^2 \\ \vdots \\ C_2 A^{N-1} \end{bmatrix} \qquad \tilde{D}_{22} = \begin{bmatrix} D_{22} & 0 & \cdots & 0 & 0 \\ C_2 B_2 & D_{22} & \cdots & 0 & 0 \\ C_2 A B_2 & C_2 B_2 & \ddots & \vdots & \vdots \\ \vdots & & \ddots & D_{22} & 0 \\ C_2 A^{N-2} B_2 & & \cdots & C_2 B_2 & D_{22} \end{bmatrix}$$

$$\tilde{D}_{23} = \begin{bmatrix} D_{23} & 0 & \cdots & 0 & 0 \\ C_2 B_3 & D_{23} & \cdots & 0 & 0 \\ C_2 A B_3 & C_2 B_3 & \ddots & \vdots & \vdots \\ \vdots & & \ddots & D_{23} & 0 \\ C_2 A^{N-2} B_3 & & \cdots & C_2 B_3 & D_{23} \end{bmatrix}$$

then the vector $\tilde{z}(k)$ with the predicted output values can be written as

$$\begin{aligned} \tilde{z}(k) &= \tilde{C}_2\, x(k) + \tilde{D}_{22}\, \tilde{w}(k) + \tilde{D}_{23}\, \tilde{v}(k) \\ &= \tilde{z}_0(k) + \tilde{D}_{23}\, \tilde{v}(k) \end{aligned}$$

In this equation $\tilde{z}_0(k) = \tilde{C}_2\, x(k) + \tilde{D}_{22}\, \tilde{w}(k)$ is the prediction of the output if $\tilde{v}(k)$ is chosen equal to zero, denoted as the free-run performance signal, and $\tilde{D}_{23}$ is the predictor matrix, describing the relation between future control vector $\tilde{v}(k)$ and predicted output $\tilde{z}(k)$. Matrices $\tilde{D}_{22}$ and $\tilde{D}_{23}$ are Toeplitz matrices. These matrices respectively relate future disturbances and future inputs to future output signals.

The matrix $\tilde{D}_{23}$ contains the elements D_{23}, and $C_2 A^j B_3$, $j \geq 0$, which are exactly equal to the impulse response parameters g_i of the transfer function $P_{23}(q) = C_2(qI - A)^{-1} B_3 + D_{23}$, which describes the input-output relation between the signals $v(k)$ and $z(k)$.
The impulse response parmeters of P_{23} are given by:

$$g_i = \begin{cases} D_{23} & \text{for} \quad i = 0 \\ C_2 A^{i-1} B_3 & \text{for} \quad i > 0 \end{cases}$$

3.2 Noisy case

As was stated in the previous section, the second part of the prediction scheme is making a prediction of the response due to the noise signal $e(k)$. If we know the characteristics of the noise signal, this information can be used for prediction of the expected noise signal over the future horizon. In case we do not know anything about the chararcteristics of the disturbances, the best assumption we can make is the assumption $e(k)$ is zero-mean white noise. In these lecture notes we use this assumption of zero-mean white noise $e(k)$, so the best prediction for future values of signal $e(k+j)$, $j > 0$ will be zero ($\hat{e}(k+j) = 0$ for $j > 0$).

Consider the model given in equation (3.1):

$$\begin{aligned} x(k+1) &= A\,x(k) + B_1\,e(k) + B_2\,w(k) + B_3\,v(k) \\ y(k) &= C_1\,x(k) + D_{11}\,e(k) + D_{12}\,w(k) \\ z(k) &= C_2\,x(k) + D_{21}\,e(k) + D_{22}\,w(k) + D_{23}\,v(k) \end{aligned}$$

the prediction of the state using successive substitution of the state equation is given by (Kinnaert [62]):

$$\begin{aligned} \hat{x}(k+j|k) &= A^j x(k) + \sum_{i=1}^{j} A^{i-1} B_1 \hat{e}(k+j-i|k) + \\ &\quad + \sum_{i=1}^{j} A^{i-1} B_2 w(k+j-i|k) + \sum_{i=1}^{j} A^{i-1} B_3 v(k+j-i|k) \\ \hat{z}(k+j|k) &= C_2\,\hat{x}(k+j|k) + D_{21}\hat{e}(k+j|k) + D_{22} w(k+j|k) + D_{23} v(k+j|k) \end{aligned}$$

The term $\hat{e}(k+m|k)$ for $m>0$ is a prediction of $e(k+m)$, based on measurements at time k. If the signal e is assumed to be zero mean white noise, the best prediction of $e(k+m)$, made at time k, is

$$\hat{e}(k+m|k) = 0 \qquad \text{for } m>0$$

Substitution then results in ($j>0$)

$$\begin{aligned} \hat{x}(k+j|k) &= A^j x(k) + \sum_{i=1}^{j} A^{i-1} B_3 v(k+j-i|k) + \sum_{i=1}^{j} A^{i-1} B_2 w(k+j-i|k) + A^{j-1} B_1 \hat{e}(k|k) \\ \hat{z}(k+j|k) &= C_2\,\hat{x}(k+j|k) + D_{22} w(k+j|k) + D_{23} v(k+j|k) \\ &= C_2\,A^j x(k) + C_2 A^{j-1} B_1 \hat{e}(k|k) + \\ &\quad + \sum_{i=1}^{j} C_2\,A^{i-1} B_2 w(k+j-i|k) + D_{22} w(k+j|k) \\ &\quad + \sum_{i=1}^{j} C_2\,A^{i-1} B_3 v(k+j-i|k) + D_{23} v(k+j|k) \\ \hat{z}(k|k) &= C_2\,x(k) + D_{21}\hat{e}(k|k) + D_{22} w(k) + D_{23} v(k) \end{aligned}$$

Define

$$\tilde{D}_{21} = \begin{bmatrix} D_{21} \\ C_2 B_1 \\ C_2 A B_1 \\ \vdots \\ C_2 A^{N-2} B_1 \end{bmatrix}$$

then, the vector $\tilde{z}$ with the predicted output values can be written as

$$
\begin{aligned}
\tilde{z}(k) &= \tilde{C}_2\, x(k) + \tilde{D}_{21}\, \hat{e}(k|k) + \tilde{D}_{22}\, \tilde{w}(k) + \tilde{D}_{23}\, \tilde{v}(k) \\
&= \tilde{z}_0(k) + \tilde{D}_{23}\, \tilde{v}(k)
\end{aligned} \tag{3.2}
$$

In this equation the free-run performance signal is equal to

$$
\tilde{z}_0(k) = \tilde{C}_2\, x(k) + \tilde{D}_{21}\, \hat{e}(k|k) + \tilde{D}_{22}\, \tilde{w}(k) \tag{3.3}
$$

Now, assuming D_{11} to be invertible, we derive an expression for $\hat{e}(k)$ from (3.1):

$$
\hat{e}(k|k) = D_{11}^{-1}\Big(y(k) - C_1\, x(k) - D_{12}\, w(k)\Big)
$$

in which $y(k)$ is the measured process output and $x(k)$ is the estimate of state $x(k)$ at previous time instant $k-1$.

$$
\begin{aligned}
\tilde{z}_0(k) &= \tilde{C}_2\, x(k) + \tilde{D}_{21}\, \hat{e}(k) + \tilde{D}_{22}\, \tilde{w}(k) = && (3.4)\\
&= \tilde{C}_2\, x(k) + \tilde{D}_{21}\, D_{11}^{-1}\Big(y(k) - C_1\, x(k) - D_{12}\, w(k)\Big) + \tilde{D}_{22}\, \tilde{w}(k) = && (3.5)\\
&= (\tilde{C}_2 - \tilde{D}_{21} D_{11}^{-1} C_1)\, x(k) + \tilde{D}_{21}\, D_{11}^{-1} y(k) + (\tilde{D}_{22} - \tilde{D}_{21} D_{11}^{-1} D_{12} E_w)\, \tilde{w}(k) && (3.6)
\end{aligned}
$$

where $E_w = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix}$ is a selection matrix such that

$$
w(k) = E_w \tilde{w}(k)
$$

Example 4 : Prediction of the output signal of a system

In this example we consider the prediction of the output signal $z(k) = y(k)$ of the second order system from example 1, page 45.
We have given

$$
\begin{aligned}
x_{o1}(k+1) &= 0.7\, x_{o1}(k) + x_{o2}(k) + e_o(k) + 0.1\, d_o(k) + u(k) \\
x_{o2}(k+1) &= -0.1\, x_{o1}(k) - 0.2\, e_o(k) - 0.05\, d_o(k) \\
y(k) &= x_{o1}(k) + e_o(k) \\
z(k) &= x_{o1}(k) + e_o(k)
\end{aligned}
$$

where $y(k)$ is the proces output, $v(k) = u(k)$ is the proces input, $w(k) = d(k)$ is a known disturbance signal and $e(k)$ is assumed to be ZMWN. We obtain the model of (3.1) by setting

$$
A = \begin{bmatrix} 0.7 & 1 \\ -0.1 & 0 \end{bmatrix} \qquad B_1 = \begin{bmatrix} 1 \\ -0.2 \end{bmatrix} \qquad B_2 = \begin{bmatrix} 0.1 \\ -0.05 \end{bmatrix} \qquad B_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}
$$

$$
C_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad C_2 = \begin{bmatrix} 1 & 0 \end{bmatrix}
$$

$$
D_{11} = \begin{bmatrix} 1 \end{bmatrix} \qquad D_{12} = \begin{bmatrix} 0 \end{bmatrix} \qquad D_{21} = \begin{bmatrix} 1 \end{bmatrix} \qquad D_{22} = \begin{bmatrix} 0 \end{bmatrix} \qquad D_{23} = \begin{bmatrix} 0 \end{bmatrix}
$$

For $N=4$ we find:

$$\tilde{C}_2 = \begin{bmatrix} C_2 \\ C_2 A \\ C_2 A^2 \\ C_2 A^3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0.7 & 1.0 \\ 0.39 & 0.7 \\ 0.203 & 0.39 \end{bmatrix}$$

$$\tilde{D}_{21} = \begin{bmatrix} D_{21} \\ C_2 B_1 \\ C_2 A B_1 \\ C_2 A^2 B_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0.5 \\ 0.25 \end{bmatrix}$$

$$\tilde{D}_{22} = \begin{bmatrix} D_{22} & 0 & 0 & 0 \\ C_2 B_2 & D_{22} & 0 & 0 \\ C_2 A B_2 & C_2 B_2 & D_{22} & 0 \\ C_2 A^2 B_2 & C_2 A B_2 & C_2 B_2 & D_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0 \\ 0.02 & 0.1 & 0 & 0 \\ 0.004 & 0.02 & 0.1 & 0 \end{bmatrix}$$

$$\tilde{D}_{23} = \begin{bmatrix} D_{23} & 0 & 0 & 0 \\ C_2 B_3 & D_{23} & 0 & 0 \\ C_2 A B_3 & C_2 B_3 & D_{23} & 0 \\ C_2 A^2 B_3 & C_2 A B_3 & C_2 B_3 & D_{23} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0.7 & 1 & 0 & 0 \\ 0.39 & 0.7 & 1 & 0 \end{bmatrix}$$

3.3 Prediction using step response models

prediction for infinite-length step response model:

The step response model has been derived in section 2.2 and is given by:

$$y(k) = \sum_{m=1}^{\infty} s_m \, \Delta u(k-m) + \sum_{m=0}^{\infty} t_m \, d_i(k-m) + \sum_{m=0}^{\infty} e_i(k-m) \tag{3.7}$$

where y is the output signal, Δu is the increment of the input signal u, d_i is the increment of the (known) disturbance signal d_o, and e_i, which is the increment of the output noise signal e_o, is assumed to be zero-mean white noise.
Note that the output signal $y(k)$ in equation (3.7) is build up with all past values of the input, disturbance and noise signals. We introduce a signal $y_0(k|k-\ell)$, $\ell > 0$, which gives the value of $y(k)$ based on the inputs, disturbances and noise signals up to time $k-\ell$, assuming future values to be zero, so $\Delta u(m)=0$, $d_i(m)=0$ and $e_i(m)=0$ for $m > k-\ell$:

$$y_0(k|k-\ell) = \sum_{m=0}^{\infty} s_{\ell+m} \, \Delta u(k-\ell-m) + \sum_{m=0}^{\infty} t_{\ell+m} \, d_i(k-\ell-m) + \sum_{m=0}^{\infty} e_i(k-\ell-m) \tag{3.8}$$

Based on the above definition it easy to derive a recursive expression for y_0:

$$\begin{aligned}
y_0(k|k-\ell) &= \sum_{m=1}^{\infty} s_{\ell+m}\,\Delta u(k-\ell-m) + \sum_{m=1}^{\infty} t_{\ell+m}\,d_i(k-\ell-m) \\
&\quad + \sum_{m=1}^{\infty} e_i(k-\ell-m) + s_\ell\,\Delta u(k-\ell) + t_\ell\,d_i(k-\ell) + e_i(k-\ell) \\
&= y_0(k|k-\ell-1) + s_\ell\,\Delta u(k-\ell) + t_\ell\,d_i(k-\ell) + e_i(k-\ell)
\end{aligned}$$

Future values of y can be derived as follows:

$$\begin{aligned}
y(k+j) &= \sum_{m=1}^{\infty} s_m\,\Delta u(k+j-m) + \sum_{m=0}^{\infty} t_m\,d_i(k+j-m) + \sum_{m=0}^{\infty} e_i(k+j-m) \\
&= \sum_{m=1}^{\infty} s_{j+m}\,\Delta u(k-m) + \sum_{m=1}^{\infty} t_{j+m}\,d_i(k-m) + \sum_{m=1}^{\infty} e_i(k-m) \\
&\quad + \sum_{m=1}^{j} s_m\,\Delta u(k+j-m) + \sum_{m=0}^{j} t_m\,d_i(k+j-m) + \sum_{m=0}^{j} e_i(k+j-m) \\
&= y_0(k+j|k-1) \\
&\quad + \sum_{m=1}^{j} s_m\,\Delta u(k+j-m) + \sum_{m=0}^{j} t_m\,d_i(k+j-m) + \sum_{m=0}^{j} e_i(k+j-m)
\end{aligned}$$

Suppose we are at time k, then we only have knowlede up to time k, so $e_i(k+j)$, $j>0$ is unknown. The best estimate we can make of the zero-mean white noise signal for future time instants is (see also previous section):

$$\hat{e}_i(k+j|k) = 0 \quad ,\text{for } \; j>0$$

If we assume $\Delta u(\ell)$ and $d_i(\ell)$ to be known in the interval $k \le \ell \le k+j$ we obtain the prediction of $y(k+j)$, based on knowledge until time k, and denoted as $\hat{y}(k+j|k)$:

$$\hat{y}(k+j|k) = y_0(k+j|k-1) + \sum_{m=1}^{j} s_m\,\Delta u(k+j-m) + \sum_{m=0}^{j} t_m\,d_i(k+j-m) + e_i(k)$$

prediction for truncated step response model:

We derived the following relations:

$$\begin{aligned}
y_0(k|k-\ell) &= y_0(k|k-\ell-1) + s_{\ell+m}\,\Delta u(k-\ell) + t_{\ell+m}\,d_i(k-\ell) \\
y(k) &= y_0(k|k-1) + t_0\,d_i(k) + e_i(k) \\
\hat{y}(k+j|k) &= y_0(k+j|k-1) + \sum_{m=1}^{j} s_m\,\Delta u(k+j-m) + \sum_{m=0}^{j} t_m\,d_i(k+j-m) + e_i(k)
\end{aligned}$$

In section 2.2 the truncated step response model was introduced. For a model with step-response length n there holds:

$$s_m = s_n \qquad \text{for} \qquad m > n$$

Using this property we can derive from 3.8 for $j \geq 0$:

$$\begin{aligned}
y_0(k{+}n{+}j|k-1) &= \sum_{m=1}^{\infty} s_{n+m+j}\,\Delta u(k-m) + \sum_{m=1}^{\infty} t_{n+m+j}\,d_i(k{-}m) + \sum_{m=1}^{\infty} e_i(k{-}m) \\
&= \sum_{m=1}^{\infty} s_n\,\Delta u(k{-}m) + \sum_{m=1}^{\infty} t_n\,d_i(k{-}m) + \sum_{m=1}^{\infty} e_i(k{-}m) \\
&= y_0(k{+}n{+}j{-}1|k-1)
\end{aligned}$$

and so $y_0(k{+}n{+}j|k{-}1) = y_0(k{+}j{-}1|k{-}1)$ for $j \geq 0$. This means that for computation of $\hat{y}(k{+}n{+}j|k)$ where $j \geq 0$, so when the prediction is beyond the length of the step-response we can use the equation:

$$\begin{aligned}
\hat{y}(k+n+j|k) &= y_0(k+n-1|k-1) + \sum_{m=1}^{n+j} s_m\,\Delta u(k+n+j-m) \\
&\quad + \sum_{m=0}^{n+j} t_m\,d_i(k+n+j-m) + e_i(k)
\end{aligned}$$

Matrix notation

From the above derivation we found that computation of $y_0(k{+}j|k{-}1)$ for $j = 0, \ldots, n{-}1$ is sufficient to make predictions for all $j \leq 0$. We obtain the equations:

$$\begin{aligned}
&y_0(k{+}j|k) = y_0(k{+}j|k{-}!1) + s_j\Delta u(k) + t_j d_i(k) + e_i(k) && \text{for} \quad 0 \leq j \leq n{-}1 \\
&y_0(k{+}j|k) = y_0(k{+}n{-}1|k{-}1) + s_n\Delta u(k) + t_j d_i(k) + e_i(k) && \text{for} \quad j \geq n \\
&y(k|k) = y_0(k|k{-}1) + t_0 d_i(k) + e_i(k)
\end{aligned} \tag{3.9}$$

In matrix form:

$$\begin{bmatrix} y_0(k{+}1|k) \\ y_0(k{+}2|k) \\ \vdots \\ y_0(k{+}n{-}1|k) \\ y_0(k{+}n|k) \end{bmatrix} = \begin{bmatrix} y_0(k{+}1|k{-}1) \\ y_0(k{+}2|k{-}1) \\ \vdots \\ y_0(k{+}n{-}1|k{-}1) \\ y_0(k{+}n{-}1|k{-}1) \end{bmatrix} + \begin{bmatrix} s_1\Delta u(k) \\ s_2\Delta u(k) \\ \vdots \\ s_{n-1}\Delta u(k) \\ s_n\Delta u(k) \end{bmatrix} + \begin{bmatrix} t_1 d_i(k) \\ t_2 d_i(k) \\ \vdots \\ t_{n-1} d_i(k) \\ t_n d_i(k) \end{bmatrix} + \begin{bmatrix} e_i(k) \\ e_i(k) \\ e_i(k) \\ \vdots \\ e_i(k) \end{bmatrix}$$

$$y(k|k) = y_0(k|k-1) + t_0 d_i(k) + e_i(k)$$

Define the following signal vector:

$$\tilde{y}_0(k|k) = \begin{bmatrix} y_0(k|k) \\ y_0(k+1|k) \\ \vdots \\ y_0(k+n-2|k) \\ y_0(k+n-1|k) \end{bmatrix}$$

and matrices

$$M = \begin{bmatrix} 0 & I & 0 & \cdots & 0 & 0 \\ 0 & 0 & I & & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & 0 & & I & 0 \\ 0 & 0 & 0 & & 0 & I \\ 0 & 0 & 0 & \cdots & 0 & I \end{bmatrix} \qquad S = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{n-1} \\ s_n \end{bmatrix} \qquad T = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_{n-1} \\ t_n \end{bmatrix}$$

$$C = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix}$$

Then equations (3.9) can be written as:

$$\tilde{y}_0(k+1|k) = M\,\tilde{y}_0(k|k-1) + S\Delta u(k) + Td_i(k) + I_n\,e_i(k) \tag{3.10}$$
$$y(k) = C\,\tilde{y}_0(k|k-1) + t_o d_i(k) + e_i(k) \tag{3.11}$$

To make predictions $y(k+j|k)$, we can extend (3.11) over the prediction horizon N. In matrix form:

$$\begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+n-1|k) \\ \hat{y}(k+n|k) \\ \vdots \\ \hat{y}(k+N|k) \end{bmatrix} = \begin{bmatrix} y_0(k+1|k-1) \\ y_0(k+2|k-1) \\ \vdots \\ y_0(k+n-1|k-1) \\ y_0(k+n-1|k-1) \\ \vdots \\ y_0(k+n-1|k-1) \end{bmatrix} + \begin{bmatrix} s_1\Delta u(k) \\ s_2\Delta u(k)+s_1\Delta u(k+1) \\ \vdots \\ s_{n-1}\Delta u(k)+\cdots+s_1\Delta u(k+n-2) \\ s_n\Delta u(k)+\cdots+s_1\Delta u(k+n-1) \\ \vdots \\ s_N\Delta u(k)+\cdots+s_1\Delta u(k+N-1) \end{bmatrix} +$$

$$+ \begin{bmatrix} t_1 d_i(k) \\ t_2 d_i(k)+t_1 d_i(k+1) \\ \vdots \\ t_{n-1} d_i(k)+\cdots+t_1 d_i(k+n-2) \\ t_n d_i(k)+\cdots+t_1 d_i(k+n-1) \\ \vdots \\ t_N d_i(k)+\cdots+t_1 d_i(k+N-1) \end{bmatrix} + \begin{bmatrix} e_i(k) \\ e_i(k) \\ \vdots \\ e_i(k) \\ e_i(k) \\ \vdots \\ e_i(k) \end{bmatrix}$$

Define the following signal vectors:

$$\tilde{z}(k)=\begin{bmatrix}\hat{y}(k+1|k)\\ \hat{y}(k+|k)\\ \vdots\\ \hat{y}(k+N-1|k)\\ \hat{y}(k+N|k)\end{bmatrix}\qquad \tilde{u}(k)=\begin{bmatrix}\Delta u(k)\\ \Delta u(k+1)\\ \vdots\\ \Delta u(k+N-2)\\ \Delta u(k+N-1)\end{bmatrix}\qquad \tilde{w}(k)=\begin{bmatrix}d_i(k)\\ d_i(k+1)\\ \vdots\\ d_i(k+N-2)\\ d_i(k+N-1)\end{bmatrix}$$

and matrices

$$\tilde{M}=\begin{bmatrix}0 & I & 0 & \cdots & 0 & 0\\ 0 & 0 & I & & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & & \vdots\\ 0 & 0 & 0 & & I & 0\\ 0 & 0 & 0 & & 0 & I\\ \vdots & \vdots & \vdots & & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & 0 & I\end{bmatrix}\qquad \tilde{S}=\begin{bmatrix}s_1 & 0 & \vdots & 0\\ s_2 & s_1 & & 0\\ \vdots & & \ddots & \vdots\\ s_N & s_{N-1} & & s_1\end{bmatrix}$$

$$\tilde{T}=\begin{bmatrix}t_1 & 0 & \vdots & 0\\ t_2 & t_1 & & 0\\ \vdots & & \ddots & \vdots\\ t_N & t_{N-1} & & t_1\end{bmatrix}$$

(Note that $s_m = s_n$ and $t_m = t_n$ for $m > n$.)
Now the prediction model can be written as:

$$\tilde{z}(k)=\tilde{M}\tilde{y}_0(k)+\tilde{S}\tilde{u}(k)+\tilde{T}\tilde{w}(k)+I_N\,e(k) \tag{3.12}$$

Relation to state space model

The model derived in equations (3.10), (3.11) and (3.12) are simular to the state space model derived in the previous section. By choosing

$$x(k)=\tilde{y}_0(k)\quad z(k)=y(k)\quad e(k)=e_i(k)\quad w(k)=d_i(k)\quad v(k)=\Delta u(k)$$

$$A=M\qquad B_1=I_n\qquad B_2=T\qquad B_3=S$$

$$C_1=C\qquad D_{11}=I\qquad D_{12}=t_0$$

$$\tilde{C}_2=\tilde{M}\qquad \tilde{D}_{21}=I_N\qquad \tilde{D}_{22}=\tilde{T}\qquad \tilde{D}_{23}=\tilde{S}$$

we have obtain the following prediction model:

$$\begin{aligned}x(k+1)&=A\,x(k)+B_1e(k)+B_2w(k)+B_3v(k)\\ y(k)&=C_1\,x(k)+D_{11}\,e(k)+D_{12}\,w(k)\\ \tilde{z}(k)&=\tilde{C}_2\,x(k)+\tilde{D}_{21}e(k)+\tilde{D}_{22}\tilde{w}(k)+\tilde{D}_{23}v(k)\end{aligned}$$

3.4 Prediction using polynomial models

Diophantine equation and making predictions

Consider the polynomial model

$$u_1(k) = \frac{c(q)}{a(q)} u_2(k)$$

where

$$\begin{aligned} a(q) &= 1 + a_1\, q^{-1} + \ldots + a_{n_a}\, q^{-n_a} \\ c(q) &= c_0 + c_1\, q^{-1} + \ldots + c_{n_c}\, q^{-n_c} \end{aligned}$$

Each set of polynomial functions $a(q)$, $c(q)$ together with a positive integer j satifies a set of Diophantine equations

$$c(q) = M_j(q)\, a(q) + q^{-j} L_j(q) \tag{3.13}$$

where

$$\begin{aligned} M_j(q) &= m_{j,0} + m_{j,1}\, q^{-1} + \ldots + m_{j,j-1}\, q^{-j+1} \\ L_j(q) &= \ell_{j,0} + \ell_{j,1}\, q^{-1} + \ldots + \ell_{j,n_a}\, q^{-n_a} \end{aligned}$$

Remark:
Diophantine equations are standard in the theory of prediction with polynomial models. An algorithm for solving these equations is given in appendix B of Soeterboek [108].

When making a prediction of $u_1(k+j)$ for $j > 0$ we are interested how $u_1(k+j)$ depends on future values of $u_2(k+i)$, $0 < i \leq j$ and present and past values of $u_2(k-i)$, $i \geq 0$. Using Diophantine equation (3.13) we derive (Clarke *et al.*[21][22], Bitmead *et al.*[12]):

$$\begin{aligned} u_1(k+j) &= \frac{c(q)}{a(q)} u_2(k+j) \\ &= M_j(q) u_2(k+j) + q^{-j} \frac{L_j(q)}{a(q)} u_2(k+j) \\ &= M_j(q) u_2(k+j) + \frac{L_j(q)}{a(q)} u_2(k) \end{aligned}$$

This means that $u_1(k+j)$ can be written as

$$u_1(k+j) = u_{1,future}(k+j) + u_{1,past}(k+j)$$

where the factor

$$u_{1,future}(k+j) = M_j(q) u_2(k+j) = m_{j,0} u_2(k+j) + \ldots + m_{j,j-1}\, u_2(k+1)$$

only consist of future values of $u_2(k+i)$, $1 \leq i \leq j$, and the second term

$$u_{1,past}(k+j) = \frac{L_j(q)}{a(q)} u_2(k)$$

only present and past values of $u_2(k-i)$, $i \geq 0$.

Prediction using polynomial models

Let model (3.1) be given in polynomial form as:

$$\begin{aligned} a(q)\,y(k) &= c(q)\,e(k) + f_1(q)\,w_1(k) + f_2(q)\,w_2(k) + b(q)\,v(k) \qquad (3.14) \\ n(q)\,z(k) &= h(q)\,e(k) + m_1(q)\,w_1(k) + m_2(q)\,w_2(k) + g(q)\,v(k) \qquad (3.15) \end{aligned}$$

where w_1 and w_2 usually corresponds to the reference signal $r(k)$ and the known distrubance signal $d(k)$, respectively. In the sequel of this section we will use

$$\begin{aligned} f(q)w(q) &= \begin{bmatrix} f_1(q) & f_2(q) \end{bmatrix} \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix} \qquad (3.16) \\ m(q)w(q) &= \begin{bmatrix} m_1(q) & m_2(q) \end{bmatrix} \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix} \qquad (3.17) \end{aligned}$$

as an easier notation.

To make predictions of future output signals of CARIMA models one first needs to solve the following Diophantine equation

$$h(q) = E_j(q)\,n(q) + q^{-j}F_j(q) \qquad (3.18)$$

Using (3.15) and (3.18), we derive

$$\begin{aligned} z(k+j) &= \frac{h(q)}{n(q)}\,e(k+j) + \frac{m(q)}{n(q)}\,w(k+j) + \frac{g(q)}{n(q)}\,v(k+j) \\ &= E_j(q)\,e(k+j) + \frac{q^{-j}F_j(q)}{n(q)}\,e(k+j) + \frac{m(q)}{n(q)}\,w(k+j) + \frac{g(q)}{n(q)}\,v(k+j) \\ &= E_j(q)\,e(k+j) + \frac{F_j(q)}{n(q)}\,e(k) + \frac{m(q)}{n(q)}\,w(k+j) + \frac{g(q)}{n(q)}\,v(k+j) \qquad (3.19) \end{aligned}$$

From (3.14) we derive

$$e(k) = \frac{a(q)}{c(q)}\,y(k) - \frac{f(q)}{c(q)}\,w(k) - \frac{b(q)}{c(q)}\,v(k)$$

Substitution in (3.19) yields:

$$\begin{aligned} z(k+j) &= E_j(q)\,e(k+j) + \frac{F_j(q)}{n(q)}\,e(k) + \frac{m(q)}{n(q)}\,w(k+j) + \frac{g(q)}{n(q)}\,v(k+j) \\ &= E_j(q)\,e(k+j) + \frac{F_j(q)a(q)}{n(q)c(q)}\,y(k) - \frac{F_j(q)f(q)}{n(q)c(q)}\,w(k) - \frac{F_j(q)b(q)}{n(q)c(q)}\,v(k) + \\ &\quad + \frac{m(q)}{n(q)}\,w(k+j) + \frac{g(q)}{n(q)}\,v(k+j) \\ &= E_j(q)\,e(k+j) + \frac{F_j(q)a(q)}{n(q)c(q)}\,y(k) + \left(\frac{m(q)}{n(q)} - \frac{q^{-j}F_j(q)f(q)}{n(q)c(q)}\right) w(k+j) + \end{aligned}$$

$$
\begin{aligned}
&+\left(\frac{g(q)}{n(q)} - \frac{q^{-j}F_j(q)b(q)}{n(q)c(q)}\right) v(k+j) \\
&= E_j(q)\, e(k+j) + \frac{F_j(q)a(q)}{n(q)c(q)}\, y(k) + \frac{\bar{m}(q)}{n(q)c(q)}\, w(k+j) + \frac{\bar{g}(q)}{n(q)c(q)}\, v(k+j)
\end{aligned}
$$

where

$$
\begin{aligned}
\bar{m}(q) &= m(q)c(q) - q^{-j}F_j(q)f(q) \\
\bar{g}(q) &= g(q)c(q) - q^{-j}F_j(q)b(q)
\end{aligned}
$$

Finally solve the Diophantine equation:

$$
q\bar{g}(q) = \Phi_j(q)\, n(q)\, c(q) + q^{-j}\Gamma_j(q) \tag{3.20}
$$

Then:

$$
\begin{aligned}
z(k+j) &= E_j(q)\, e(k+j) + \frac{F_j(q)a(q)}{n(q)c(q)}\, y(k) + \frac{\bar{m}(q)}{n(q)c(q)}\, w(k+j) + \frac{\bar{g}(q)}{n(q)c(q)}\, v(k+j) \\
&= E_j(q)\, e(k+j) + \frac{F_j(q)a(q)}{n(q)c(q)}\, y(k) + \frac{\bar{m}(q)}{n(q)c(q)}\, w(k+j) + \\
&\quad + \frac{q^{-1}\Gamma(q)}{n(q)c(q)}\, v(k) + q^{-1}\Phi_j(q)\, v(k+j) \\
&= E_j(q)\, e(k+j) + \frac{F_j(q)a(q)}{n(q)c(q)}\, y(k) + \frac{\bar{m}(q)}{n(q)c(q)}\, w(k+j) + \\
&\quad + \frac{\Gamma(q)}{n(q)c(q)}\, v(k-1) + \Phi_j(q)\, v(k+j-1)
\end{aligned}
$$

Using the fact that the prediction $\hat{e}(k+j|k) = 0$ for $j > 0$ the first term vanishes and the optimal prediction $\hat{z}(k+j)$ for $j > 0$ is given by:

$$
\begin{aligned}
\hat{z}(k+j) &= \frac{F_j(q)a(q)}{n(q)c(q)}\, y(k) + \left(\frac{\bar{m}(q)}{n(q)c(q)}\right) w(k+j) + \\
&\quad + \left(\frac{\Gamma(q)}{n(q)c(q)}\right) v(k-1) + \Phi_j(q)\, v(k+j-1) \\
&= F_j(q)a(q)\, y^f(k) + \bar{m}(q)\, w^f(k+j) + \Gamma(q)\, v^f(k-1) + \Phi_j(q)\, v(k+j-1)
\end{aligned}
$$

where y^f, w^f and v^f are the filtered signals

$$
y^f(k) = \frac{1}{n(q)c(q)}\, y(k)
$$

$$
w^f(k) = \frac{1}{n(q)c(q)}\, w(k)
$$

$$
v^f(k) = \frac{1}{n(q)c(q)}\, v(k)
$$

The free-run response is now given by $F_j(q)a(q)\,y^f(k) + \bar{m}(q)\,w^f(k+j) + \Gamma_j(q)\,v^f(k-1)$. which is the output response if all future input signals are taken zero. The last term $\Phi_j\,v(k+j-1)$ accounts for the influence of the future input signals $v(k),\ldots,v(k+j-1)$.

The prediction of the performance signal can now be given as:

$$
\begin{aligned}
\tilde{z}(k) &= \begin{bmatrix} F_0(q)\,y^f(k) + \bar{m}_0(q)\,w^f(k) + \Gamma_0(q)\,v^f(k{-}1) + \Phi_0(q)\,v(k) \\ F_1(q)\,y^f(k) + \bar{m}_1(q)\,w^f(k{+}1) + \Gamma_1(q)\,v^f(k{-}1) + \Phi_1(q)\,v(k{+}1) \\ \vdots \\ F_{N-1}(q)\,y^f(k) + \bar{m}_{N-1}(q)\,w^f(k{+}N{-}1) + \Gamma_{N-1}(q)\,v^f(k{-}1) + \Phi_{N-1}(q)\,v(k{+}N{-}1) \end{bmatrix} \\
&= \tilde{z}_0(k) + \tilde{D}_{vz}\,\tilde{v}(k)
\end{aligned}
$$

The free response vector $\tilde{z}_0$ and the predictor matrix $\tilde{D}_{vz}$ will be equal to $\tilde{z}_0$ and $\tilde{D}_3$, respectively, in the case of a state-space model.

Relation to state-space model:

Consider the above system $a(q)\,u_1(k+j) = c(q)\,u_2(k+j)$ in state-space representation, where A, B, C, D are given by the controller canonical form:

$$
A = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & & 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}
$$

$$
C = \begin{bmatrix} c_1 - c_o a_1 & c_2 - c_o a_2 & \ldots & c_n - c_o a_n \end{bmatrix} \qquad D = c_o
$$

Then based on the previous sections we know that the prediction of $u_1(k+j)$ is given by:

$$
u_1(k+j) = CA^j x(k) + \sum_{i=1}^{j} CA^{i-1} B u_2(k+j-i) + D u_2(k+j)
$$

For this description we will find that:

$$
\begin{aligned}
CA^j &= \begin{bmatrix} \ell_{j,0} & \ell_{j,1} & \cdots & \ell_{j,n_a} \end{bmatrix} \\
CA^{i-1}B &= m_{j,j-i} \quad \text{for} \quad 0 < i \leq j \\
D &= m_{j,0}
\end{aligned}
$$

and so where

$$
CA^j x(k) = \frac{L_j(q)}{a(q)} u_2(k)
$$

reflects the past of the system, the equation

$$\begin{aligned} Du_2(k+j) + \sum_{i=1}^{j} CA^{i-1}Bu_2(k+j-i) &= m_{j,0}u_2(k+j) + \ldots + m_{j,j-1}\,u_2(k+1) \\ &= M_j(q)u_2(k+j) \end{aligned}$$

gives the response of the future inputs $u_2(k+i)$, $i > 0$.

Chapter 4

Standard formulation

In this chapter we will standardise the predictive control setting and formulate the standard predictive control problem. First we consider the performance index, secondly the constraints.

4.1 The performance index

In predictive control various types of performance indices are presented, in which either the input signal is measured together with the state (LQPC), or the input increment signal is measured together with the tracking error (GPC). This leads to two different types of performance indices:

The LQPC performance index:

$$J(u,k) = \sum_{j=N_m}^{N} \hat{x}_o^T(k+j|k) Q \hat{x}_o(k+j|k) + \sum_{j=1}^{N} u^T(k+j-1|k) R u(k+j-1|k) \quad (4.1)$$

where x_o is the state of the IO-model

$$\begin{aligned} x_o(k+1) &= A_o x_o(k) + K_o e_o(k) + L_o d_o(k) + B_o u(k) && (4.2) \\ y(k) &= C_o x_o(k) + D_H e_o(k) + D_F d_o(k) && (4.3) \end{aligned}$$

and $N \geq N_m \geq 1$ and $N \geq N_c \geq 1$, Q and R are positive semi-definite.

The GPC performance index:

$$J(u,k) = \sum_{j=N_m}^{N} \Big(\hat{y}_p(k+j|k) - r(k+j)\Big)^T \Big(\hat{y}_p(k+j|k) - r(k+j)\Big) +$$

$$+\lambda^2 \sum_{j=1}^{N} \Delta u^T(k+j-1|k) \Delta u(k+j-1|k) \quad (4.4)$$

where $N \geq N_m \geq 1$ and $N \geq N_c \geq 1$ and $\lambda \in \mathbb{R}$. The output $y(k)$ and increment input $\Delta u(k)$ are related by the IIO model

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + K_i e_i(k) + L_i d_i(k) + B_i \Delta u(k) && (4.5)\\ y(k) &= C_i x_i(k) + D_H e_i(k) && (4.6)\end{aligned}$$

and the weighted output $y_p(k) = P(q)y(k)$ is given in state space representation

$$\begin{aligned} x_p(k+1) &= A_p\, x_p(k) + B_p\, y(k) && (4.7)\\ \phi(k) &= C_p\, x_p(k) + D_p\, y(k) && (4.8)\end{aligned}$$

(Note that for GPC we choose $D_F = 0$)

The zone performance index:

$$J(u,k) = \sum_{j=N_m}^{N} \hat{\epsilon}^T(k+j|k)\hat{\epsilon}(k+j|k) + \sum_{j=1}^{N} \lambda^2 u^T(k+j-1|k)u(k+j-1|k) \qquad (4.9)$$

where $\epsilon_i(k)$, $i = 1, \ldots, m$ contribute to the performance index only if $|y_i(k) - r_i(k)| > \delta_{max,i}$:

$$\epsilon_i(k) = \begin{cases} 0 & \text{for} \quad |y_i(k) - r_i(k)| \leq \delta_{max,i} \\ y_i(k) - r_i(k) - \delta_{max,i} & \text{for} \quad y_i(k) - r_i(k) \geq \delta_{max,i} \\ y_i(k) - r_i(k) + \delta_{max,i} & \text{for} \quad y_i(k) - r_i(k) \leq -\delta_{max,i} \end{cases}$$

so

$$|\epsilon_i(k)| = \min_{|\delta_i(k)| \leq \delta_{max,i}} |y_i(k) - r_i(k) + \delta_i(k)|$$

There holds $N \geq N_m \geq 1$ and $\lambda \in \mathbb{R}$. The output $y(k)$ and input $u(k)$ are related by the IO model

$$\begin{aligned} x_o(k+1) &= A_o x_o(k) + K_o e_o(k) + L_o d_o(k) + B_o u(k) && (4.10)\\ y(k) &= C_o x_o(k) + D_H e_o(k) + D_F d_o(k) && (4.11)\end{aligned}$$

Both the LQPC performance index and the GPC performance index are frequently used in practice and in literature. The LQPC performance index resembles the LQ optimal control performance index. The choices of the matrices Q and R can be done in a similar way as in LQ optimal control. The GPC performance index originates from the adaptive control community and is based on SISO polynomial models.

The standard predictive control performance index:

In this course we will use the state space setting of LQPC as well as the use of a reference signal as is done in GPC. Instead of the state x or the output y we will use a general signal z in the performance index. We therefore adopt the standard predictive control performance index:

$$J(v,k) = \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \qquad (4.12)$$

where $\hat{z}(k+j|k)$ is the prediction of $z(k+j)$ at time k and $\Gamma(j)$ is a diagonal selection matrix with ones and zeros on the diagonal. Finally, we use a generalized input signal $v(k)$, which can either be the input signal $u(k)$ or the input increment signal $\Delta u(k)$ and we use a signal $w(k)$, which contains all known external signals, such as the reference signal $r(k)$ and the known disturbance signal $d(k)$. The state signal x, input signal v, output signal y, external signal w and performance signal z are related by the following equations:

$$\begin{aligned} x(k+1) &= A\,x(k) + B_1\,e(k) + B_2\,w(k) + B_3\,v(k) && (4.13)\\ y(k) &= C_1\,x(k) + D_{11}\,e(k) + D_{12}\,w(k) && (4.14)\\ z(k) &= C_2\,x(k) + D_{21}\,e(k) + D_{22}\,w(k) + D_{23}\,v(k) && (4.15) \end{aligned}$$

Theorem 5 Transformation of LQPC into standard form
Given the MPC problem with LQPC performance index (4.1) for IO model (4.2),(4.3). This LQPC problem can be translated into a standard predictive control problem with performance index (4.12) and model (4.13),(4.14),(4.15) by the following substutions:

$$x(k) = x_o(k) \quad v(k) = u(k) \quad w(k) = d(k) \quad e(k) = e_o(k)$$

$$A = A_o \qquad B_1 = K_o \qquad B_2 = L_o \qquad B_3 = B_o$$

$$C_1 = C_o \qquad D_{11} = D_H \qquad D_{12} = D_F$$

$$C_2 = \begin{bmatrix} Q^{1/2}A_o \\ 0 \end{bmatrix} \qquad D_{21} = \begin{bmatrix} Q^{1/2}K_o \\ 0 \end{bmatrix} \qquad D_{22} = \begin{bmatrix} Q^{1/2}L_o \\ 0 \end{bmatrix} \qquad D_{23} = \begin{bmatrix} Q^{1/2}B_o \\ R^{1/2} \end{bmatrix}$$

$$\Gamma(j) = \begin{bmatrix} \Gamma_1(j) & 0 \\ 0 & I \end{bmatrix}$$

$$\Gamma_1(j) = \begin{cases} 0 & \textit{for} \quad 0 \le j < N_m - 1 \\ I & \textit{for} \quad N_m - 1 \le j \le N - 1 \end{cases}$$

proof:

First choose the performance signal

$$z(k) = \begin{bmatrix} Q^{1/2}x_o(k+1) \\ R^{1/2}u(k) \end{bmatrix} \tag{4.16}$$

Then it easy to see that

$$J(v,k) = \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k)$$

$$
\begin{aligned}
&= \sum_{j=0}^{N-1} \hat{x}^T(k+1+j|k) Q^{1/2} \Gamma_1(j) Q^{1/2} \hat{x}(k+1+j|k) + \\
&\quad \sum_{j=0}^{N-1} u^T(k+j|k) R^{1/2} I R^{1/2} u(k+j|k) \\
&= \sum_{j=1}^{N} \hat{x}^T(k+j|k) Q^{1/2} \Gamma_1(j-1) Q^{1/2} \hat{x}(k+j|k) + \\
&\quad \sum_{j=1}^{N} u^T(k+j-1|k) R u(k+j-1|k) \\
&= \sum_{j=N_m}^{N} \hat{x}^T(k+j|k) Q \hat{x}(k+j|k) + \sum_{j=1}^{N} u^T(k+j-1|k) R u(k+j-1|k)
\end{aligned}
$$

and so performance index (4.12) is equivalent to (4.1) for the above given z, N and Γ. From the IO state space equation

$$x_o(k+1) = A_o\, x_o(k) + B_o\, u(k) + L_o\, d(k) + K_o\, e_o(k)$$

and using the definitions of x, e, v and w, we find:

$$
\begin{aligned}
x(k+1) &= A\, x(k) + B_1\, e(k) + B_2\, d(k) + B_3\, v(k) \\
z(k) &= \begin{bmatrix} Q^{1/2} x_o(k+1) \\ R^{1/2} u(k) \end{bmatrix} \\
&= \begin{bmatrix} Q^{1/2} \Big(A_o\, x_o(k) + B_o\, u(k) + L_o\, d(k) + K_o\, e_o(k) \Big) \\ R^{1/2} u(k) \end{bmatrix} \\
&= \begin{bmatrix} Q^{1/2} A_o \\ 0 \end{bmatrix} x_o(k) + \begin{bmatrix} Q^{1/2} K_o \\ 0 \end{bmatrix} e_o(k) + \begin{bmatrix} Q^{1/2} L_o \\ 0 \end{bmatrix} d_o(k) + \begin{bmatrix} Q^{1/2} B_o \\ R^{1/2} \end{bmatrix} u(k) \\
&= C_2\, x(k) + D_{21}\, e(k) + D_{22}\, w(k) + D_{23}\, v(k)
\end{aligned}
$$

for the given choices of A, B_1, B_2, B_3, C_1, C_2, D_{21}, D_{22} and D_{23}. □ End Proof

Theorem 6 Transformation of GPC into standard form
Given the MPC problem with GPC performance index (4.4) for IIO model (4.5),(4.6) and the weighting filter $P(q)$, given by (4.7),(4.8). This GPC problem can be translated into a standard predictive control problem with performance index (4.12) and model (4.13),(4.14),(4.15) by the following substutions:

$$x(k) = \begin{bmatrix} x_p(k) \\ x_i(k) \end{bmatrix}, \quad v(k) = \Delta u(k), \quad w(k) = \begin{bmatrix} d_i(k) \\ r(k+1) \end{bmatrix} \quad e(k) = e_i(k)$$

$$A = \begin{bmatrix} A_p & B_p C_i \\ 0 & A_i \end{bmatrix} \quad B_1 = \begin{bmatrix} B_p D_H \\ K_i \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 & 0 \\ L_i & 0 \end{bmatrix} \quad B_3 = \begin{bmatrix} 0 \\ B_i \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & C_i \end{bmatrix} \qquad D_{11} = D_H \qquad D_{12} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} C_pA_p & C_pB_pC_i + D_pC_iA_i \\ 0 & 0 \end{bmatrix} \qquad D_{21} = \begin{bmatrix} C_pB_pD_H + D_pC_iK_i \\ 0 \end{bmatrix}$$

$$D_{22} = \begin{bmatrix} D_pC_iL_i & -I \\ 0 & 0 \end{bmatrix} \qquad D_{23} = \begin{bmatrix} D_pC_iB_i \\ \lambda I \end{bmatrix}$$

$$\Gamma(j) = \begin{bmatrix} \Gamma_1(j) & 0 \\ 0 & I \end{bmatrix}$$

$$\Gamma_1(j) = \begin{cases} 0 & \mathit{for} \quad 0 \le j < N_m - 1 \\ I & \mathit{for} \quad N_m - 1 \le j \le N - 1 \end{cases}$$

proof:

The GPC performance index as given in (4.4) can be transformed into the standard performance index (4.12). First choose the performance signal

$$z(k) = \begin{bmatrix} \hat{y}_p(k+1|k) - r(k+1) \\ \lambda \Delta u(k) \end{bmatrix} \tag{4.17}$$

Then it easy to see that

$$\begin{aligned}
J(v,k) &= \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) = \\
&= \sum_{j=0}^{N-1} \begin{bmatrix} \hat{y}_p^T(k+1+j|k) - r^T(k+1+j) & \lambda \Delta u^T(k+j|k) \end{bmatrix} \\
&\qquad\qquad \times \Gamma(j) \begin{bmatrix} \hat{y}_p(k+1+j|k) - r(k+1+j|k) \\ \lambda \Delta u(k+j|k) \end{bmatrix} = \\
&= \sum_{j=0}^{N-1} \Big(\hat{y}_p(k+1+j|k) - r(k+1+j)\Big)^T \Gamma_1(j) \Big(\hat{y}_p(k+1+j|k) - r(k+1+j)\Big) \\
&\quad + \lambda^2 \sum_{j=0}^{N-1} \Delta u^T(k+j|k) \Delta u(k+j|k) \\
&= \sum_{j=N_m}^{N} \Big(\hat{y}_p(k+j|k) - r(k+j)\Big)^T \Big(\hat{y}_p(k+j|k) - r(k+j)\Big) \\
&\quad + \lambda^2 \sum_{j=1}^{N} \Delta u^T(k+j-1|k) \Delta u(k+j-1|k)
\end{aligned}$$

and so performance index (4.12) is equivalent to (4.4) for the above given $z(k)$, N and $\Gamma(j)$. Substitution of the IIO-output equation

$$y(k) = C_i\, x_i(k) + D_H e_i(k)$$

in (4.7),(4.8) results in:

$$\begin{aligned}
x_p(k+1) &= A_p\, x_p(k) + B_p\, C_i\, x_i(k) + B_p\, D_H\, e_i(k) \\
y_p(k) &= C_p\, x_p(k) + D_p\, C_i\, x_i(k) + D_p\, D_H\, e_i(k)
\end{aligned}$$

and so

$$\begin{aligned}
x(k+1) &= \begin{bmatrix} x_p(k+1) \\ x_i(k+1) \end{bmatrix} = \\
&= \begin{bmatrix} A_p & B_pC_i \\ 0 & A_i \end{bmatrix} \begin{bmatrix} x_p(k) \\ x_i(k) \end{bmatrix} + \begin{bmatrix} B_p\, D_H \\ K_i \end{bmatrix} e_i(k) + \\
&\quad \begin{bmatrix} 0 & 0 \\ L_i & 0 \end{bmatrix} \begin{bmatrix} d_i(k) \\ r(k+1) \end{bmatrix} + \begin{bmatrix} 0 \\ B_i \end{bmatrix} v(k) = \\
&= A\, x(k) + B_1\, e(k) + B_2\, w(k) + B_3\, v(k) \\
y(k) &= C_1\, x(k) + D_H\, e_i(k) + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} d_i(k) \\ r(k+1) \end{bmatrix} \\
&= C_1\, x(k) + D_{11}\, e_i(k) + D_{12}\, w(k) \\
y_p(k) &= \begin{bmatrix} C_p & D_pC_i \end{bmatrix} x(k) + D_p\, D_H\, e_i(k) + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} d_i(k) \\ r(k+1) \end{bmatrix}
\end{aligned}$$

for the given choices of A, B_1, B_2, B_3 and C_1. For $\hat{y}_p(k+1|k)$ we derive:

$$\begin{aligned}
\hat{y}_p(k+1|k) &= \begin{bmatrix} C_p & D_pC_i \end{bmatrix} x(k+1) + D_pD_H\, \hat{e}_i(k+1) \\
&= \begin{bmatrix} C_p & D_pC_i \end{bmatrix} \begin{bmatrix} A_p & B_pC_i \\ 0 & A_i \end{bmatrix} \begin{bmatrix} x_p(k) \\ x_i(k) \end{bmatrix} + \begin{bmatrix} C_p & D_pC_i \end{bmatrix} \begin{bmatrix} B_pD_H \\ K_i \end{bmatrix} \hat{e}_i(k) + \\
&\quad \begin{bmatrix} C_p & D_pC_i \end{bmatrix} \begin{bmatrix} 0 & 0 \\ L_i & 0 \end{bmatrix} \begin{bmatrix} d_i(k) \\ r(k+1) \end{bmatrix} + \begin{bmatrix} C_p & D_pC_i \end{bmatrix} \begin{bmatrix} 0 \\ B_i \end{bmatrix} v(k) + \\
&\quad D_pD_H\, \hat{e}_i(k+1) = \\
&= \begin{bmatrix} C_pA_p & C_pB_pC_i + D_pC_iA_i \end{bmatrix} \begin{bmatrix} x_p(k) \\ x_i(k) \end{bmatrix} + \begin{bmatrix} C_pB_pD_H + D_pC_iK_i \end{bmatrix} \hat{e}_i(k) + \\
&\quad \begin{bmatrix} D_pC_iL_i & 0 \end{bmatrix} \begin{bmatrix} d_i(k) \\ r(k+1) \end{bmatrix} + \begin{bmatrix} D_pC_iB_i \end{bmatrix} v(k)
\end{aligned}$$

where we used the estimate $\hat{e}_i(k+1) = 0$.
For $\hat{z}(k)$ we derive:

$$\begin{aligned}
\hat{z}(k) &= \begin{bmatrix} \hat{y}_p(k+1|k) - r(k+1) \\ \lambda \Delta u(k) \end{bmatrix} \\
&= \begin{bmatrix} \hat{y}_p(k+1|k) \\ 0 \end{bmatrix} + \begin{bmatrix} -r(k+1) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \lambda v(k) \end{bmatrix} \\
&= \begin{bmatrix} C_pA_p & C_pB_pC_i + D_pC_iA_i \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_p(k) \\ x_i(k) \end{bmatrix} + \begin{bmatrix} C_pB_pD_H + D_pC_iK_i \\ 0 \end{bmatrix} \hat{e}_i(k) +
\end{aligned}$$

$$
\begin{aligned}
& \left[\begin{array}{cc} D_pC_iL_i & 0 \\ 0 & 0 \end{array}\right]\left[\begin{array}{c} d_i(k) \\ r(k{+}1) \end{array}\right] + \left[\begin{array}{c} D_pC_iB_i \\ 0 \end{array}\right] v(k) + \\
& \left[\begin{array}{c} -I \\ 0 \end{array}\right] r(k{+}1) + \left[\begin{array}{c} 0 \\ \lambda I \end{array}\right] v(k) \\
= & \left[\begin{array}{cc} C_pA_p & C_pB_pC_i + D_pC_iA_i \\ 0 & 0 \end{array}\right]\left[\begin{array}{c} x_p(k) \\ x_i(k) \end{array}\right] + \left[\begin{array}{c} C_pB_pD_H + D_pC_iK_i \\ 0 \end{array}\right] \hat{e}_i(k) + \\
& \left[\begin{array}{cc} D_pC_iL_i & -I \\ 0 & 0 \end{array}\right] w(k) + \left[\begin{array}{c} D_pC_iB_i \\ \lambda I \end{array}\right] v(k) \\
= & \; C_2\, x(k) + D_{21}\, e(k) + D_{22}\, w(k) + D_{23}\, v(k)
\end{aligned}
$$

for the given choices of C_2, D_{21}, D_{22} and D_{23}, results in a performance index (4.12) equivalent to (4.1). □ End
Proof

Theorem 7 Transformation of Zone control into standard form
Given the MPC problem with zone performance index (4.9) for IO model (4.10),(4.11). This GPC problem can be translated into a standard predictive control problem with performance index (4.12) and model (4.13),(4.14),(4.15) by the following substutions:

$$
z(k) = \left[\begin{array}{c} \hat{y}(k+1|k) - r(k+1) + \delta(k) \\ \lambda u(k) \end{array}\right] \qquad v(k) = \left[\begin{array}{c} u(k) \\ \delta(k) \end{array}\right]
$$

$$
w(k) = \left[\begin{array}{c} d_o(k) \\ r(k+1) \end{array}\right] \qquad e(k) = e_o(k)
$$

$$
A = \left[\begin{array}{c} A_o \end{array}\right] \quad B_1 = \left[\begin{array}{c} K_o \end{array}\right] \quad B_2 = \left[\begin{array}{cc} L_o & 0 \end{array}\right] \quad B_3 = \left[\begin{array}{cc} B_o & 0 \end{array}\right]
$$

$$
C_1 = C_o \quad D_{11} = D_H \quad D_{12} = \left[\begin{array}{cc} D_F & 0 \end{array}\right]
$$

$$
C_2 = \left[\begin{array}{c} C_oA_o \\ 0 \end{array}\right] \quad D_{21} = \left[\begin{array}{c} C_oK_o \\ 0 \end{array}\right] \quad D_{22} = \left[\begin{array}{cc} C_oL_o & -I \\ 0 & 0 \end{array}\right] \quad D_{23} = \left[\begin{array}{cc} C_oB_o & I \\ \lambda I & 0 \end{array}\right]
$$

$$
C_4 = \left[\begin{array}{c} 0 \\ 0 \end{array}\right] \quad D_{41} = \left[\begin{array}{c} 0 \\ 0 \end{array}\right] \quad D_{42} = \left[\begin{array}{c} 0 \\ 0 \end{array}\right] \quad D_{43} = \left[\begin{array}{cc} 0 & I \\ 0 & -I \end{array}\right] \quad \tilde{\Psi} = \left[\begin{array}{c} \delta_{max} \\ \delta_{max} \end{array}\right]
$$

then:

$$
\begin{aligned}
x(k+1) &= A\, x(k) + B_1\, e(k) + B_2\, w(k) + B_3\, v(k) \\
y(k) &= C_1\, x(k) + D_{11}\, e(k) + D_{12}\, w(k) \\
z(k) &= C_2\, x(k) + D_{21}\, e(k) + D_{22}\, w(k) + D_{23}\, v(k) \\
\psi(k) &= C_4\, x(k) + D_{41}\, e(k) + D_{42}\, w(k) + D_{43}\, v(k)
\end{aligned}
\tag{4.18}
$$

The proof is left to the reader.

4.2 Handling constraints

As was already discussed in chapter 1, in practical situations there will be signal constraints. These constraints on control, state and output signals are motivated by safety and environmental demands, economical perspectives and equipment limitations. There are two types of constraints:

Equality constraints:
Equality constraints are usually motivated by the control algorithm. For example, to decrease the degrees of freedom, we introduce the control horizon N_c, and to guarantee stability we can add a state end-point constraint to the problem. In this course we look at equality constraints in a generalized form:

$$\tilde{\phi}(k) = \tilde{C}_3 x(k) + \tilde{D}_{31} e(k) + \tilde{D}_{32} \tilde{w}(k) + \tilde{D}_{33} \tilde{v}(k) = 0 \tag{4.19}$$

Inequality constraints:
Inequality constraints are due to physical, economical or safety limitations. In this course we look at inequality constraints in a generalized form:

$$\tilde{\psi}(k) = \tilde{C}_4 x(k) + \tilde{D}_{41} e(k) + \tilde{D}_{42} \tilde{w}(k) + \tilde{D}_{43} \tilde{v}(k) \leq \tilde{\Psi}(k) \tag{4.20}$$

where $\tilde{\Psi}(k)$ does not depend on future input signals.

In the above formulas $\tilde{\phi}(k)$ and $\tilde{\psi}(k)$ are vectors and the equalities and inequalities are meant element-wise. If there are multiple constraints, for example $\tilde{\phi}_1(k) = 0$ and $\tilde{\phi}_2(k) = 0$, $\tilde{\psi}_1(k) =\leq \tilde{\Psi}_1(k), \ldots, \tilde{\psi}_4(k) \leq \tilde{\Psi}_4(k)$ we can combine them by both stacking them into one equality constraint and one inequality constraint.

$$\tilde{\phi}(k) = \begin{bmatrix} \tilde{\phi}_1(k) \\ \tilde{\phi}_2(k) \end{bmatrix} = 0 \qquad \tilde{\psi}(k) = \begin{bmatrix} \tilde{\psi}_1(k) \\ \vdots \\ \tilde{\psi}_4(k) \end{bmatrix} \leq \begin{bmatrix} \tilde{\Psi}_1(k) \\ \vdots \\ \tilde{\Psi}_4(k) \end{bmatrix}$$

We will show how the transformation is done for some common equality and inequality constraints.

4.2.1 Equality constraints

Control horizon constraint for IO systems

The control horizon constraint for an IO-system (with $v(k) = u(k)$) is given by

$$v(k + N_c + j|k) = v(k + N_c - 1|k) \quad \text{for} \quad j = 0, \ldots, N - N_c - 1$$

Then the control horizon constraint is equivalent with

$$
\begin{aligned}
&\begin{bmatrix} v(k+N_c|k)-v(k+N_c-1|k) \\ v(k+N_c+1|k)-v(k+N_c-1|k) \\ \vdots \\ v(k+N-1|k)-v(k+N_c-1|k) \end{bmatrix} = \\
&= \begin{bmatrix} 0 & \dots & 0 & -I & I & 0 & \dots & 0 \\ 0 & \dots & 0 & -I & 0 & I & & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & -I & 0 & 0 & \dots & I \end{bmatrix} \begin{bmatrix} v(k|k) \\ \vdots \\ v(k+N_c-2|k) \\ v(k+N_c-1|k) \\ v(k+N_c|k) \\ v(k+N_c+1|k) \\ \vdots \\ v(k+N-1|k) \end{bmatrix} \\
&= \tilde{D}_{33}\tilde{v}(k) \\
&= 0
\end{aligned}
$$

By defining $\tilde{C}_3 = 0$, $\tilde{D}_{31} = 0$ and $\tilde{D}_{32} = 0$, the control horizon constraint can be translated into the standard form of equation (4.19).

Control horizon constraint for IIO systems

The control horizon constraint for an IIO-system (with $v(k) = \Delta u(k)$) is given by

$$
v(k+N_c+j|k) = 0 \quad \text{for} \quad j = 0, \dots, N-N_c-1
$$

Then the control horizon constraint is equivalent with

$$
\begin{aligned}
\begin{bmatrix} v(k+N_c|k) \\ v(k+N_c+1|k) \\ \vdots \\ v(k+N-1|k) \end{bmatrix} &= \begin{bmatrix} 0 & \dots & 0 & I & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & I & & 0 \\ \vdots & & \vdots & \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & I \end{bmatrix} \begin{bmatrix} v(k|k) \\ \vdots \\ v(k+N_c-1|k) \\ v(k+N_c|k) \\ v(k+N_c+1|k) \\ \vdots \\ v(k+N-1|k) \end{bmatrix} \\
&= \tilde{D}_{33}\tilde{v}(k) \\
&= 0
\end{aligned}
$$

By defining $\tilde{C}_3 = 0$, $\tilde{D}_{31} = 0$, $\tilde{D}_{32} = 0$, the control horizon constraint can be translated into the standard form of equation (4.19).

State end-point constraint

The state end-point constraint is defined as

$$\hat{x}(k+N|k) = x_{ss}$$

where x_{ss} is the steady-state. The state end-point constraint guarantees stability of the closed loop (see chapter 6). In section 5.1 we will discuss steady-state behavior, and we derive that x_{ss} can be given by

$$x_{ss} = D_{ssx}\tilde{w}(k)$$

where the matrix D_{ssx} is related to the steady-state properties of the system. The state end-point constraint now becomes

$$\begin{aligned}
\hat{x}(k+N|k) - D_{ssx}\,\tilde{w}(k) = & \\
= \;& A^N x(k) + A^{N-1}B_1 e(k) + \begin{bmatrix} A^{N-1}B_2 & A^{N-2}B_2 & \cdots & B_2 \end{bmatrix}\tilde{w}(k) \\
& + \begin{bmatrix} A^{N-1}B_3 & A^{N-2}B_3 & \cdots & B_3 \end{bmatrix}\tilde{v}(k) - D_{ssx}\,\tilde{w}(k) \\
= \;& 0
\end{aligned}$$

By defining

$$\tilde{C}_3 = A^N \qquad \tilde{D}_{32} = \begin{bmatrix} A^{N-1}B_2 & A^{N-2}B_2 & \cdots & B_2 \end{bmatrix} - D_{ssx}$$

$$\tilde{D}_{31} = A^{N-1}B_1 \qquad \tilde{D}_{33} = \begin{bmatrix} A^{N-1}B_3 & A^{N-2}B_3 & \cdots & B_3 \end{bmatrix}$$

the state end-point constraint can be translated into the standard form of equation (4.19).

4.2.2 Inequality constraints

For simplicity we only consider constraints of the form

$$p(k) \le p_{max}$$

a constraint $p(k) \ge p_{min}$ is easily translated into the above form by considering

$$-p(k) \le -p_{min}$$

Consider a constraint

$$\psi(k+j) \le \Psi(k+j) \;\; , \text{ for } j = 0, \ldots, N \tag{4.21}$$

where $\psi(k)$ is given by

$$\psi(k) = C_4 x(k) + D_{41}e(k) + D_{42}w(k) + D_{43}v(k) \tag{4.22}$$

Linear inequality constraints on output $y(k)$ and state $x(k)$ can easily be translated into linear constraints on the input vector $v(k)$. By using the results of chapter 3, values of $\psi(k+j)$ can be predicted and we obtain the inequality constraint:

$$\tilde{\psi}(k) = \tilde{C}_4 x(k) + \tilde{D}_{41} e(k) + \tilde{D}_{42}\tilde{w}(k) + \tilde{D}_{43}\tilde{v}(k) \leq \tilde{\Psi}(k) \tag{4.23}$$

where

$$\tilde{\psi}(k) = \begin{bmatrix} \hat{\psi}(k) \\ \hat{\psi}(k+1) \\ \vdots \\ \hat{\psi}(k+N-1) \end{bmatrix} \qquad \tilde{\Psi}(k) = \begin{bmatrix} \Psi(k) \\ \Psi(k+1) \\ \vdots \\ \Psi(k+N-1) \end{bmatrix}$$

$$\tilde{C}_4 = \begin{bmatrix} C_4 \\ C_4 A \\ C_4 A^2 \\ \vdots \\ C_4 A^{N-1} \end{bmatrix} \qquad \tilde{D}_{42} = \begin{bmatrix} D_{42} & 0 & \cdots & 0 & 0 \\ C_4 B_2 & D_{42} & \cdots & 0 & 0 \\ C_4 A B_2 & C_4 B_2 & \ddots & \vdots & \vdots \\ \vdots & & \ddots & D_{42} & 0 \\ C_4 A^{N-2} B_2 & & \cdots & C_4 B_2 & D_{42} \end{bmatrix}$$

$$\tilde{D}_{41} = \begin{bmatrix} D_{41} \\ C_4 B_1 \\ C_4 A B_1 \\ \vdots \\ C_4 A^{N-2} B_1 \end{bmatrix} \qquad \tilde{D}_{43} = \begin{bmatrix} D_{43} & 0 & \cdots & 0 & 0 \\ C_4 B_3 & D_{43} & \cdots & 0 & 0 \\ C_4 A B_3 & C_4 B_3 & \ddots & \vdots & \vdots \\ \vdots & & \ddots & D_{43} & 0 \\ C_4 A^{N-2} B_3 & & \cdots & C_4 B_3 & D_{43} \end{bmatrix}$$

In this way we can formulate:

Inequality constraints on the output ($y(k+j) \leq y_{max}$):

$$\psi(k) = y(k) = C_4 x(k) + D_{41} e(k) + D_{42} w(k) + D_{43} v(k)$$

for $C_4 = C_o$, $D_{41} = D_{11}$, $D_{42} = D_{12}$, $D_{43} = D_{13}$ and $\Psi(k) = y_{max}$.

Inequality constraints on the input ($v(k+j) \leq v_{max}$):

$$\psi(k) = v(k) = C_4 x(k) + D_{41} e(k) + D_{42} w(k) + D_{43} v(k)$$

for $C_4 = 0$, $D_{41} = 0$, $D_{42} = 0$, $D_{43} = I$ and $\Psi(k) = v_{max}$.

Inequality constraints on the state ($x(k+j) \leq x_{max}$):

$$\psi(k) = x(k) = C_4 x(k) + D_{41} e(k) + D_{42} w(k) + D_{43} v(k)$$

for $C_4 = I$, $D_{41} = 0$, $D_{42} = 0$, $D_{43} = 0$ and $\Psi(k) = x_{max}$.

Problems occur when $\Psi(k)$ in equation (4.21) depends on past values of the input. In that case, prediction of $\Psi(k+j)$ may depends on future values of the input, when we use the derived prediction formulas. Two clear cases where this happen will be discussed next:

Constraint on the increment input signal for an IO model:

$$\Delta u(k+j|k) \le \Delta u_{max} \quad \text{for} \quad j=0,\ldots,N-1$$

Consider the case of an IO model, so $v(k)=u(k)$. Then by defining

$$\tilde{\Psi}(k) = \begin{bmatrix} u(k-1)+\Delta u_{max} \\ \Delta u_{max} \\ \vdots \\ \Delta u_{max} \end{bmatrix} \qquad \tilde{D}_{43} = \begin{bmatrix} I & 0 & \ldots & 0 \\ -I & I & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & -I & I \end{bmatrix}$$

$$\tilde{C}_4 = 0 \qquad \tilde{D}_{41} = 0 \qquad \tilde{D}_{42} = 0$$

the input increment constraint can be translated into the standard form of equation (4.20).

Constraint on the input signal for IIO model:

$$u(k+j|k) \le u_{max} \quad \text{for} \quad j=0,\ldots,N-1$$

Consider the case of an IIO model, so $v(k)=\Delta u(k)$. Then by defining

$$\tilde{\Psi}(k) = \begin{bmatrix} -u(k-1)+u_{max} \\ -u(k-1)+u_{max} \\ \vdots \\ -u(k-1)+u_{max} \end{bmatrix} \qquad \tilde{D}_{43} = \begin{bmatrix} I & 0 & \ldots & 0 \\ I & I & & \vdots \\ \vdots & & \ddots & 0 \\ I & \ldots & I & I \end{bmatrix}$$

the input constraint can be translated into the standard form of equation (4.20).

4.3 Structuring the input with orthogonal basis functions

On page 26 we introduced a parametrization of the input signal using a set of basis functions:

$$v(k+j|k) = \sum_{i=0}^{M} S_i(j)\,\alpha_i(k)$$

where $S_i(j)$ are the basis functions. A special set of basis functions are orthogonal basis functions [121], which have the property:

$$\sum_{k=-\infty}^{\infty} S_i(k)S_j(k) = \begin{cases} 0 & \text{for} \quad i \neq j \\ 1 & \text{for} \quad i = j \end{cases}$$

Lemma 8 *[120] Let A_b, B_b, C_b and D_b be matrices such that*

$$\begin{bmatrix} A_b & B_b \\ C_b & D_b \end{bmatrix}^T \begin{bmatrix} A_b & B_b \\ C_b & D_b \end{bmatrix} = I$$

(this means that A_b, B_b, C_b and D_b are the balanced system matrices of a unitary function). Define

$$A_v = \begin{bmatrix} A_b & B_bC_b & B_bD_bC_b & \cdots & B_bD_b^{n_v-2}C_b \\ 0 & A_b & B_bC_b & \cdots & B_bD_b^{n_v-3}C_b \\ \vdots & 0 & A_b & & \vdots \\ & & & \ddots & B_bC_b \\ 0 & \cdots & & \cdots & A_b \end{bmatrix}$$

$$C_v = \begin{bmatrix} C & D_bC_b & D_b^2C_b & \cdots & D_b^{n_v-1}C_b \end{bmatrix}$$

and let

$$S_i(k) = C_v A_v^k e_i$$

where

$$e_i = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T$$

Then, the functions $S_i(k)$ $i = 0, \ldots, M$ form an orthogonal basis.

When we define the parameter vector $\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_M \end{bmatrix}^T$, the input vector is given by the expression

$$v(k+j|k) = C_v\, A_v^j\, \alpha \tag{4.24}$$

The number of degrees of freedom in the input signal is now equal to $\dim(\alpha) = M = n_v n_1$, the dimension of parameter vector. One of the main reasons to apply orthogonal basis functions is to obtain a reduction of the optimization problem by reducing the degrees of freedom, so usually M will be much smaller than horizon N.
Consider an input sequence of N future values:

$$\tilde{v}(k) = \begin{bmatrix} v(k|k) \\ v(k+1|k) \\ \vdots \\ v(k+N|k) \end{bmatrix} = \begin{bmatrix} C_v \\ C_vA_v \\ \vdots \\ C_vA_v^N \end{bmatrix} \alpha = \mathcal{S}_v \alpha$$

Suppose N is such that $\mathcal{S}_v$ has full column-rank, and a left-complement is given by $\mathcal{S}_v^{\ell\perp}$ (so $\mathcal{S}_v^{\ell\perp}\mathcal{S}_v = 0$, see appendix C for a definition). Now we find that

$$\mathcal{S}_v^{\ell\perp}\, \tilde{v}(k) \;=\; 0 \tag{4.25}$$

We observe that the orthogonal basis function can be described by equality constraint 4.25.

4.4 The standard predictive control problem

From the previous section we learned that the GPC and LQPC performance index can be formulated in a standard form (equation 4.12). In fact most common quadratic performance indices can be given in this standard form. The same holds for many kinds of linear equality and inequality constraints, which can be formulated as (4.20) or (4.19). Summarizing we come to the formulation of the standard predictive control problem:

Definition 9 *Standard Predictive Control Problem (SPCP) Consider a system given by the state-space realization*

$$\begin{aligned} x(k+1) &= A\,x(k) + B_1\,e(k) + B_2\,w(k) + B_3\,v(k) && (4.26)\\ y(k) &= C_1\,x(k) + D_{11}\,e(k) + D_{12}\,w(k) && (4.27)\\ z(k) &= C_2\,x(k) + D_{21}\,e(k) + D_{22}\,w(k) + D_{23}\,v(k) && (4.28) \end{aligned}$$

The goal is to find a controller $v(k) = K(\tilde{w}, y, k)$ such that the performance index

$$J(v,k) = \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \tag{4.29}$$

is minimized subject to the constraints

$$\tilde{\phi}(k) = \tilde{C}_3 x(k) + \tilde{D}_{31} e(k) + \tilde{D}_{32}\tilde{w}(k) + \tilde{D}_{33}\tilde{v}(k) = 0 \tag{4.30}$$

$$\tilde{\psi}(k) = \tilde{C}_4 x(k) + \tilde{D}_{41} e(k) + \tilde{D}_{42}\tilde{w}(k) + \tilde{D}_{43}\tilde{v}(k) \le \tilde{\Psi}(k) \tag{4.31}$$

where

$$\tilde{v}(k) = \begin{bmatrix} v(k|k) \\ v(k+1|k) \\ \vdots \\ v(k+N-1|k) \end{bmatrix} \qquad \text{and} \qquad \tilde{w}(k) = \begin{bmatrix} w(k|k) \\ w(k+1|k) \\ \vdots \\ w(k+N-1|k) \end{bmatrix}$$

How the SPCP problem is solved will be discussed in the next chapter.

4.5 Examples

Example 10 : GPC as a standard predictive control problem

Consider the GPC problem on the IIO model of example 2. The purpose is to minimize the GPC performance index with

$$P(q) = 1 - 1.4q^{-1} + 0.48q^{-2}$$

In state space form this becomes:

$$A_p = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad B_p = \begin{bmatrix} -1.4 \\ 0.48 \end{bmatrix} \qquad C_p = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad D_p = \begin{bmatrix} 1 \end{bmatrix}$$

Further we choose $N = N_c = 4$ and $\lambda = 0.001$:

Performance index:

Following the formulas in section 4.1 the GPC performance index is tranformed into the standard performance index by choosing

$$A = \begin{bmatrix} A_p & B_pC_i \\ 0 & A_i \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1.4 & -1.4 & 0 \\ 0 & 0 & 0.48 & 0.48 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0.7 & 1 \\ 0 & 0 & 0 & -0.1 & 0 \end{bmatrix} \qquad B_1 = \begin{bmatrix} B_pD_H \\ K_i \end{bmatrix} = \begin{bmatrix} -1.4 \\ 0.48 \\ 1 \\ 1 \\ -0.2 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0 & 0 \\ L_i & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.1 & 0 \\ -0.05 & 0 \end{bmatrix} \qquad B_3 = \begin{bmatrix} 0 \\ B_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & C_i \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} C_pA_p & C_pB_pC_i + D_pC_iA_i \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -0.4 & 0.3 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D_{21} = \begin{bmatrix} C_pB_p + D_pC_iK_i \\ 0 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0 \end{bmatrix}$$

$$D_{22} = \begin{bmatrix} D_pC_iL_i & -I \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.1 & -1 \\ 0 & 0 \end{bmatrix} \qquad D_{23} = \begin{bmatrix} D_pC_iB_i \\ \lambda I \end{bmatrix} = \begin{bmatrix} 1 \\ 0.001 \end{bmatrix}$$

The prediction matrices are constructed according to chapter 3. We obtain:

$$\tilde{C}_2 = \begin{bmatrix} 0 & 1 & -0.4 & 0.3 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.08 & 0.19 & 0.3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.08 & 0.183 & 0.19 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.08 & 0.1891 & 0.183 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \tilde{D}_{21} = \begin{bmatrix} 0.6 \\ 0 \\ 0.18 \\ 0 \\ 0.21 \\ 0 \\ 0.225 \\ 0 \end{bmatrix}$$

$$\tilde{D}_{22} = \begin{bmatrix} 0.1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.02 & 0 & 0.1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.004 & 0 & -0.02 & 0 & 0.1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0088 & 0 & 0.004 & 0 & -0.02 & 0 & 0.1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{D}_{23} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.001 & 0 & 0 & 0 \\ 0.3 & 1 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0.19 & 0.3 & 1 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0.183 & 0.19 & 0.3 & 1.000 \\ 0 & 0 & 0 & 0.001 \end{bmatrix}$$

Example 11 : Constrained GPC as a standard predictive control problem

Now consider example 10 for the case where we have a constraint on the input signal:

$$u(k) \leq 1$$

and a control horizon constraint

$$\Delta u(k+N_c+j|k) = 0 \quad \textit{for} \;\; j = 0, \ldots, N-N_c-1$$

We will compute the matrices $\tilde{C}_3$*,* $\tilde{C}_4$ *,* $\tilde{D}_{31}$*,* $\tilde{D}_{32}$*,* $\tilde{D}_{33}$*,* $\tilde{D}_{41}$*,* $\tilde{D}_{42}$*,* $\tilde{D}_{43}$ *and the vector* $\tilde{\Psi}(k)$ *for* $N=4$ *and* $N_c=2$*.*

Following the formulas of section 4.2 the vector $\tilde{\Psi}$ *and the matrix* $\tilde{D}_{43}$ *become:*

$$\tilde{\Psi}(k) = \begin{bmatrix} -u(k-1)+u_{max} \\ -u(k-1)+u_{max} \\ -u(k-1)+u_{max} \\ -u(k-1)+u_{max} \end{bmatrix} = \begin{bmatrix} 1-u(k-1) \\ 1-u(k-1) \\ 1-u(k-1) \\ 1-u(k-1) \end{bmatrix}$$

$$\tilde{D}_{43} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Further $\tilde{C}_4 = 0$*,* $\tilde{D}_{41} = 0$ *and* $\tilde{D}_{42} = 0$*.*
Next we compute the matrices $\tilde{C}_3$*,* $\tilde{D}_{31}$*,* $\tilde{D}_{32}$ *and* $\tilde{D}_{33}$ *corresponding to the control horizon constraint (*$N=4$*,* $N_c=2$*):*

$$\tilde{C}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \tilde{D}_{31} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\tilde{D}_{32} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \tilde{D}_{33} = \begin{bmatrix} 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

Chapter 5

Solving the standard predictive control problem

In this chapter we solve the standard predictive control problem as defined in section 4.4. We consider the system given by the state-space realization

$$\begin{aligned} x(k+1) &= A\,x(k) + B_1\,e(k) + B_2\,w(k) + B_3\,v(k) && (5.1)\\ y(k) &= C_1\,x(k) + D_{11}\,e(k) + D_{12}\,w(k) && (5.2)\\ z(k) &= C_2\,x(k) + D_{21}\,e(k) + D_{22}\,w(k) + D_{23}\,v(k) && (5.3) \end{aligned}$$

The performance index and the constraints are given by

$$J(v,k) = \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \qquad (5.4)$$

$$\begin{aligned} \tilde{\phi}(k) &= \tilde{C}_3 x(k) + \tilde{D}_{31}e(k) + \tilde{D}_{32}\tilde{w}(k) + \tilde{D}_{33}\tilde{v}(k) = 0 && (5.5)\\ \tilde{\psi}(k) &= \tilde{C}_4 x(k) + \tilde{D}_{41}e(k) + \tilde{D}_{42}\tilde{w}(k) + \tilde{D}_{43}\tilde{v}(k) \le \tilde{\Psi}(k) && (5.6) \end{aligned}$$

The standard predictive control problem is to minimize (5.4) subject to (5.5) and (5.6). In this chapter we consider 3 subproblems:

1. The unconstrained SPCP: Minimize (5.4).

2. The equality constrained SPCP: Minimize (5.4) subject to (5.5).

3. The (full) SPCP: Minimize (5.4) subject to (5.5) and (5.6).

Assumptions

To be able to solve the standard predictive control problem, the following assumptions are made:

1. The matrix $\tilde{D}_{33}$ has full row rank.

2. The matrix $\tilde{D}_{23}\bar{\Gamma}\tilde{D}_{23}^T$ has full rank, where $\bar{\Gamma} = \text{diag}(\Gamma(0), \Gamma(1), \ldots, \Gamma(N-1)) \in \mathbb{R}^{n_z N \times n_z N}$.

3. The matrix D_{11} is square and invertible.

4. All eigenvalues of the matrix $(A - B_1 D_{11}^{-1} C_1)$ have magnitude smaller than 1.

5. The pair (A, B_3) is stabilizable.

Assumption 1 is necessary to be able to satisfy (5.5). Assumption 2 is necessary to obtain a unique solution to the optimization problem. Assumption 3 is necessary to obtain a stable observer (see section 6.1.1). Assumption 4 is necessary to be able to control all unstable modes of the system.

5.1 Steady-state behaviour

In predictive control it is often important to know how the signals behave for $k \longrightarrow \infty$. We therefore define the steady-state of a system in the standard predictive control formulation:

Definition 12
The quadruple $(v_{ss}, x_{ss}, w_{ss}, z_{ss})$ *is called* **a steady state** *if the following equations are satisfied:*

$$x_{ss} = A\, x_{ss} + B_2\, w_{ss} + B_3\, v_{ss} \tag{5.7}$$

$$z_{ss} = C_2\, x_{ss} + D_{22}\, w_{ss} + D_{23}\, v_{ss} \tag{5.8}$$

To be able to solve the predictive control problem it is important that for every possible w_{ss}, the quadruple $(v_{ss}, x_{ss}, w_{ss}, z_{ss})$ exists. We therefore look at the existence and uniqueness of a state x_{ss}, an input v_{ss} and a performance signal z_{ss}, corresponding to a specific signal w_{ss}.
Consider the matrix

$$N_{ss} = \begin{bmatrix} I - A & -B_3 & 0 \\ -C_2 & -D_{23} & I \end{bmatrix} \tag{5.9}$$

then (5.7) and (5.8) can be rewritten as:

$$N_{ss} \begin{bmatrix} x_{ss} \\ v_{ss} \\ z_{ss} \end{bmatrix} = \begin{bmatrix} B_2 \\ D_{22} \end{bmatrix} w_{ss}$$

Let the singular value decomposition be given by

$$N_{ss} = \begin{bmatrix} U_{N1} & U_{N2} \end{bmatrix} \begin{bmatrix} \Sigma_N & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{N1}^T \\ V_{N2}^T \end{bmatrix}$$

where $\Sigma_N = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_i > 0$.
A necessary and sufficient condition for existence of x_{ss}, v_{ss} and z_{ss} is that

$$U_{N2}^T \begin{bmatrix} B_2 \\ D_{22} \end{bmatrix} = 0 \tag{5.10}$$

The solution is given by:

$$\begin{bmatrix} x_{ss} \\ v_{ss} \\ z_{ss} \end{bmatrix} = V_{N1} \Sigma^{-1} U_{N1}^T \begin{bmatrix} B_2 \\ D_{22} \end{bmatrix} w_{ss} + V_{N2}\, \alpha_N \tag{5.11}$$

where α_N is an arbitrary vector with the apropriate dimension. Substitution of (5.11) in (5.7) and (5.8) shows that the quadruple $(v_{ss}, x_{ss}, w_{ss}, z_{ss})$ is a steady state. The solution (x_{ss}, v_{ss}, z_{ss}) for a given w_{ss} is unique if V_{N2} is empty, so if $N_{ss}^T N_{ss}$ has full rank.

To be able to solve the infinite horizon predictive control problem it is important that the performance signal z_{ss} is equal to zero for every w_{ss}. We therefore look at the existence of a steady state

$$(v_{ss}, x_{ss}, w_{ss}, z_{ss}) = (v_{ss}, x_{ss}, w_{ss}, 0)$$

Consider the matrix

$$M_{ss} = \begin{bmatrix} I - A & -B_3 \\ -C_2 & -D_{23} \end{bmatrix} \tag{5.12}$$

then (5.7) and (5.8) can be rewritten as:

$$M_{ss} \begin{bmatrix} x_{ss} \\ v_{ss} \end{bmatrix} = \begin{bmatrix} B_2 \\ D_{22} \end{bmatrix} w_{ss}$$

Let the singular value decomposition be given by

$$M_{ss} = \begin{bmatrix} U_{M1} & U_{M2} \end{bmatrix} \begin{bmatrix} \Sigma_M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{M1}^T \\ V_{M2}^T \end{bmatrix}$$

where $\Sigma_M = \mathrm{diag}(\sigma_1, \ldots, \sigma_m)$ with $\sigma_i > 0$.
A necessary and sufficient condition for existence of x_{ss} and v_{ss} is that

$$U_{M2}^T \begin{bmatrix} B_2 \\ D_{22} \end{bmatrix} = 0 \tag{5.13}$$

The solution is given by:

$$\begin{bmatrix} x_{ss} \\ v_{ss} \end{bmatrix} = V_{M1}\Sigma_M^{-1}U_{M1}^T \begin{bmatrix} B_2 \\ D_{22} \end{bmatrix} w_{ss} + V_{M2}\,\alpha_M \tag{5.14}$$

where α_M is an arbitrary vector with the apropriate dimension. Substitution of (5.14) in (5.7) and (5.8) shows that the quadruple $(v_{ss}, x_{ss}, w_{ss}, z_{ss})$=$(v_{ss}, x_{ss}, w_{ss}, 0)$ is a steady state. The solution (x_{ss}, v_{ss}) for a given w_{ss} is unique if V_{M2} is empty, so if $M_{ss}^T M_{ss}$ has full rank.
In the section on infinite horizon MPC we will assume that $z_{ss} = 0$ and that x_{ss} and v_{ss} exist, are unique, and related to w_{ss} by

$$\begin{bmatrix} x_{ss} \\ v_{ss} \end{bmatrix} = V_{M1}\Sigma^{-1}U_{M1}^T \begin{bmatrix} B_2 \\ D_{22} \end{bmatrix} w_{ss} \tag{5.15}$$

Usually we assume a linear relation between w_{ss} and $\tilde{w}(k)$:

$$w_{ss} = D_{ssw}\tilde{w}(k)$$

Sometimes w_{ss} is found by some extrapolation, but often we choose $w(k)$ to be constant beyond the prediction horizon, so $w_{ss} = w(k+j) = w(k+N-1)$ for $j \geq N$. The selection matrix is then given by $D_{ssw} = \begin{bmatrix} 0 & \cdots & 0 & I \end{bmatrix}$, such that $w_{ss} = w(k+N-1) = D_{ssw}\,\tilde{w}(k)$. Now define the matrices

$$\begin{bmatrix} D_{ssx} \\ D_{ssv} \end{bmatrix} = V_{M1}\Sigma^{-1}U_{M1}^T \begin{bmatrix} B_2 \\ D_{22} \end{bmatrix} D_{ssw} \tag{5.16}$$

and

$$D_{ssy} = C_1 D_{ssx} + D_{12}D_{ssw} + D_{13}D_{ssv} \tag{5.17}$$

then a direct relation between $(x_{ss}, v_{ss}, w_{ss}, y_{ss})$ and $\tilde{w}(k)$ is given by

$$\begin{bmatrix} x_{ss} \\ v_{ss} \\ w_{ss} \\ y_{ss} \end{bmatrix} = \begin{bmatrix} D_{ssx} \\ D_{ssv} \\ D_{ssw} \\ D_{ssy} \end{bmatrix} \tilde{w}(k) \tag{5.18}$$

5.1.1 Dynamic steady-state behaviour

Sometimes it is interesting to consider dynamic steady-state behaviour. This means that the signals x, w and v do not become constant for $k \longrightarrow \infty$, but some asymptotic or cyclic behaviour, so:

$$x_{ss}(k+j+1) = A\,x_{ss}(k+j) + B_2\,w_{ss}(k+j) + B_3\,v_{ss}(k+j) \tag{5.19}$$

$$z_{ss}(k+j) = C_2\,x_{ss}(k+j) + D_{22}\,w_{ss}(k+j) + D_{23}\,v_{ss}(k+j) \tag{5.20}$$

The quadruple $(v_{ss}(k), x_{ss}(k), w_{ss}(k), z_{ss}(k))$ is called **a dynamic steady state** if the equations (5.19) and (5.20) hold for all $j \geq 0$. For a cyclic steady state we assume there is a cycle period P such that $(v_{ss}(k+P), x_{ss}(k+P), w_{ss}(k+P), z_{ss}(k+P)) = (v_{ss}(k), x_{ss}(k), w_{ss}(k), z_{ss}(k))$. For infinite horizon predictive control we again make the assumption that $z_{ss}(k+j) = 0$, for all $j \geq 0$. In this way we can isolate the steady-state behaviour from the predictive control problem without affecting the performance index.

5.2 The finite horizon SPCP

5.2.1 Unconstrained standard predictive control problem

In this section we consider the problem of minimizing the performance index

$$J(v,k) = \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \tag{5.21}$$

for the system as given in (5.1), (5.2) and (5.3), and without equality or inequality constraints. In this section we consider the case for finite N. The infinite case will be treated in section 5.3.

The performance index, given in (5.4), is minimized for each time instant k, and the optimal future control sequence $\tilde{v}(k)$ is computed. We use the prediction law

$$\begin{aligned} \tilde{z}(k) &= \tilde{C}_2\, x(k) + \tilde{D}_{21}\, e(k) + \tilde{D}_{22}\, \tilde{w}(k) + \tilde{D}_{23}\, \tilde{v}(k) && (5.22) \\ &= \tilde{z}_0(k) + \tilde{D}_{23}\, \tilde{v}(k) && (5.23) \end{aligned}$$

as derived in chapter 3, where

$$\tilde{z}_0(k) = \tilde{C}_2\, x(k) + \tilde{D}_{21}\, e(k) + \tilde{D}_{22}\, \tilde{w}(k)$$

is the free-response signal given in equation (3.3). Without constraints, the problem becomes a standard least squares problem, and the solution of the predictive control problem can be computed analytically. Assume the reference trajectory $w(k+j)$ to be known over the whole prediction horizon $j = 0, \ldots, N$. Consider the performance index (5.4). It can be rewritten in vector-notation:

$$J(v,k) = \tilde{z}^T(k)\bar{\Gamma}\tilde{z}(k)$$

where $\bar{\Gamma} = \mathrm{diag}(\Gamma(0), \Gamma(1), \ldots, \Gamma(N-1)) \in \mathbb{R}^{n_z N \times n_z N}$.

Now substitute (5.23) and define matrix H, vector $f(k)$ and scalar $c(k)$ as

$$H = 2\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23} \quad , \quad f(k) = 2\tilde{D}_{23}^T\bar{\Gamma}\tilde{z}_0(k) \quad \text{and} \quad c(k) = \tilde{z}_0^T(k)\bar{\Gamma}\tilde{z}_0(k)$$

We obtain

$$\begin{aligned} J(v,k) &= \tilde{v}^T(k)\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\tilde{v}(k) + 2\tilde{v}^T(k)\tilde{D}_{23}^T\bar{\Gamma}\tilde{z}_0(k) + \tilde{z}_0^T(k)\bar{\Gamma}\tilde{z}_0(k) = \\ &= \frac{1}{2}\tilde{v}^T(k)H\tilde{v}(k) + \tilde{v}^T(k)f(k) + c(k) \end{aligned}$$

The minimization of $J(v,k)$ has become a linear algebra problem. It can be solved, setting the gradient of J to zero:

$$\frac{\partial J}{\partial \tilde{v}} = H\tilde{v}(k) + f(k) = 0$$

For invertible H, the solution $\tilde{v}(k)$ is given by

$$\begin{aligned}
\tilde{v}(k) &= -H^{-1}f(k) \\
&= -\Big(\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\Big)^{-1}\tilde{D}_{23}^T\bar{\Gamma}\tilde{z}_0(k) \\
&= -\Big(\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\Big)^{-1}\tilde{D}_{23}^T\bar{\Gamma}\Big(\tilde{C}_2\,x(k) + \tilde{D}_{21}e(k) + \tilde{D}_{22}\,\tilde{w}(k)\Big)
\end{aligned}$$

Because we use the receding horizon principle, only the first computed change in the control signal is implemented and at time $k+1$ the computation is repeated with the horizon moved one time interval. This means that at time k the control signal is given by

$$v(k|k) = \begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix}\tilde{v}(k) = E_v\,\tilde{v}(k)$$

We obtain

$$\begin{aligned}
v(k|k) &= E_v\,\tilde{v}(k) \\
&= -E_v\Big(\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\Big)^{-1}\tilde{D}_{23}^T\bar{\Gamma}\Big(\tilde{C}_2\,x(k) + \tilde{D}_{21}\,e(k) + \tilde{D}_{22}\tilde{w}(k)\Big) \\
&= -F\,x(k) + D_e\,e(k) + D_w\,\tilde{w}(k)
\end{aligned}$$

where

$$\begin{aligned}
F &= E_v\,(\,\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\,)^{-1}\tilde{D}_{23}^T\bar{\Gamma}\tilde{C}_2 \\
D_e &= -E_v\,(\,\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\,)^{-1}\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{21} \\
D_w &= -E_v\,(\,\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\,)^{-1}\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{22} \\
E_v &= \begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix}
\end{aligned}$$

with E_v such that $v(k|k) = E_v\tilde{v}(k)$.
The results are summarized in the following theorem (Kinnaert [62],Lee *et al.* [69],Van den Boom & de Vries [118]):

Theorem 13
Consider system (5.1) - (5.3) The unconstrained (finite) horizon standard predictive control problem of minimizing performance index

$$J(v,k) = \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \tag{5.24}$$

is solved by control law

$$v(k) = -F\,x(k) + D_e\,e(k) + D_w\,\tilde{w}(k) \tag{5.25}$$

where

$$
\begin{aligned}
F &= E_v\,(\,\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\,)^{-1}\tilde{D}_{23}^T\bar{\Gamma}\tilde{C}_2 && (5.26)\\
D_e &= -E_v\,(\,\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\,)^{-1}\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{21} && (5.27)\\
D_w &= -E_v\,(\,\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\,)^{-1}\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{22} && (5.28)\\
E_v &= \begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix} && (5.29)
\end{aligned}
$$

Note that for the computation of optimal input signal $v(k)$, the state $x(k)$ and noise signal $e(k)$ have to be available. This can easily be done with an observer and will be discussed in section 5.4.

5.2.2 Equality constrained standard predictive control problem

In this section we consider the problem of minimizing the performance index

$$J(v,k) \;=\; \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \qquad (5.30)$$

for the system as given in (5.1), (5.2) and (5.3), and with equality constraints (but without inequality constraints). In this section we consider the case for finite N. In section 5.3.3 we discuss the case where $N=\infty$ and a finite control horizon as the equality constraint.

Consider the equality constrained standard problem

$$\min_{\tilde{v}}\ \tilde{z}^T(k)\bar{\Gamma}\tilde{z}(k)$$

subject to the constraint:

$$\tilde{\phi}(k) \;=\; \tilde{C}_3x(k)+\tilde{D}_{31}e(k)+\tilde{D}_{32}\tilde{w}(k)+\tilde{D}_{33}\tilde{v}(k)=0$$

We will solve this problem by elimination of the equality constraint. We therefore give the following lemma:

Lemma 14
Consider the equality constraint

$$\mathcal{C}\,\theta=\beta \qquad (5.31)$$

for full-row rank matrix $\mathcal{C}\in \mathbb{R}^{m\times n}$*,* $\beta\in\mathbb{R}^{m\times 1}$*,* $\theta\in\mathbb{R}^{n\times 1}$*, and* $m\le n$*. Define* $\mathcal{C}^r$ *as the right-inverse of* $\mathcal{C}$ *and* $\mathcal{C}^{r\perp}$ *as the right-complement of* $\mathcal{C}$*, then all* θ *satisfying (5.31) are given by*

$$\theta=\mathcal{C}^r\,\beta+\mathcal{C}^{r\perp}\eta$$

where $\eta\in\mathbb{R}^{(n-m)\times 1}$ *is arbitrary.*

□ End Lemma

Proof of lemma 14:
From appendix C we know that for

$$C = U \begin{bmatrix} \Sigma & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

the right-inverse $\mathcal{C}^r$ and right-complement $\mathcal{C}^{r\perp}$ are defined as

$$C^r = V_1\,\Sigma^{-1}\,U^T \qquad C^{r\perp} = V_2$$

Choose vectors ζ and η as follows:

$$\begin{bmatrix} \zeta \\ \eta \end{bmatrix} = \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}\theta \qquad \text{so} \qquad \theta = \begin{bmatrix} V_1 & V_2 \end{bmatrix}\begin{bmatrix} \zeta \\ \eta \end{bmatrix}$$

then there must hold

$$\mathcal{C}\,\theta = U \begin{bmatrix} \Sigma & 0 \end{bmatrix} \begin{bmatrix} \zeta \\ \eta \end{bmatrix} = U\,\Sigma\,\zeta = \beta$$

and so we have to choose $\zeta = \Sigma^{-1}\,U^T\,\beta$, while η can be any vector (with the proper dimension). Therefore, all θ satisfying (5.31) are given by

$$\theta = V_1\,\Sigma^{-1}\,U^T\beta + V_2\eta = \mathcal{C}^r\,\beta + \mathcal{C}^{r\perp}\eta$$

□ End Proof

Define $\tilde{D}_{33}^r$ and $\tilde{D}_{33}^{r\perp}$ as in the above lemma, then all $\tilde{v}$ that satisfy the equality constraint $\tilde{\phi}(k) = 0$ are given by:

$$\begin{aligned} \tilde{v}(k) &= -\tilde{D}_{33}^r\Big(\tilde{C}_3 x(k) + \tilde{D}_{31} e(k) + \tilde{D}_{32}\tilde{w}(k)\Big) + \tilde{D}_{33}^{r\perp}\tilde{\mu}(k) \\ &= \tilde{v}_E(k) + \tilde{D}_{33}^{r\perp}\tilde{\mu}(k) \end{aligned}$$

Note that by choosing $\tilde{v}(k) = \tilde{v}_E(k) + \tilde{D}_{33}^{r\perp}\tilde{\mu}(k)$ the equality constraint is always satisfied while all remaining freedom is still in the vector $\tilde{\mu}(k)$, which is of lower dimension. So by this choice the problem has reduced in order while the equality constraint is always satisfied for all $\tilde{\mu}(k)$.

Substitution of $\tilde{v}(k) = \tilde{v}_E(k) + \tilde{D}_{33}^{r\perp}\tilde{\mu}(k)$ in (5.22) gives:

$$\begin{aligned} \tilde{z}(k) &= \tilde{C}_2\,x(k) + \tilde{D}_{21}\,e(k) + \tilde{D}_{22}\,\tilde{w}(k) + \tilde{D}_{23}\,\tilde{v}(k) \\ &= (\tilde{C}_2 - \tilde{D}_{23}\tilde{D}_{33}^r\tilde{C}_3)x(k) + (\tilde{D}_{21} - \tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{31})e(k) \\ &+ (\tilde{D}_{22} - \tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{32})\tilde{w}(k) + \tilde{D}_{23}\tilde{D}_{33}^{r\perp}\tilde{\mu}(k) \\ &= \tilde{z}_E(k) + \tilde{D}_{23}\tilde{D}_{33}^{r\perp}\tilde{\mu}(k) \end{aligned}$$

where $\tilde{z}_E$ is given by:

$$\tilde{z}_E(k)=(\tilde{C}_2-\tilde{D}_{23}\tilde{D}_{33}^r\tilde{C}_3)x(k)+(\tilde{D}_{21}-\tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{31})e(k)+(\tilde{D}_{22}-\tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{32})\tilde{w}(k)$$

The performance index is given by

$$\begin{aligned}
J(k) &= \tilde{z}^T(k)\bar{\Gamma}\tilde{z}(k) = && (5.32)\\
&= (\tilde{z}_E(k)+\tilde{D}_{23}\tilde{D}_{33}^{r\perp}\tilde{\mu}(k))^T\bar{\Gamma}(\tilde{z}_E(k)+\tilde{D}_{23}\tilde{D}_{33}^{r\perp}\tilde{\mu}(k)) = && (5.33)\\
&= \tilde{\mu}^T(k)\Big((\tilde{D}_{33}^{r\perp})^T\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\tilde{D}_{33}^{r\perp}\Big)\tilde{\mu}(k)+2\tilde{z}_E^T(k)\bar{\Gamma}\tilde{D}_{23}\tilde{D}_{33}^{r\perp}\tilde{\mu}(k)+\tilde{z}_E^T(k)\bar{\Gamma}\tilde{z}_E(k)= && (5.34)\\
&= \frac{1}{2}\tilde{\mu}^TH\tilde{\mu}+f^T(k)\tilde{\mu}(k)+c(k) && (5.35)
\end{aligned}$$

where the matrices H, f and c are given by

$$\begin{aligned}
H &= 2\,(\tilde{D}_{33}^{r\perp})^T\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\tilde{D}_{33}^{r\perp}\\
f(k) &= 2\,(\tilde{D}_{33}^{r\perp})^T\tilde{D}_{23}^T\bar{\Gamma}z_E(k)\\
c(k) &= z_E^T\bar{\Gamma}z_E
\end{aligned}$$

When inequality constraints are absent, we only have left an unconstrained optimization

$$\min_{\tilde{\mu}}\frac{1}{2}\tilde{\mu}^T\,H\,\tilde{\mu}+f^T\tilde{\mu}$$

which has an analytical solution. In the same way as in section 5.1 we derive:

$$\frac{\partial J}{\partial\tilde{\mu}}=H\tilde{\mu}(k)+f(k)=0$$

and so

$$\begin{aligned}
\tilde{\mu}(k) &= -H^{-1}f(k)\\
&= -\Big((\tilde{D}_{33}^{r\perp})^T\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\tilde{D}_{33}^{r\perp}\Big)^{-1}(\tilde{D}_{33}^{r\perp})^T\tilde{D}_{23}^T\bar{\Gamma}z_E(k)\\
&= -\Xi\,\tilde{z}_E(k) && (5.36)\\
&= (\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{C}_3-\Xi\tilde{C}_2)x(k)+(\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{31}-\Xi\tilde{D}_{21})e(k)\\
&+ (\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{32}-\Xi\tilde{D}_{22})\tilde{w}(k) && (5.37)
\end{aligned}$$

where

$$\Xi=\Big((\tilde{D}_{33}^{r\perp})^T\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\tilde{D}_{33}^{r\perp}\Big)^{-1}(\tilde{D}_{33}^{r\perp})^T\tilde{D}_{23}^T\bar{\Gamma}$$

Substitution gives:

$$\begin{aligned}
\tilde{v}(k) &= -\tilde{D}_{33}^r\Big(\tilde{C}_3x(k)+\tilde{D}_{31}e(k)+\tilde{D}_{32}\tilde{w}(k)\Big)+\tilde{D}_{33}^{r\perp}\tilde{\mu}(k)\\
&= -\tilde{D}_{33}^r\Big(\tilde{C}_3x(k)+\tilde{D}_{31}e(k)+\tilde{D}_{32}\tilde{w}(k)\Big)+\tilde{D}_{33}^{r\perp}(\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{C}_3-\Xi\tilde{C}_2)x(k)+\\
&\quad +\tilde{D}_{33}^{r\perp}(\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{31}-\Xi\tilde{D}_{21})e(k)+\tilde{D}_{33}^{r\perp}(\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{32}-\Xi\tilde{D}_{22})\tilde{w}(k)\\
&= (\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{C}_3-\tilde{D}_{33}^{r\perp}\Xi\tilde{C}_2-\tilde{D}_{33}^r\tilde{C}_3)x(k)\\
&\quad +(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{31}-\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{21}-\tilde{D}_{33}^r\tilde{D}_{31})e(k)+\\
&\quad +(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{32}-\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{22}-\tilde{D}_{33}^r\tilde{D}_{32})\tilde{w}(k)
\end{aligned}$$

The control signal can now be written as:

$$\begin{aligned}
v(k|k) &= E_v \tilde{v}(k) \\
&= E_v(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{C}_3 - \tilde{D}_{33}^{r\perp}\Xi\tilde{C}_2 - \tilde{D}_{33}^{r}\tilde{C}_3)x(k) + \\
&\quad +E_v(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{D}_{31} - \tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{21} - \tilde{D}_{33}^{r}\tilde{D}_{31})e(k) + \\
&\quad +E_v(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{D}_{32} - \tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{22} - \tilde{D}_{33}^{r}\tilde{D}_{32})\tilde{w}(k) \\
&= -F\,x(k) + D_e\,e(k) + D_w\,\tilde{w}(k)
\end{aligned}$$

where

$$\begin{aligned}
F &= -E_v(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{C}_3 + \tilde{D}_{33}^{r\perp}\Xi\tilde{C}_2 + \tilde{D}_{33}^{r}\tilde{C}_3) \\
D_e &= E_v(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{D}_{31} - \tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{21} - \tilde{D}_{33}^{r}\tilde{D}_{31}) \\
D_w &= E_v(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{D}_{32} - \tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{22} - \tilde{D}_{33}^{r}\tilde{D}_{32})
\end{aligned}$$

This means that also for the equality constrained case, the optimal predictive controller is linear and time-invariant.
The control law has the same form as for the unconstrained case in the previous section (of course for the alternative choices of F, D_w and D_e).
The results are summarized in the following theorem:

Theorem 15
Consider system (5.1) - (5.3) The equality constrained (finite) horizon standard predictive control problem of minimizing performance index

$$J(v,k) = \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \tag{5.38}$$

subject to equality constraint

$$\tilde{\phi}(k) = \tilde{C}_3 x(k) + \tilde{D}_{31}e(k) + \tilde{D}_{32}\tilde{w}(k) + \tilde{D}_{33}\tilde{v}(k) = 0$$

is solved by control law

$$v(k) = -F\,x(k) + D_e\,e(k) + D_w\,\tilde{w}(k) \tag{5.39}$$

where

$$\begin{aligned}
F &= -E_v\left(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{C}_3 - \tilde{D}_{33}^{r\perp}\Xi\tilde{C}_2 - \tilde{D}_{33}^{r}\tilde{C}_3\right) \\
D_e &= E_v\left(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{D}_{31} - \tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{21} - \tilde{D}_{33}^{r}\tilde{D}_{31}\right) \\
D_w &= E_v\left(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{D}_{32} - \tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{22} - \tilde{D}_{33}^{r}\tilde{D}_{32}\right) \\
E_v &= \begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix} \\
\Xi &= \left((\tilde{D}_{33}^{r\perp})^T\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\tilde{D}_{33}^{r\perp}\right)^{-1}(\tilde{D}_{33}^{r\perp})^T\tilde{D}_{23}^T\bar{\Gamma}
\end{aligned}$$

Again, implementation issues are discussed in section 5.4.

5.2.3 Full standard predictive control problem

In this section we consider the problem of minimizing the performance index

$$J(v,k) \;=\; \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \tag{5.40}$$

for the system as given in (5.1), (5.2) and (5.3), and with equality and inequality constraints. In this section we consider the case for finite N. The infinite horizon case is treated in the sections 5.3.3and 5.3.4.

Consider the constrained standard problem

$$\min_{\tilde{v}} \tilde{z}^T(k)\bar{\Gamma}\tilde{z}(k)$$

subject to the constraints:

$$\tilde{\phi}(k) \;=\; \tilde{C}_3x(k)+\tilde{D}_{31}e(k)+\tilde{D}_{32}\tilde{w}(k)+\tilde{D}_{33}\tilde{v}(k)=0 \tag{5.41}$$

$$\tilde{\psi}(k) \;=\; \tilde{C}_4x(k)+\tilde{D}_{41}e(k)+\tilde{D}_{42}\tilde{w}(k)+\tilde{D}_{43}\tilde{v}(k)\leq\tilde{\Psi}(k) \tag{5.42}$$

Now $\tilde{v}(k)$ is chosen as in the previous section

$$\begin{aligned}\tilde{v}(k) &= -\tilde{D}_{33}^{r}\Big(\tilde{C}_3x(k)+\tilde{D}_{31}e(k)+\tilde{D}_{32}\tilde{w}(k)\Big)+\tilde{D}_{33}^{r\perp}\tilde{\mu}(k)\\ &= \tilde{v}_E(k)+\tilde{D}_{33}^{r\perp}\tilde{\mu}(k)\end{aligned}$$

This results in

$$\begin{aligned}\tilde{\phi}(k) &= \tilde{C}_3x(k)+\tilde{D}_{31}e(k)+\tilde{D}_{32}\tilde{w}(k)+\tilde{D}_{33}\tilde{v}(k)=\\ &= \tilde{C}_3x(k)+\tilde{D}_{31}e(k)+\tilde{D}_{32}\tilde{w}(k)+\tilde{D}_{33}\tilde{v}_E(k)+\tilde{D}_{33}\tilde{D}_{33}^{r\perp}\tilde{\mu}(k)=0\end{aligned}$$

and so the equality constraint is eliminated. The optimization vector $\tilde{\mu}(k)$ can now be written as

$$\tilde{\mu}(k) \;=\; \tilde{\mu}_E(k)+\tilde{\mu}_I(k) \tag{5.43}$$

$$\;=\; -H^{-1}f(k)+\tilde{\mu}_I(k) \tag{5.44}$$

$$\tag{5.45}$$

where $\tilde{\mu}_E(k)=-H^{-1}f(k)$ is the equality constrained solution given in the previous section, and $\tilde{\mu}_i(k)$ is an additional term to take the inequality constraints into account. Now consider performance index (5.35) from the previous section. Substitution of $\tilde{\mu}(k) = -H^{-1}f(k)+\tilde{\mu}_I(k)$ gives us:

$$\begin{aligned}&\frac{1}{2}\tilde{\mu}^T(k)\,H\,\tilde{\mu}(k)+f^T(k)\tilde{\mu}(k)+c(k)=\\ &\quad= \frac{1}{2}\Big(-H^{-1}f(k)+\tilde{\mu}_I(k)\Big)^T H\Big(-H^{-1}f(k)+\tilde{\mu}_I(k)\Big)\end{aligned}$$

$$\begin{aligned}
&+f^T(k)\Big(-H^{-1}f(k)+\tilde{\mu}_I(k)\Big)+c(k) = \\
= \;& \frac{1}{2}\tilde{\mu}_I^T(k)\,H\,\tilde{\mu}_I(k) - f^T(k)H^{-1}H\,\tilde{\mu}_I(k) \\
&+\frac{1}{2}f^T(k)H^{-1}\,H\,H^{-1}f(k) - f^T(k)H^{-1}f(k) + f^T(k)\tilde{\mu}_I(k) + c(k) \\
= \;& \frac{1}{2}\tilde{\mu}_I^T(k)\,H\,\tilde{\mu}_I(k) - \frac{1}{2}f^T(k)H^{-1}f(k) + c(k)
\end{aligned}$$

Using equations (5.36)-(5.37), the vector $\tilde{\mu}(k)$ can be written as

$$\begin{aligned}
\tilde{\mu}(k) &= \tilde{\mu}_E(k) + \tilde{\mu}_I(k) \\
&= -\Xi\,\tilde{z}_E(k) + \tilde{\mu}_I(k) \\
&= (\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{C}_3 - \Xi\tilde{C}_2)x(k) + (\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{31} - \Xi\tilde{D}_{21})e(k) \\
&+ (\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{32} - \Xi\tilde{D}_{22})\tilde{w}(k) + \tilde{\mu}_I(k)
\end{aligned} \tag{5.46}$$

Substitution of $\tilde{v}(k) = \tilde{v}_E(k) + \tilde{D}_{33}^{r\perp}\tilde{\mu}(k) = \tilde{v}_E(k) + \tilde{D}_{33}^{r\perp}\tilde{\mu}_E(k) + \tilde{D}_{33}^{r\perp}\tilde{\mu}_I(k)$ in inequality constraint (5.42) gives:

$$\begin{aligned}
\tilde{\psi}(k) &= (\tilde{C}_4 - \tilde{D}_{43}\tilde{D}_{33}^r\tilde{C}_3)x(k) + (\tilde{D}_{41} - \tilde{D}_{43}\tilde{D}_{33}^r\tilde{D}_{31})e(k) \\
&\quad +(\tilde{D}_{42} - \tilde{D}_{43}\tilde{D}_{33}^r\tilde{D}_{32})\tilde{w}(k) + \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\tilde{\mu}_E(k) + \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\tilde{\mu}_I(k) \\
&= (\tilde{C}_4 - \tilde{D}_{43}\tilde{D}_{33}^r\tilde{C}_3 + \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{C}_3 - \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\Xi\tilde{C}_2)x(k) \\
&\quad +(\tilde{D}_{41} - \tilde{D}_{43}\tilde{D}_{33}^r\tilde{D}_{31} + \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{31} - \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{21})e(k) \\
&\quad +(\tilde{D}_{42} - \tilde{D}_{43}\tilde{D}_{33}^r\tilde{D}_{32} + \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^r\tilde{D}_{32} - \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{22})\tilde{w}(k) \\
&\quad +\tilde{D}_{43}\tilde{D}_{33}^{r\perp}\tilde{\mu}_I(k) \\
&= \tilde{\psi}_E(k) + \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\tilde{\mu}_I(k)
\end{aligned}$$

Now let

$$\begin{aligned}
A_\psi &= \tilde{D}_{43}\tilde{D}_{33}^{r\perp} \\
b_\psi(k) &= \tilde{\psi}_E(k) - \tilde{\Psi}(k)
\end{aligned}$$

then we obtain the quadratic programming problem of minimizing

$$\min_{\tilde{\mu}(k)} \frac{1}{2}\tilde{\mu}_I^T(k)\,H\,\tilde{\mu}_I(k)$$

subject to:

$$A_\psi\tilde{\mu}(k) + b_\psi(k) \le 0$$

where H is as in the previous section and we dropped the term $-\frac{1}{2}f^T(k)H^{-1}f(k)+c(k)$, because it is not dependent on $\tilde{\mu}_I(k)$ and so it has no influence on the minimization. The above optimization problem, which has to be solved numerically and on-line, is a quadratic programming problem and can be solved in finite number of iterative steps using reliable and fast algorithms (see appendix A).

The control law of the full SPCP is nonlinear, and cannot be expressed in a linear form as the unconstrained SPCP or equality constrained SPCP.

Theorem 16
Consider system (5.1) - (5.3) and the constrained (finite horizon) standard predictive control problem (CSPCP) of minimizing performance index

$$J(v,k) = \sum_{j=0}^{N-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \tag{5.47}$$

subject to equality and inequality constraints

$$\begin{aligned}
\tilde{\phi}(k) &= \tilde{C}_3 x(k) + \tilde{D}_{31}e(k) + \tilde{D}_{32}\tilde{w}(k) + \tilde{D}_{33}\tilde{v}(k) = 0 \\
\tilde{\psi}(k) &= \tilde{C}_4 x(k) + \tilde{D}_{41}e(k) + \tilde{D}_{42}\tilde{w}(k) + \tilde{D}_{43}\tilde{v}(k) \le \tilde{\Psi}(k)
\end{aligned}$$

Define:

$$\begin{aligned}
A_\psi &= \tilde{D}_{43}\tilde{D}_{33}^{r\perp} \\
b_\psi(k) &= \tilde{\psi}_E(k) - \tilde{\Psi}(k) \\
&= (\tilde{C}_4 - \tilde{D}_{43}\tilde{D}_{33}^{r}\tilde{C}_3 + \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{C}_3 - \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\Xi\tilde{C}_2)x(k) \\
&\quad +(\tilde{D}_{41} - \tilde{D}_{43}\tilde{D}_{33}^{r}\tilde{D}_{31} + \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{D}_{31} - \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{21})e(k) \\
&\quad +(\tilde{D}_{42} - \tilde{D}_{43}\tilde{D}_{33}^{r}\tilde{D}_{32} + \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{D}_{32} - \tilde{D}_{43}\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{22})\tilde{w}(k) \\
&\quad -\tilde{\Psi}(k) \\
H &= 2\,(\tilde{D}_{33}^{r\perp})^T\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\tilde{D}_{33}^{r\perp} \\
F &= -E_v\left(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{C}_3 - \tilde{D}_{33}^{r\perp}\Xi\tilde{C}_2 - \tilde{D}_{33}^{r}\tilde{C}_3\right) \\
D_e &= E_v\left(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{D}_{31} - \tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{21} - \tilde{D}_{33}^{r}\tilde{D}_{31}\right) \\
D_w &= E_v\left(\tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{23}\tilde{D}_{33}^{r}\tilde{D}_{32} - \tilde{D}_{33}^{r\perp}\Xi\tilde{D}_{22} - \tilde{D}_{33}^{r}\tilde{D}_{32}\right) \\
D_\mu &= E_v\tilde{D}_{33}^{r\perp} \\
\Xi &= \left((\tilde{D}_{33}^{r\perp})^T\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23}\tilde{D}_{33}^{r\perp}\right)^{-1}(\tilde{D}_{33}^{r\perp})^T\tilde{D}_{23}^T\bar{\Gamma} \\
E_v &= \begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix}
\end{aligned}$$

The optimal control law that optimizes this CSPCP problem is given by:

$$v(k) = -F\,x(k) + D_e\,e(k) + D_w\,\tilde{w}(k) + D_\mu\,\tilde{\mu}_I(k)$$

where $\tilde{\mu}$ is the solution for the quadratic programming problem

$$\min_{\tilde{\mu}_I(k)} \frac{1}{2}\tilde{\mu}_I^T(k)\,H\,\tilde{\mu}_I(k)$$

subject to:

$$A_\psi\tilde{\mu}_I(k) + b_\psi(k) \le 0$$

5.3 Infinite horizon SPCP

In this section we consider the standard predictive control problem for the case where the prediction horizon is infinite ($N = \infty$) [106].

5.3.1 Structuring the input signal for infinite horizon MPC

For infinite horizon predictive control we will encounter an optimization problem with an infinite degrees of freedom, parameterized by $v(k+j|k)$, $j = 1, \ldots, \infty$. For the unconstrained case this is not a problem, and we can find the optimal predictive controller using theory on Linear Quadratic Gaussian (LQG) controllers (Strejc, [112]). This is given in section 5.3.2.

In the constrained case, we can tackle the problem by giving the input signal only a limited degrees of freedom. We can do that by introducing the so-called switching horizon, denoted by N_s. This N_s has to be chosen such, that all degrees of freedom are in the first N_s samples of the input signal, so in $v(k+j|k)$, $j = 0, \ldots, N_s - 1$. In this section we therefore adopt the input sequence vector

$$\tilde{v}(k) = \begin{bmatrix} v(k|k) \\ v(k+1|k) \\ \vdots \\ v(k+N_s-1|k) \end{bmatrix}$$

which contains all degrees of freedom at sample time k.

There are several ways to expand the input signal beyond the switching horizon:

Control horizon: By taking $N_s = N_c$, we choose $v(k+j|k) = v_{ss}$ for $j \geq N_s$. (Note that $v_{ss} = 0$ for IIO models).

Constant state feedback: We can choose $v(k+j|k) = F_v\left(\hat{x}(k+j|k) - x_{ss}\right) + v_{ss}$ for $j \geq N_s$, where F_v is chosen on beforehand. An appropriate value for F_v is the optimal feedback F for the unconstrained case where we choose $N = N_c = N_s$. By taking the steady state values x_{ss} and v_{ss} into account, we are sure that $v(k+j) = v_{ss}$ as soon as $\hat{x} = x_{ss}$. This means that the steady state is not affected.

Basis functions: We can choose $v(k+j|k)$ for $j \geq N_s$ as a finite sum of (orthogonal) basis functions. The future input $v(k+j|k)$ can be formulated using equation (4.24):

$$v(k+j|k) = C_v\, A_v^j\, \alpha(k) + v_{ss}$$

where we have taken into account the steady-state value v_{ss}. Following the definitions from section 4.3, we derived the relation

$$\tilde{v}(k) = \begin{bmatrix} v(k) \\ v(k+1) \\ \vdots \\ v(k+N_s-1) \end{bmatrix} = \begin{bmatrix} C_v \\ C_v A_v \\ \vdots \\ C_v A_v^{N_s-1} \end{bmatrix} \alpha + \begin{bmatrix} v_{ss} \\ v_{ss} \\ \vdots \\ v_{ss} \end{bmatrix} = \mathcal{S}_v \alpha + \tilde{v}_{ss}$$

Suppose N_s is such that $\mathcal{S}_v$ has full column-rank, and a left-complement is given by $\mathcal{S}_v^{\ell\perp}$ (so $\mathcal{S}_v^{\ell\perp}\mathcal{S}_v = 0$, see appendix C for a definition). Now we find that

$$\mathcal{S}_v^{\ell\perp}\,(\tilde{v}(k) - \tilde{v}_{ss}) \;=\; 0 \tag{5.48}$$

or, by defining

$$\tilde{v}_{ss} = \begin{bmatrix} v_{ss} \\ v_{ss} \\ \vdots \\ v_{ss} \end{bmatrix} = \begin{bmatrix} D_{ssv} \\ D_{ssv} \\ \vdots \\ D_{ssv} \end{bmatrix} \tilde{w}(k) = \tilde{D}_{ssv}\,\tilde{w}(k)$$

we obtain

$$\mathcal{S}_v^{\ell\perp}\,\tilde{v}(k) - \mathcal{S}_v^{\ell\perp}\,\tilde{D}_{ssv}\tilde{w}(k) \;=\; 0 \tag{5.49}$$

We observe that the orthogonal basis function can be described by equality constraint 5.49.

By defining $B_v = A_v^{N_s}\mathcal{S}_v^{\ell}$ (where $\mathcal{S}_v^{\ell}$ is the left inverse of $\mathcal{S}_v$, see appendix C for a definition), we can express α in $\tilde{v}$

$$A_v^{N_s}\alpha(k) \;=\; B_v\,\tilde{v}(k)$$

and the future input signals beyond N_s can be expressed in terms of $\tilde{v}(k)$:

$$v(k+j) = C_v\,A_v^j\,\alpha(k) + v_{ss} = C_v\,A_v^{j-N_s}\,B_v\,\tilde{v}(k) + v_{ss} \tag{5.50}$$

Combining the option of a state feedback and a description with (orthogonal) basis functions gives us that, for the constrained case, the input signal beyond N_s can be described as follows:

$$v(k+j|k) = C_v\,A_v^{j-N_s}\,B_v\,\tilde{v}(k) + F_v(\hat{x}(k+j|k) - x_{ss}) + v_{ss} \qquad j \geq N_s \tag{5.51}$$

or

$$\begin{aligned} x_v(k+N_s|k) &= B_v\,\tilde{v}(k) && && (5.52)\\ x_v(k+j+1|k) &= A_v\,x_v(k+j|k) && j \geq N_s && (5.53)\\ v(k+j|k) &= C_v\,x_v(k+j|k) + F_v(x(k+j|k) - x_{ss}) + v_{ss} && j \geq N_s && (5.54) \end{aligned}$$

where x_v is an additional state, describing the dynamics of the (orthogonal) basis function beyond the switching horizon.

5.3.2 Unconstrained infinite horizon SPCP

In this section we consider the unconstrained infinite horizon standard predictive control problem. To be able to tackle this problem, we will consider a constant external signal w, a zero steady-state performance signal and the weighting matrix $\Gamma(j)$ is equal to identity, so

$$\begin{aligned} w(k+j) &= w_{ss} \quad \text{for all } \; j \geq 0 \\ z_{ss} &= 0 \\ \Gamma(j) &= I \quad \text{for all } \; j \geq 0 \end{aligned}$$

Define

$$\begin{aligned} x_\Delta(k+j|k) &= x(k+j|k) - x_{ss} \quad \text{for all } \; j \geq 0 \\ v_\Delta(k+j|k) &= v(k+j|k) - v_{ss} \quad \text{for all } \; j \geq 0 \end{aligned}$$

then it follows that

$$\begin{aligned} \hat{x}_\Delta(k+j+1|k) &= \hat{x}(k+j+1|k) - x_{ss} \\ &= A\,\hat{x}(k+j|k) + B_1\,\hat{e}(k+j|k) + B_2\,w(k+j) + B_3\,v(k+j) - x_{ss} \\ &= A\,\hat{x}(k+j|k) + B_1\,\hat{e}(k+j|k) + B_2\,w_{ss} + B_3\,(v_\Delta(k+j) + v_{ss}) - x_{ss} \\ &= A\,\hat{x}(k+j|k) + B_3\,v_\Delta(k+j) + 0 + (I-A)\,x_{ss} - x_{ss} = \\ &= A\,\hat{x}_\Delta(k+j|k) + B_3\,v_\Delta(k+j) \\ \hat{z}(k+j|k) &= C_2\,\hat{x}(k+j|k) + D_{21}\,\hat{e}(k+j|k) + D_{22}\,w(k+j) + D_{23}\,v(k+j) \\ &= C_2\,\hat{x}_\Delta(k+j|k) + D_{22}\,w_{ss} + D_{23}\Big(v_\Delta(k+j) + v_{ss}\Big) \\ &= C_2\,\hat{x}_\Delta(k+j|k) + D_{23}\,v_\Delta(k+j) - C_2\,x_{ss} \\ &= C_2\,\hat{x}_\Delta(k+j|k) + D_{23}\,v_\Delta(k+j) \end{aligned}$$

where we used the fact that $\hat{e}(k+j) = 0$ for $j > 0$ and (following equations 5.7 and 5.8):

$$B_2\,w_{ss} + B_3\,v_{ss} = x_{ss} - A\,x_{ss} = (I-A)\,x_{ss} \tag{5.55}$$

$$D_{22}\,w_{ss} + D_{23}\,v_{ss} = -C_2 x_{ss} \tag{5.56}$$

Substitution in the performance index leads to:

$$\begin{aligned} J(v,k) &= \sum_{j=0}^{\infty} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \\ &= \sum_{j=0}^{\infty} \Big(\hat{x}_\Delta^T(k+j|k)C_2^T C_2 \hat{x}_\Delta(k+j|k) + 2\hat{x}_\Delta^T(k+j|k)C_2^T D_{23}\,v_\Delta(k+j) \\ &\qquad + v_\Delta^T(k+j)D_{23}^T D_{23}\,v_\Delta(k+j)\Big) \end{aligned}$$

Now define

$$\bar{v}(k) = v_\Delta(k) + (D_{23}^T D_{23})^{-1} D_{23}^T C_2 \hat{x}_\Delta(k|k) \tag{5.57}$$

Then the state equation is given by:

$$\begin{aligned}
\hat{x}_\Delta(k{+}j{+}1|k) &= A\,\hat{x}_\Delta(k+j|k) + B_3\,v_\Delta(k+j)\\
&= \Big(A - B_3(D_{23}^T D_{23})^{-1} D_{23}^T C_2\Big)\hat{x}_\Delta(k+j|k) + B_3\,\bar{v}(k+j)\\
&= \bar{A}\hat{x}_\Delta(k+j|k) + B_3\,\bar{v}(k+j)
\end{aligned}$$

and the performance index becomes

$$\begin{aligned}
J(v,k) &= \sum_{j=0}^{\infty} \hat{x}_\Delta^T(k+j|k) C_2^T (I - D_{23}(D_{23}^T D_{23})^{-1} D_{23}^T) C_2 \hat{x}_\Delta(k+j|k)\\
&\quad + \bar{v}^T(k) D_{23}^T D_{23}\,\bar{v}(k)\\
&= \sum_{j=0}^{\infty} \hat{x}_\Delta^T(k+j|k)\bar{Q}\hat{x}_\Delta(k+j|k) + \bar{v}^T(k)\bar{R}\bar{v}(k)
\end{aligned}$$

where $\bar{Q} = C_2^T(I - D_{23}(D_{23}^T D_{23})^{-1} D_{23}^T)C_2$ and $\bar{R} = D_{23}^T D_{23}$. Minimizing the performance index has now becomes a standard LQG problem (Strejc, [112]), and the optimal control signal $\bar{v}$ is given by

$$\bar{v}(k) = -(B_3^T P B_3)^{-1} B_3^T P \bar{A}\hat{x}_\Delta(k|k) \tag{5.58}$$

where P is the solution of the discrete time Riccati equation

$$P = \bar{A}^T P \bar{A} - \bar{A}^T P B_3 (B_3^T P B_3 + \bar{R})^{-1} B_3^T P \bar{A} + \bar{Q}$$

The closed loop equations can now be derived by substitution:

$$\begin{aligned}
v(k) &= v_\Delta(k+j) + v_{ss}\\
&= \bar{v}(k) - (D_{23}^T D_{23})^{-1} D_{23}^T C_2 x_\Delta(k) + v_{ss}\\
&= -(B_3^T P B_3)^{-1} B_3^T P \bar{A} x_\Delta(k) - (D_{23}^T D_{23})^{-1} D_{23}^T C_2 x_\Delta(k) + v_{ss}\\
&= -(B_3^T P B_3)^{-1} B_3^T P \bar{A}\Big(x(k) - x_{ss}\Big) - (D_{23}^T D_{23})^{-1} D_{23}^T C_2\Big(x(k) - x_{ss}\Big) + v_{ss}\\
&= -\Big((B_3^T P B_3)^{-1} B_3^T P \bar{A} + (D_{23}^T D_{23})^{-1} D_{23}^T C_2\Big)x(k) +\\
&\quad + \Big[\begin{array}{cc} (B_3^T P B_3)^{-1} B_3^T P \bar{A} + (D_{23}^T D_{23})^{-1} D_{23}^T C_2 & I \end{array}\Big]\left[\begin{array}{c} x_{ss}\\ v_{ss}\end{array}\right]\\
&= -\Big((B_3^T P B_3)^{-1} B_3^T P \bar{A} + (D_{23}^T D_{23})^{-1} D_{23}^T C_2\Big)x(k) +\\
&\quad + \Big[\begin{array}{cc} (B_3^T P B_3)^{-1} B_3^T P \bar{A} + (D_{23}^T D_{23})^{-1} D_{23}^T C_2 & I \end{array}\Big]\left[\begin{array}{c} D_{ssx}\\ D_{ssv}\end{array}\right] w(k)\\
&= -F\,x(k) + D_w\,w(k) + D_e\,e(k)
\end{aligned}$$

where

$$\begin{aligned}
F &= (B_3^T P B_3)^{-1} B_3^T P \bar{A} + (D_{23}^T D_{23})^{-1} D_{23}^T C_2\\
D_w &= (B_3^T P B_3)^{-1} B_3^T P \bar{A} D_{ssx} + (D_{23}^T D_{23})^{-1} D_{23}^T C_2 D_{ssx} + D_{ssv} = F D_{ssx} + D_{ssv}\\
D_e &= 0
\end{aligned}$$

and $w_{ss} = \tilde{w}(k) = w(k)$.

The results are summarized in the following theorem:

Theorem 17
Consider system (5.1) - (5.3) with a steady-state $(v_{ss}, x_{ss}, w_{ss}, z_{ss}) = (D_{ssv}w_{ss}, D_{ssx}w_{ss}, w_{ss}, 0)$. *Further let* $w(k+j) = w_{ss}$ *for* $j > 0$. *The unconstrained infinite horizon standard predictive control problem of minimizing performance index*

$$J(v,k) = \sum_{j=0}^{\infty} \hat{z}^T(k+j|k)\hat{z}(k+j|k) \tag{5.59}$$

is solved by control law

$$v(k) \;=\; -F\,x(k) + D_w\,\tilde{w}(k) \tag{5.60}$$

where

$$\begin{aligned} F &= (B_3^T P B_3)^{-1} B_3^T P \bar{A} + (D_{23}^T D_{23})^{-1} D_{23}^T C_2 \\ D_w &= (B_3^T P B_3)^{-1} B_3^T P \bar{A} D_{ssx} + (D_{23}^T D_{23})^{-1} D_{23}^T C_2 D_{ssx} + D_{ssv} = F D_{ssx} + D_{ssv} \\ \bar{A} &= A - B_3 (D_{23}^T D_{23})^{-1} D_{23}^T C_2 \end{aligned}$$

and P *is the solution of the discrete time Riccati equation*

$$P = \bar{A}^T P \bar{A} - \bar{A}^T P B_3 (B_3^T P B_3 + D_{23}^T D_{23})^{-1} B_3^T P \bar{A} + C_2^T (I - D_{23}(D_{23}^T D_{23})^{-1} D_{23}^T) C_2$$

5.3.3 The infinite Horizon Standard Predictive Control Problem with control horizon constraint

The infinite Horizon Standard Predictive Control Problem with control horizon constraint is defined as follows:

Definition 18 *Consider a system given by the state-space realization*

$$\begin{aligned} x(k+1) &= A\,x(k) + B_1\,e(k) + B_2\,w(k) + B_3\,v(k) && (5.61) \\ y(k) &= C_1\,x(k) + D_{11}\,e(k) + D_{12}\,w(k) && (5.62) \\ z(k) &= C_2\,x(k) + D_{21}\,e(k) + D_{22}\,w(k) + D_{23}\,v(k) && (5.63) \end{aligned}$$

The goal is to find a controller $v(k) = K(\tilde{w}, y, k)$ *such that the performance index*

$$J(v,k) = \sum_{j=0}^{\infty} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \tag{5.64}$$

is minimized subject to the control horizon constraint

$$v(k+j|k) = 0 \;, \quad \text{for } j \geq N_c \tag{5.65}$$

and inequality constraints

$$\tilde{\psi}(k) = \tilde{C}_{N_s,4}x(k) + \tilde{D}_{N_s,41}e(k) + \tilde{D}_{N_s,42}\tilde{w}(k) + \tilde{D}_{N_s,43}\tilde{v}(k) \leq \tilde{\Psi}(k) \tag{5.66}$$

and for all $j \geq N_c$:

$$\psi_\infty(k) = C_5x(k+j) + D_{51}e(k+j) + D_{52}w(k+j) + D_{53}v(k+j) \leq \Psi_\infty \tag{5.67}$$

where

$$\tilde{w}(k) = \begin{bmatrix} w(k|k) \\ w(k+1|k) \\ \vdots \\ w(k+N_s-1|k) \end{bmatrix} \qquad \tilde{v}(k) = \begin{bmatrix} v(k|k) \\ v(k+1|k) \\ \vdots \\ v(k+N_s-1|k) \end{bmatrix}$$

Theorem 19

Consider system (5.61) - (5.63) and let

$$\begin{array}{rcl} N_s & < & \infty \\ \bar{\Gamma}_{N_s} & = & \mathrm{diag}\Big(\Gamma(0), \Gamma(1), \ldots, \Gamma(N_s-1)\Big) \\ \Gamma(j) & = & \Gamma_{ss} \qquad\qquad \textit{for } \ j \geq N_s \\ w(k+j) & = & w_{ss} \qquad\qquad \textit{for } \ j \geq N_s \\ \begin{bmatrix} v_{ss} \\ x_{ss} \end{bmatrix} & = & \begin{bmatrix} D_{ssv} \\ D_{ssx} \end{bmatrix} \tilde{w}(k) \\ w(k) & = & \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} \tilde{w}(k) = E_w\,\tilde{w}(k) \end{array}$$

Let A be partioned into a stable part $\bar{A}_s$ and an unstable part $\bar{A}_u$ as follows:

$$A = \begin{bmatrix} T_s & T_u \end{bmatrix} \begin{bmatrix} \bar{A}_s & 0 \\ 0 & \bar{A}_u \end{bmatrix} \begin{bmatrix} \bar{T}_s \\ \bar{T}_u \end{bmatrix} \tag{5.68}$$

where

$$\begin{bmatrix} T_s & T_u \end{bmatrix} \begin{bmatrix} \bar{T}_s \\ \bar{T}_u \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

where $\bar{A}_s$ is assumed to be a diagonal matrix. Define the vector

$$\epsilon = \Psi_\infty - (C_5 D_{ssx} + D_{52}E_w + D_{53}D_{ssv})\tilde{w}(k) \in \mathbb{R}^{n_{c5}\times 1} \tag{5.69}$$

diagonal matrix

$$E = diag(\epsilon_1, \epsilon_2, \ldots, \epsilon_{nc5}) > 0, \tag{5.70}$$

Define for any integer $m > 0$ matrix

$$W_m = \begin{bmatrix} E^{-1}C_5A \\ E^{-1}C_5A^2 \\ \vdots \\ E^{-1}C_5A^m \end{bmatrix} \tag{5.71}$$

Let constant n_E be the smallest integer such that the matrix W_{n_E} has full column-rank, and define the left inverse $W^{\ell}_{n_E}$. Let ρ_{max} be the spectral radius of diagonal matrix $\bar{A}_s$ ($\rho_{max} = \max(|\lambda_i|)$ = largest magnitude of eigenvalues). Further define

$$n_\psi = \max\left(\frac{-log\left(\|E^{-1}C_5T_s\|\|\bar{T}_sW^l_{n_E}\|\sqrt{n_E\, n_{c5}}\right)}{log(\rho_{max})}\,,\, n_E\right) \tag{5.72}$$

Let $\tilde{A}_{N_s}$, $\tilde{B}_{N_s,1}$, $\tilde{B}_{N_s,2}$, $\tilde{B}_{N_s,3}$, $\tilde{C}_{N_s,2}$, $\tilde{D}_{N_s,21}$, $\tilde{D}_{N_s,22}$ and $\tilde{D}_{N_s,23}$ be given by

$$\tilde{A}_{N_s} = \begin{bmatrix} A^{N_s} \end{bmatrix} \qquad \tilde{B}_{N_s,1} = \begin{bmatrix} A^{N_s-1}B_1 \end{bmatrix} \tag{5.73}$$

$$\tilde{B}_{N_s,2} = \begin{bmatrix} [\, A^{N_s-1}B_2 \ \cdots \ B_2\,] - D_{ssx} \end{bmatrix} \tag{5.74}$$

$$\tilde{B}_{N_s,3} = \begin{bmatrix} [\, A^{N_s-1}B_3 \ \cdots \ B_3\,] \end{bmatrix} \tag{5.75}$$

$$\tilde{C}_{N_s,2} = \begin{bmatrix} C_2 \\ C_2A \\ \vdots \\ C_2A^{N_s-1} \end{bmatrix} \qquad \tilde{D}_{N_s,21} = \begin{bmatrix} D_{21} \\ C_2B_1 \\ \vdots \\ C_2A^{N_s-2}B_1 \end{bmatrix} \tag{5.76}$$

$$\tilde{D}_{N_s,22} = \begin{bmatrix} D_{22} & 0 & \cdots & 0 \\ C_2B_2 & D_{22} & & 0 \\ \vdots & & \ddots & \vdots \\ C_2A^{N_s-2}B_2 & C_2A^{N_s-3}B_2 & \cdots & D_{22} \end{bmatrix} \tag{5.77}$$

$$\tilde{D}_{N_s,23} = \begin{bmatrix} D_{23} & 0 & \cdots & 0 \\ C_2B_3 & D_{23} & & 0 \\ \vdots & & \ddots & \vdots \\ C_2A^{N_s-2}B_3 & C_2A^{N_s-2}B_3 & \cdots & D_{23} \end{bmatrix} \tag{5.78}$$

Let Y be given by

$$Y = \begin{bmatrix} \bar{T}_u\tilde{B}_{N_s,3} \end{bmatrix}, \tag{5.79}$$

Let Y have full row-rank with a right-complement $Y^{r\perp}$ and a right-inverse Y^r. Further let $\bar{M}$ be the solution of the discrete-time lyapunov equation

$$\bar{A}_s^T\,\bar{M}\,\bar{A}_s - \bar{M} + T_s^TC_T^T\Gamma_{ss}C_TT_s = 0$$

Consider the infinite-horizon model predictive control problem, defined as minimizing 5.64 subject to an input signal $v(k+j|k)$, with $v(k+j|k)=0$ for $j \geq N_c$ and inequality constraints (5.66) and (5.67).
Define:

$$\begin{aligned}
A_\psi &= \begin{bmatrix} \tilde{D}_{N_s,43} \\ W_{n_\psi}\tilde{B}_{N_s,3} \end{bmatrix} Y^{r\perp} \\
H &= 2(Y^{r\perp})^T\Big(\tilde{B}^T_{N_s,3}\bar{T}^T_s\bar{M}\bar{T}_s\tilde{B}_{N_s,3}+\tilde{D}^T_{N_s,23}\bar{\Gamma}_{N_s}\tilde{D}_{N_s,23}\Big)Y^{r\perp} \\
b_\psi(k) &= \begin{bmatrix} \tilde{C}_{N_s,4}-\tilde{D}_{N_s,43}\tilde{F} \\ W_{n_\Psi}(\tilde{A}_{N_s}-\tilde{B}_{N_s,3}\tilde{F}) \end{bmatrix} x(k) + \begin{bmatrix} \tilde{D}_{N_s,41}+\tilde{D}_{N_s,43}\tilde{D}_e \\ W_{n_\Psi}(\tilde{B}_{N_s,1}+\tilde{B}_{N_s,3}\tilde{D}_e) \end{bmatrix} e(k) + \qquad (5.80) \\
&= \begin{bmatrix} \tilde{D}_{N_s,42}+\tilde{D}_{N_s,43}\tilde{D}_w \\ W_{n_\Psi}(\tilde{B}_{N_s,2}+\tilde{B}_{N_s,3}\tilde{D}_w) \end{bmatrix} \tilde{w}(k) - \begin{bmatrix} \tilde{\Psi}(k) \\ \underline{1} \end{bmatrix} \qquad (5.81)
\end{aligned}$$

where

$$\begin{aligned}
\tilde{F} &= -Z_1\tilde{A}_{N_s}-Z_2\tilde{C}_{N_s,2}-Z_3Y^r\bar{T}_u\tilde{A}_{N_s} \\
\tilde{D}_e &= Z_1\tilde{B}_{N_s,1}+Z_2\tilde{D}_{N_s,21}+Z_3Y^r\bar{T}_u\tilde{B}_{N_s,1} \\
\tilde{D}_w &= Z_1\tilde{B}_{N_s,2}+Z_2\tilde{D}_{N_s,22}+Z_3Y^r\bar{T}_u\tilde{B}_{N_s,2} \\
Z_1 &= -Y^r\bar{T}_u-2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{B}^T_{N_s,3}\bar{T}^T_s\bar{M}\bar{T}_s \\
Z_2 &= -2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{D}^T_{N_s,23}\bar{\Gamma}_{N_s} \\
Z_3 &= 2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{B}^T_{N_s,3}\bar{T}^T_s\bar{M}\bar{T}_s\tilde{B}_{N_s,3}+2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{D}^T_{N_s,23}\bar{\Gamma}_{N_s}\tilde{D}_{N_s,23} \\
F &= E_v\tilde{F} \quad , \quad D_e = E_v\tilde{D}_e \quad , \quad D_w = E_v\tilde{D}_w
\end{aligned}$$

and $\underline{1} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$.
The optimal control law that optimizes the inequality constrained infinite-horizon MPC problem is given by:

$$v(k) = -F\,x(k)+D_e\,e(k)+D_w\,\tilde{w}(k)+E_vY^{r\perp}\tilde{\mu}_I(k) \qquad (5.82)$$

where $\tilde{\mu}_I$ is the solution for the quadratic programming problem

$$\min_{\tilde{\mu}_I} \frac{1}{2}\tilde{\mu}_I^T\,H\,\tilde{\mu}_I$$

subject to:

$$A_\psi\tilde{\mu}_I+b_\psi(k) \leq 0$$

and $E_v = \begin{bmatrix} I & 0 & \ldots & 0 \end{bmatrix}$ *such that* $v(k)=E_v\tilde{v}(k)$.
In the absence of inequality constraints (5.89) and (5.90), the optimum $\tilde{\mu}_I = 0$ is obtained and so the control law is given analytically by:

$$v(k) = -F\,x(k)+D_e\,e(k)+D_w\,\tilde{w}(k) \qquad (5.83)$$

Theorem 19 is a special case of theorem 21 in the next subsection.

5.3.4 The infinite Horizon Standard Predictive Control Problem with structured input signals

The infinite Horizon Standard Predictive Control Problem with structured input signals is defined as follows:

Definition 20 *Consider a system given by the state-space realization*

$$\begin{aligned} x(k+1) &= A\,x(k) + B_1\,e(k) + B_2\,w(k) + B_3\,v(k) && (5.84)\\ y(k) &= C_1\,x(k) + D_{11}\,e(k) + D_{12}\,w(k) && (5.85)\\ z(k) &= C_2\,x(k) + D_{21}\,e(k) + D_{22}\,w(k) + D_{23}\,v(k) && (5.86) \end{aligned}$$

Consider the finite switching horizon N_s, then the goal is to find a controller such that the performance index

$$J(v,k) = \sum_{j=0}^{\infty} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \qquad (5.87)$$

is minimized subject to the constraints

$$\begin{aligned} \tilde{\phi}(k) &= \tilde{C}_{N_s,3}x(k) + \tilde{D}_{N_s,31}e(k) + \tilde{D}_{N_s,32}\tilde{w}(k) + \tilde{D}_{N_s,33}\tilde{v}(k)\\ &= -\mathcal{S}_v^{\ell\perp}\,\tilde{D}_{ssv}\tilde{w}(k) + \mathcal{S}_v^{\ell\perp}\,\tilde{v}(k) = 0 && (5.88)\\ \tilde{\psi}(k) &= \tilde{C}_{N_s,4}x(k) + \tilde{D}_{N_s,41}e(k) + \tilde{D}_{N_s,42}\tilde{w}(k) + \tilde{D}_{N_s,43}\tilde{v}(k) \le \tilde{\Psi}(k) && (5.89) \end{aligned}$$

and for all $j \ge N_s$:

$$\hat{\psi}_\infty(k+j|k) \le \Psi_\infty \qquad (5.90)$$

where

$$\psi_\infty(k) = C_5x(k) + D_{51}e(k) + D_{52}w(k) + D_{53}v(k) \quad ,$$

$$\tilde{w}(k) = \begin{bmatrix} w(k|k) \\ w(k+1|k) \\ \vdots \\ w(k+N_s-1|k) \end{bmatrix} \qquad \tilde{v}(k) = \begin{bmatrix} v(k|k) \\ v(k+1|k) \\ \vdots \\ v(k+N_s-1|k) \end{bmatrix}$$

and $v(k+j|k)$ beyond the switching horizon ($j \ge N_s$) is given by

$$v(k+j|k) = C_v\,A_v^{j-N_s}\,B_v\,\tilde{v}(k) + F_v(x(k+j|k) - x_{ss}) + v_{ss} \qquad j \ge N_s \quad (5.91)$$

Remark 1: The structuring of the input signal in the equations (5.91) and (5.88) is discussed in section 5.3.1.

Remark 2: Equation (5.89) covers inequality constraints up to the switching horizon, equation (5.90) covers inequality constraints beyond the switching horizon.

Theorem 21
Consider system (5.84) - (5.86) and let

$$
\begin{array}{rcl}
N_s & < & \infty \\
\bar{\Gamma}_{N_s} & = & \mathrm{diag}\Big(\Gamma(0), \Gamma(1), \ldots, \Gamma(N_s - 1)\Big) \\
\Gamma(j) & = & \Gamma_{ss} \qquad\qquad \textit{for } \; j \geq N_s \\
w(k+j) & = & w_{ss} \qquad\qquad \textit{for } \; j \geq N_s \\
\begin{bmatrix} v_{ss} \\ x_{ss} \end{bmatrix} & = & \begin{bmatrix} D_{ssv} \\ D_{ssx} \end{bmatrix} \tilde{w}(k) \\
w(k) & = & \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} \tilde{w}(k) = E_w \, \tilde{w}(k)
\end{array}
$$

Let A_T, C_T be given by

$$
\begin{aligned}
A_T &= \begin{bmatrix} A + B_3 F_v & B_3 C_v \\ 0 & A_v \end{bmatrix} \\
C_T &= \begin{bmatrix} C_2 + D_{23} \, F_v & D_{23} \, C_v \end{bmatrix}
\end{aligned}
\tag{5.92}
$$

with a partitioning of A_T into a stable part $\bar{A}_s$ and an unstable part $\bar{A}_u$ as follows:

$$
A_T = \begin{bmatrix} T_s & T_u \end{bmatrix} \begin{bmatrix} \bar{A}_s & 0 \\ 0 & \bar{A}_u \end{bmatrix} \begin{bmatrix} \bar{T}_s \\ \bar{T}_u \end{bmatrix}
\tag{5.93}
$$

where

$$
\begin{bmatrix} T_s & T_u \end{bmatrix} \begin{bmatrix} \bar{T}_s \\ \bar{T}_u \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}
$$

where $\bar{A}_s$ is assumed to be a diagonal matrix. Define the vector

$$
\epsilon = \Psi_\infty - (C_5 D_{ssx} + D_{52} E_w + D_{53} D_{ssv}) \tilde{w}(k) \in \mathbb{R}^{n_{c5} \times 1}
\tag{5.94}
$$

diagonal matrix

$$
E = \mathit{diag}(\epsilon_1, \epsilon_2, \ldots, \epsilon_{nc5}) > 0,
\tag{5.95}
$$

and

$$
C_\psi = \begin{bmatrix} C_5 + D_{53} F_v & D_{53} C_v \end{bmatrix}
\tag{5.96}
$$

Define for any integer $m > 0$ matrix

$$
W_m = \begin{bmatrix} E^{-1} C_\psi A_T \\ E^{-1} C_\psi A_T^2 \\ \vdots \\ E^{-1} C_\psi A_T^m \end{bmatrix}
\tag{5.97}
$$

Let constant n_E be the smallest integer such that the matrix W_{n_E} has full column-rank, and define the left inverse $W^{\ell}_{n_E}$. Let ρ_{max} be the spectral radius of diagonal matrix $\bar{A}_s$ ($\rho_{max} = \max(|\lambda_i|)$ = largest magnitude of eigenvalues). Further define

$$n_\psi = \max\left(\frac{-log\left(\|E^{-1}C_\psi T_s\|\|\bar{T}_s W^l_{n_E}\|\sqrt{n_E\, n_{c5}}\right)}{log(\rho_{max})}\, ,\, n_E\right) \tag{5.98}$$

Let $\tilde{A}_{N_s}$, $\tilde{B}_{N_s,1}$, $\tilde{B}_{N_s,2}$, $\tilde{B}_{N_s,3}$, $\tilde{C}_{N_s,2}$, $\tilde{D}_{N_s,21}$, $\tilde{D}_{N_s,22}$ and $\tilde{D}_{N_s,23}$ be given by

$$\tilde{A}_{N_s} = \begin{bmatrix} A^{N_s} \\ 0 \end{bmatrix} \qquad \tilde{B}_{N_s,1} = \begin{bmatrix} A^{N_s-1}B_1 \\ 0 \end{bmatrix} \tag{5.99}$$

$$\tilde{B}_{N_s,2} = \begin{bmatrix} [\, A^{N_s-1}B_2 \ \cdots \ B_2\,] - D_{ssx} \\ 0 \end{bmatrix} \tag{5.100}$$

$$\tilde{B}_{N_s,3} = \begin{bmatrix} [\, A^{N_s-1}B_3 \ \cdots \ B_3\,] \\ B_v \end{bmatrix} \tag{5.101}$$

$$\tilde{C}_{N_s,2} = \begin{bmatrix} C_2 \\ C_2A \\ \vdots \\ C_2A^{N_s-1} \end{bmatrix} \qquad \tilde{D}_{N_s,21} = \begin{bmatrix} D_{21} \\ C_2B_1 \\ \vdots \\ C_2A^{N_s-2}B_1 \end{bmatrix} \tag{5.102}$$

$$\tilde{D}_{N_s,22} = \begin{bmatrix} D_{22} & 0 & \cdots & 0 \\ C_2B_2 & D_{22} & & 0 \\ \vdots & & \ddots & \vdots \\ C_2A^{N_s-2}B_2 & C_2A^{N_s-3}B_2 & \cdots & D_{22} \end{bmatrix} \tag{5.103}$$

$$\tilde{D}_{N_s,23} = \begin{bmatrix} D_{23} & 0 & \cdots & 0 \\ C_2B_3 & D_{23} & & 0 \\ \vdots & & \ddots & \vdots \\ C_2A^{N_s-2}B_3 & C_2A^{N_s-2}B_3 & \cdots & D_{23} \end{bmatrix} \tag{5.104}$$

Let Y be given by

$$Y = \begin{bmatrix} \bar{T}_u\tilde{B}_{N_s,3} \\ \mathcal{S}_v^{\ell\perp} \end{bmatrix}, \tag{5.105}$$

Let Y have full row-rank with a right-complement $Y^{r\perp}$ and a right-inverse Y^r, partioned as

$$Y^r = \begin{bmatrix} Y_1^r & Y_2^r \end{bmatrix} \tag{5.106}$$

such that $\bar{T}_u\tilde{B}_{N_s,3}Y_1^r = I$, $\bar{T}_u\tilde{B}_{N_s,3}Y_2^r = 0$ $\mathcal{S}_v^{\ell\perp}Y_1^r = 0$ and $\mathcal{S}_v^{\ell\perp}Y_2^r = I$. Further let $\bar{M}$ be the solution of the discrete-time lyapunov equation

$$\bar{A}_s^T\,\bar{M}\,\bar{A}_s - \bar{M} + T_s^TC_T^T\Gamma_{ss}C_TT_s = 0$$

Consider the infinite-horizon model predictive control problem, defined as minimizing 5.87 subject to an input signal $v(k+j|k)$, given by (5.49) and (5.52)-(5.54) and inequality constraints (5.89) and (5.90).
Define:

$$
\begin{aligned}
A_\psi &= \begin{bmatrix} \tilde{D}_{N_s,43} \\ W_{n_\psi}\tilde{B}_{N_s,3} \end{bmatrix} Y^{r\perp} \\
H &= 2(Y^{r\perp})^T\Big(\tilde{B}^T_{N_s,3}\bar{T}^T_s\bar{M}\bar{T}_s\tilde{B}_{N_s,3} + \tilde{D}^T_{N_s,23}\bar{\Gamma}_{N_s}\tilde{D}_{N_s,23}\Big)Y^{r\perp} \\
b_\psi(k) &= \begin{bmatrix} \tilde{C}_{N_s,4} - \tilde{D}_{N_s,43}\tilde{F} \\ W_{n_\Psi}(\tilde{A}_{N_s} - \tilde{B}_{N_s,3}\tilde{F}) \end{bmatrix} x(k) + \begin{bmatrix} \tilde{D}_{N_s,41} + \tilde{D}_{N_s,43}\tilde{D}_e \\ W_{n_\Psi}(\tilde{B}_{N_s,1} + \tilde{B}_{N_s,3}\tilde{D}_e) \end{bmatrix} e(k) + \qquad (5.107) \\
&= \begin{bmatrix} \tilde{D}_{N_s,42} + \tilde{D}_{N_s,43}\tilde{D}_w \\ W_{n_\Psi}(\tilde{B}_{N_s,2} + \tilde{B}_{N_s,3}\tilde{D}_w) \end{bmatrix} \tilde{w}(k) - \begin{bmatrix} \tilde{\Psi}(k) \\ \underline{1} \end{bmatrix} \qquad (5.108)
\end{aligned}
$$

where

$$
\begin{aligned}
\tilde{F} &= -Z_1\tilde{A}_{N_s} - Z_2\tilde{C}_{N_s,2} - Z_3 Y_1^r\bar{T}_u\tilde{A}_{N_s} \\
\tilde{D}_e &= Z_1\tilde{B}_{N_s,1} + Z_2\tilde{D}_{N_s,21} + Z_3 Y_1^r\bar{T}_u\tilde{B}_{N_s,1} \\
\tilde{D}_w &= Z_1\tilde{B}_{N_s,2} + Z_2\tilde{D}_{N_s,22} + Z_3 Y_1^r\bar{T}_u\tilde{B}_{N_s,2} - (Z_3 - I)Y_2^r\mathcal{S}_v^{\ell\perp}\tilde{D}_{ssv} \\
Z_1 &= -Y_1^r\bar{T}_u - 2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{B}^T_{N_s,3}\bar{T}^T_s\bar{M}\bar{T}_s \\
Z_2 &= -2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{D}^T_{N_s,23}\bar{\Gamma}_{N_s} \\
Z_3 &= 2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{B}^T_{N_s,3}\bar{T}^T_s\bar{M}\bar{T}_s\tilde{B}_{N_s,3} + 2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{D}^T_{N_s,23}\bar{\Gamma}_{N_s}\tilde{D}_{N_s,23} \\
F &= E_v\tilde{F} \quad , \quad D_e = E_v\tilde{D}_e \quad , \quad D_w = E_v\tilde{D}_w
\end{aligned}
$$

and $\underline{1} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$.
The optimal control law that optimizes the inequality constrained infinite-horizon MPC problem is given by:

$$v(k) \;=\; -F\,x(k) + D_e\,e(k) + D_w\,\tilde{w}(k) + E_v Y^{r\perp}\tilde{\mu}_I(k) \qquad (5.109)$$

where $\tilde{\mu}_I$ is the solution for the quadratic programming problem

$$\min_{\tilde{\mu}_I} \frac{1}{2}\tilde{\mu}_I^T\, H\, \tilde{\mu}_I$$

subject to:

$$A_\psi\tilde{\mu}_I + b_\psi(k) \le 0$$

and $E_v = \begin{bmatrix} I & 0 & \ldots & 0 \end{bmatrix}$ *such that* $v(k) = E_v\tilde{v}(k)$.

In the absence of inequality constraints (5.89) and (5.90), the optimum $\tilde{\mu}_I = 0$ is obtained and so the control law is given analytically by:

$$v(k) \;=\; -F\,x(k) + D_e\,e(k) + D_w\,\tilde{w}(k) \qquad (5.110)$$

Remark: Note that if for any $i = 1, \ldots, n_{c5}$ there holds:

$$[\epsilon]_i \leq 0$$

the i-th constraint

$$[\psi_\infty(k+j)]_i \leq [\Psi_\infty]_i$$

cannot be satisfied for $j \leftarrow \infty$ and the infinite horizon predictive control problem is infeasible. A necessary condition for feasibility is therefore

$$E = \text{diag}(\epsilon_1, \epsilon_2, \ldots, \epsilon_{nc5}) > 0$$

Proof of theorem 21:

The performance index can be split in two parts:

$$J(v,k) \;=\; \sum_{j=0}^{\infty} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) = J_1(v,k) + J_2(v,k)$$

where

$$\begin{aligned} J_1(v,k) &= \sum_{j=0}^{N_s-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) \\ J_2(v,k) &= \sum_{j=N_s}^{\infty} \hat{z}^T(k+j|k)\Gamma_{ss}\hat{z}(k+j|k) \end{aligned}$$

For technical reasons we will consider the derivation of criterion J_2 before we derive criterion J_1.

Derivation of J_2:
Consider system (5.84) - (5.86) with structured input signal (5.52), (5.53) and (5.54) and constraint (5.49). Define for $j \geq N_s$:

$$\begin{aligned} \hat{x}_\Delta(k+j) &= \hat{x}(k+j) - x_{ss} \\ v_\Delta(k+j) &= v(k+j) - v_{ss} \end{aligned}$$

Then, using the fact that $w(k+j) = w_{ss}$ and $\hat{e}(k+j) = 0$ for $j \geq N_s$, it follows for $j \geq N_s$:

$$\begin{aligned} \hat{x}_\Delta(k+j+1|k) &= \hat{x}(k+j+1|k) - x_{ss} \\ &= A\,\hat{x}(k+j|k) + B_1\,\hat{e}(k+j|k) + B_2\,w(k+j|k) + B_3\,v(k+j) - x_{ss} \\ &= A\,\hat{x}(k+j|k) + B_1\,0 + B_2\,w_{ss} + B_3\,v_{ss} + B_3\,v_\Delta(k+j) - x_{ss} \\ &= A\,\hat{x}(k+j|k) + 0 + (I-A)\,x_{ss} - x_{ss} + B_3\,C_v\,x_v(k+j) + \\ &\quad +B_3\,F_v\,\hat{x}_\Delta(k+j) \\ &= (A + B_3F_v)\,\hat{x}_\Delta(k+j|k) + B_3\,C_v\,x_v(k+j) \end{aligned}$$

$$\begin{aligned}
\hat{z}(k+j|k) &= C_2\,\hat{x}(k+j|k) + D_{21}\,\hat{e}(k+j|k) + D_{22}\,w(k+j) + D_{23}\,v(k+j)\\
&= C_2\,\hat{x}_\Delta(k+j|k) + C_2\,x_{ss} + D_{22}\,w_{ss} + D_{23}\,v_{ss} + D_{23}\,v_\Delta(k+j)\\
&= C_2\,\hat{x}_\Delta(k+j|k) + 0 + D_{23}\,C_v\,x_v(k+j) +\\
&\quad +D_{23}\,F_v\,\hat{x}_\Delta(k+j)\\
&= C_2\,\hat{x}_\Delta(k+j|k) + D_{23}\,C_v\,x_v(k+j) + D_{23}\,F_v\,\hat{x}_\Delta(k+j)
\end{aligned}$$

Define

$$\hat{x}_T(k+j|k) = \begin{bmatrix} \hat{x}_\Delta(k+j|k) \\ x_v(k+j) \end{bmatrix}$$

Then, using A_T and C_T from (5.92), we find for $j \geq N_s$:

$$\begin{aligned}
\hat{x}_T(k+j+1|k) &= A_T\hat{x}_T(k+j|k)\\
\hat{z}(k+j|k) &= C_T\hat{x}_T(k+j|k)
\end{aligned}$$

which is an autonomous system. This means that $\hat{z}(k+j|k)$ can be computed for all $j \geq N_s$ if initial state $\hat{x}_T(k+N_s|k)$ is given, so if $\hat{x}(k+N_s)$ and $x_v(k+N_s)$ are known. The state $x_v(k+N_s)$ is given by equation (5.52), the state $\hat{x}(k+N_s|k)$ can be found by successive substitution:

$$\begin{aligned}
\hat{x}(k+N_s|k) &= A^{N_s}\,x(k) + A^{N_s-1}B_1\,e(k) + \begin{bmatrix} A^{N_s-1}B_2 & \cdots & B_2 \end{bmatrix}\tilde{w}(k) +\\
&\quad +\begin{bmatrix} A^{N_s-1}B_3 & \cdots & B_3 \end{bmatrix}\tilde{v}(k)
\end{aligned}$$

together with $x_v(k+N_s|k) = B_v\,\tilde{v}(k)$ we obtain

$$\begin{aligned}
\hat{x}_T(k+N_s|k) &= \begin{bmatrix} \hat{x}(k+N_s|k) - x_{ss} \\ x_v(k+N_s) \end{bmatrix}\\
&= \tilde{A}_{N_s}\,x(k) + \tilde{B}_{N_s,1}\,e(k) + \tilde{B}_{N_s,2}\,\tilde{w}(k) + \tilde{B}_{N_s,3}\,\tilde{v}(k)
\end{aligned}$$

where $\tilde{A}_{N_s}$, $\tilde{B}_{N_s,1}$, $\tilde{B}_{N_s,2}$ and $\tilde{B}_{N_s,3}$ as defined in (5.99)-(5.101), and $x_{ss} = D_{ssx}\tilde{w}(k)$.
The prediction $\hat{z}(k+j|k)$ for $j \geq N_s$ is obtained by successive substitution, resulting in:

$$\hat{z}(k+j|k) = C_T\,A_T^{j-N_s}\hat{x}_T(k+N_s|k) \tag{5.111}$$

Now a problem occurs when the matrix A_T contains unstable eigenvalues ($|\lambda_i| \geq 1$). For $j \longrightarrow \infty$, the matrix $A_T^{j-N_s}$ will become unbounded, and so will $\hat{z}$. The only way to solve this problem is to make sure that the part of the state $\hat{x}_T(k+N_s)$, related to the unstable poles is zero (Rawlings & Muske, [96]). Make a partitioning of A_T into a stable part $\bar{A}_s$ and an unstable part $\bar{A}_u$ as in (5.93), then (5.111) becomes:

$$\begin{aligned}
\hat{z}(k+j|k) &= C_T\begin{bmatrix} T_s & T_u \end{bmatrix}\begin{bmatrix} \bar{A}_s^{j-N_s} & 0 \\ 0 & \bar{A}_u^{j-N_s} \end{bmatrix}\begin{bmatrix} \bar{T}_s \\ \bar{T}_u \end{bmatrix}\hat{x}_T(k+N_s|k)\\
&= C_T\,T_s\bar{A}_s^{j-N_s}\bar{T}_s\hat{x}_T(k+N_s|k) + C_T\,T_u\bar{A}_u^{j-N_s}\bar{T}_u\hat{x}_T(k+N_s|k)
\end{aligned} \tag{5.112}$$

The matrix $\bar{A}_u^{j-N_s}$ will grow unbounded for $j \to \infty$. The prediction $\hat{z}(k+j|k)$ for $j \to \infty$ will only remain bounded if $\bar{T}_u \hat{x}_T(k+N_s|k) = 0$.
In order to guarantee closed-loop stability for systems with unstable modes in A_T, we need to satisfy equality constraint

$$\bar{T}_u \hat{x}_T(k+N_s|k) \;=\; \bar{T}_u\Big(\tilde{A}_{N_s}\, x(k) + \tilde{B}_{N_s,1}\, e(k) + \tilde{B}_{N_s,2}\, \tilde{w}(k) + \tilde{B}_{N_s,3}\, \tilde{v}(k)\Big) = 0$$

Together with the equality constraint (5.88) we obtain the final constraint:

$$\begin{bmatrix} \bar{T}_u \tilde{A}_{N_s} \\ 0 \end{bmatrix} x(k) + \begin{bmatrix} \bar{T}_u \tilde{B}_{N_s,1} \\ 0 \end{bmatrix} e(k) + \begin{bmatrix} \bar{T}_u \tilde{B}_{N_s,2} \\ -\mathcal{S}_v^{\ell\perp} \tilde{D}_{ssv} \end{bmatrix} \tilde{w}(k) + \begin{bmatrix} \bar{T}_u \tilde{B}_{N_s,3} \\ \mathcal{S}_v^{\ell\perp} \end{bmatrix} \tilde{v}(k) \;=\; 0$$

Consider Y as given in (5.105) with right-inverse $Y^r = \begin{bmatrix} Y_1^r & Y_2^r \end{bmatrix}$ and right-complement $Y^{r\perp}$, then the set of all signals $\tilde{v}(k)$ satisfying the equality constraints is given by

$$\tilde{v}(k) \;=\; \tilde{v}_E(k) + Y^{r\perp}\tilde{\mu}(k) \tag{5.113}$$

where $\tilde{v}_E$ is given by

$$\begin{aligned} \tilde{v}_E(k) &= -Y^r\left(\begin{bmatrix} \bar{T}_u \tilde{A}_{N_s} \\ 0 \end{bmatrix} x(k) + \begin{bmatrix} \bar{T}_u \tilde{B}_{N_s,1} \\ 0 \end{bmatrix} e(k) + \begin{bmatrix} \bar{T}_u \tilde{B}_{N_s,2} \\ -\mathcal{S}_v^{\ell\perp} \tilde{D}_{ssv} \end{bmatrix} \tilde{w}(k)\right) \\ &= -Y_1^r \bar{T}_u\Big(\tilde{A}_{N_s}\, x(k) + \tilde{B}_{N_s,1}\, e(k) + \tilde{B}_{N_s,2}\, \tilde{w}(k)\Big) + Y_2^r \bar{T}_u \mathcal{S}_v^{\ell\perp} \tilde{D}_{ssv}\, \tilde{w}(k) \end{aligned}$$

and $\tilde{\mu}(k)$ is a free vector with the appropriate dimensions.
If the equality $\bar{T}_u \hat{x}_T(k+N_s|k) = 0$ is satisfied, the prediction becomes:

$$\hat{z}(k+j|k) \;=\; C_T \begin{bmatrix} T_s & T_u \end{bmatrix} \begin{bmatrix} \bar{A}_s^{j-N_s} & 0 \\ 0 & \bar{A}_u^{j-N_s} \end{bmatrix} \begin{bmatrix} \bar{T}_s \hat{x}_T(k+N_s|k) \\ 0 \end{bmatrix} = \tag{5.114}$$

$$\;=\; C_T\, T_s \bar{A}_s^{j-N_s} \bar{T}_s \hat{x}_T(k+N_s) \tag{5.115}$$

Define

$$\bar{M} \;=\; \sum_{j=N_s}^{\infty} (\bar{A}_s^T)^{j-N_s} T_s^T C_T^T \Gamma_{ss} C_T\, T_s (\bar{A}_s)^{j-N_s} \tag{5.116}$$

then

$$\begin{aligned} J_2(v,k) &= \sum_{j=N_s}^{\infty} \hat{z}^T(k+j|k)\Gamma_{ss}\hat{z}(k+j|k) \\ &= \sum_{j=N_s}^{\infty} \hat{x}_T^T(k+N_s|k)\bar{T}_s^T(\bar{A}_s^T)^{j-N_s} T_s^T C_T^T \Gamma_{ss} C_T\, T_s(\bar{A}_s)^{j-N_s}\bar{T}_s\hat{x}_T(k+N_s|k) \\ &= \hat{x}_T^T(k+N_s|k)\bar{T}_s^T \bar{M} \bar{T}_s \hat{x}_T(k+N_s|k) \\ &= \frac{1}{2}\tilde{\mu}^T(k) H_2 \tilde{\mu}(k) + \tilde{\mu}^T(k) f_2(k) + c_2(k) \end{aligned}$$

where

$$\begin{aligned}
H_2 &= 2(Y^{r\perp})^T \tilde{B}^T_{N_s,3}\bar{T}^T_s \bar{M}\bar{T}_s\tilde{B}_{N_s,3}Y^{r\perp}\\
f_2(k) &= 2(Y^{r\perp})^T \tilde{B}^T_{N_s,3}\bar{T}^T_s \bar{M}\bar{T}_s\hat{x}_{T,E}(k+N_s|k)\\
c_2(k) &= \hat{x}^T_{T,E}(k+N_s|k)\bar{T}^T_s \bar{M}\bar{T}_s\hat{x}_{T,E}(k+N_s|k)\\
\hat{x}_{T,E}(k+N_s|k) &= \tilde{A}_{N_s}\,x(k) + \tilde{B}_{N_s,1}\,e(k) + \tilde{B}_{N_s,2}\,\tilde{w}(k) + \tilde{B}_{N_s,3}\,\tilde{v}_E(k)
\end{aligned}$$

The matrix $\bar{M}$ can be computed as the solution of the discrete time Lyapunov equation

$$\bar{A}^T_s\,\bar{M}\,\bar{A}_s - \bar{M} + T^T_s C^T_T \Gamma_{ss} C_T T_s = 0$$

Derivation of J_1:
The expression of $J_1(v,k)$ is equivalent to the expression of the performance index $J(v,k)$ for a finite prediction horizon. The performance signal vector $\tilde{z}(k)$ is defined for switching horizon N_s:

$$\tilde{z}(k) = \begin{bmatrix} \hat{z}(k|k) \\ \hat{z}(k+1|k) \\ \dots \\ \hat{z}(k+N_s-1|k) \end{bmatrix}$$

Using the results of chapter 3 we derive:

$$\begin{aligned}
\tilde{z}(k) &= \tilde{C}_{N_s,2}\,x(k) + \tilde{D}_{N_s,21}\,e(k) + \tilde{D}_{N_s,22}\,\tilde{w}(k) + \tilde{D}_{N_s,23}\,\tilde{v}(k)\\
&= \tilde{C}_{N_s,2}\,x(k) + \tilde{D}_{N_s,21}\,e(k) + \tilde{D}_{N_s,22}\,\tilde{w}(k) + \tilde{D}_{N_s,23}\left(\tilde{v}_E(k) + Y^{r\perp}\tilde{\mu}(k)\right)
\end{aligned}$$

where $\tilde{C}_{N_s,2}$, $\tilde{D}_{N_s,21}$, $\tilde{D}_{N_s,22}$ and $\tilde{D}_{N_s,23}$ are given in (5.102)-(5.104). Now J_1 becomes

$$\begin{aligned}
J_1(v,k) &= \sum_{j=0}^{N_s-1} \hat{z}^T(k+j|k)\Gamma(j)\hat{z}(k+j|k) =\\
&= \tilde{z}^T(k)\bar{\Gamma}_{N_s}\tilde{z}(k) =\\
&= \frac{1}{2}\tilde{\mu}^T(k)H_1\tilde{\mu}(k) + \tilde{\mu}^T(k)f_1(k) + c_1(k)
\end{aligned}$$

where

$$\begin{aligned}
H_1 &= 2(Y^{r\perp})^T \tilde{D}^T_{N_s,23}\bar{\Gamma}_{N_s}\tilde{D}_{N_s,23}Y^{r\perp}\\
f_1(k) &= 2(Y^{r\perp})^T \tilde{D}^T_{N_s,23}\bar{\Gamma}_{N_s}\tilde{z}_E(k)\\
c_1(k) &= \tilde{z}^T_E(k)\bar{\Gamma}_{N_s}\tilde{z}_E(k)\\
\tilde{z}_E(k) &= \tilde{C}_{N_s,2}\,x(k) + \tilde{D}_{N_s,21}\,e(k) + \tilde{D}_{N_s,22}\,\tilde{w}(k) + \tilde{D}_{N_s,23}\tilde{v}_E(k)
\end{aligned}$$

Minimization of $J_1 + J_2$:
Combining the results we obtain the problem of minimizing

$$J_1 + J_2 = \frac{1}{2}\tilde{\mu}^T(k)(H_1+H_2)\tilde{\mu}(k) + \tilde{\mu}^T(k)(f_1(k)+f_2(k)) + c_1(k) + c_2(k) \tag{5.117}$$

$$= \frac{1}{2}\tilde{\mu}^T(k)\,H\,\tilde{\mu}(k) + \tilde{\mu}^T(k)\,f(k) + c(k) \tag{5.118}$$

Now define:

$$\hat{x}_{T,0}(k+N_s|k) = \tilde{A}_{N_s}\,x(k) + \tilde{B}_{N_s,1}\,e(k) + \tilde{B}_{N_s,2}\,\tilde{w}(k) \qquad (5.119)$$

$$\tilde{z}_0(k) = \tilde{C}_{N_s,2}\,x(k) + \tilde{D}_{N_s,21}\,e(k) + \tilde{D}_{N_s,22}\,\tilde{w}(k) \qquad (5.120)$$

$$\tilde{\phi}_0(k) = -\mathcal{S}_v^{\ell\perp}\tilde{D}_{ssv}\,\tilde{w}(k) \qquad (5.121)$$

or

$$\begin{bmatrix} \hat{x}_{T,0}(k+N_s|k) \\ \tilde{z}_0(k) \\ \tilde{\phi}_0(k) \end{bmatrix} = \begin{bmatrix} \tilde{A}_{N_s} \\ \tilde{C}_{N_s,2} \\ 0 \end{bmatrix} x(k) + \begin{bmatrix} \tilde{B}_{N_s,1} \\ \tilde{D}_{N_s,21} \\ 0 \end{bmatrix} e(k) + \begin{bmatrix} \tilde{B}_{N_s,2} \\ \tilde{D}_{N_s,22} \\ -\mathcal{S}_v^{\ell\perp}\tilde{D}_{ssv} \end{bmatrix} \tilde{w}(k) \qquad (5.122)$$

Then:

$$\begin{aligned} \tilde{v}_E(k) &= -Y_1^r\bar{T}_u x_{T,0}(k) - Y_2^r\tilde{\phi}_0(k) \\ \hat{x}_{T,E}(k+N_s|k) &= x_{T,0}(k) + \tilde{B}_{N_s,3}\tilde{v}_E(k) \\ &= x_{T,0}(k) - \tilde{B}_{N_s,3}Y_1^r\bar{T}_u x_{T,0}(k) - \tilde{B}_{N_s,3}Y_2^r\tilde{\phi}_0(k) \\ \tilde{z}_E(k) &= \tilde{z}_0(k) + \tilde{D}_{N_s,23}\tilde{v}_E(k) \\ &= \tilde{z}_0(k) - \tilde{D}_{N_s,23}Y_1^r\bar{T}_u x_{T,0}(k) - \tilde{D}_{N_s,23}Y_2^r\tilde{\phi}_0(k) \end{aligned}$$

and thus:

$$\begin{bmatrix} \tilde{v}_E(k) \\ \hat{x}_{T,E}(k+N_s|k) \\ \tilde{z}_E(k) \end{bmatrix} = \begin{bmatrix} -Y_1^r\bar{T}_u & 0 & -Y_2^r \\ (I-\tilde{B}_{N_s,3}Y_1^r\bar{T}_u) & 0 & -\tilde{B}_{N_s,3}Y_2^r \\ -\tilde{D}_{N_s,23}Y_1^r\bar{T}_u & I & -\tilde{D}_{N_s,23}Y_2^r \end{bmatrix} \begin{bmatrix} \hat{x}_{T,0}(k+N_s|k) \\ \tilde{z}_0(k) \\ \tilde{\phi}_0(k) \end{bmatrix} \qquad (5.123)$$

Collecting the results from equations (5.119)-(5.123) we obtain:

$$\begin{aligned} f(k) &= f_1(k) + f_2(k) \qquad (5.124) \\ &= 2(Y^{r\perp})^T\tilde{D}_{N_s,23}^T\bar{\Gamma}_{N_s}\tilde{z}_E(k) + 2(Y^{r\perp})^T\tilde{B}_{N_s,3}^T\bar{T}_s^T\bar{M}\bar{T}_s\hat{x}_{T,E}(k+N_s|k) \\ &= \begin{bmatrix} 0 & 2(Y^{r\perp})^T\tilde{B}_{N_s,3}^T\bar{T}_s^T\bar{M}\bar{T}_s & 2(Y^{r\perp})^T\tilde{D}_{N_s,23}^T\bar{\Gamma}_{N_s} \end{bmatrix} \begin{bmatrix} \tilde{v}_E(k) \\ \hat{x}_{T,E}(k+N_s|k) \\ \tilde{z}_E(k) \end{bmatrix} \qquad (5.125) \end{aligned}$$

In the absence of inequality constraints we obtain the unconstrained minimization of (5.118). The minimum $\tilde{\mu} = \tilde{\mu}_E$ is found for

$$\tilde{\mu}_E(k) = -(H_1 + H_2)^{-1}(f_1(k) + f_2(k)) = -H^{-1}f(k)$$

and so

$$\begin{aligned} \tilde{v}(k) &= \tilde{v}_E(k) + Y^{r\perp}\tilde{\mu}_E(k) \\ &= \tilde{v}_E(k) - Y^{r\perp}H^{-1}f(k) \end{aligned}$$

$$
\begin{aligned}
&= \tilde{v}_E(k) - 2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{D}^T_{N_s,23}\bar{\Gamma}_{N_s}\tilde{z}_E(k) - 2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{B}^T_{N_s,3}\bar{T}^T_s\bar{M}\bar{T}_s\hat{x}_{T,E}(k+N_s|k) \\
&= \begin{bmatrix} I & -2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{B}^T_{N_s,3}\bar{T}^T_s\bar{M}\bar{T}_s & -2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{D}^T_{N_s,23}\bar{\Gamma}_{N_s} \end{bmatrix}\begin{bmatrix} \tilde{v}_E(k) \\ \hat{x}_{T,E}(k+N_s|k) \\ \tilde{z}_E(k) \end{bmatrix} \\
&= \begin{bmatrix} I & -2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{B}^T_{N_s,3}\bar{T}^T_s\bar{M}\bar{T}_s & -2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{D}^T_{N_s,23}\bar{\Gamma}_{N_s} \end{bmatrix}\cdot \\
&\quad\cdot \begin{bmatrix} -Y^r_1\bar{T}_u & 0 & -Y^r_2 \\ (I-\tilde{B}_{N_s,3}Y^r_1\bar{T}_u) & 0 & -\tilde{B}_{N_s,3}Y^r_2 \\ -\tilde{D}_{N_s,23}Y^r_1\bar{T}_u & I & -\tilde{D}_{N_s,23}Y^r_2 \end{bmatrix}\begin{bmatrix} \hat{x}_{T,0}(k+N_s|k) \\ \tilde{z}_0(k) \\ \tilde{\phi}_0(k) \end{bmatrix} \\
&= \begin{bmatrix} I & -2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{B}^T_{N_s,3}\bar{T}^T_s\bar{M}\bar{T}_s & -2Y^{r\perp}H^{-1}(Y^{r\perp})^T\tilde{D}^T_{N_s,23}\bar{\Gamma}_{N_s} \end{bmatrix}\cdot \\
&\quad\cdot \begin{bmatrix} -Y^r_1\bar{T}_u & 0 & -Y^r_2 \\ (I-\tilde{B}_{N_s,3}Y^r_1\bar{T}_u) & 0 & -\tilde{B}_{N_s,3}Y^r_2 \\ -\tilde{D}_{N_s,23}Y^r_1\bar{T}_u & I & -\tilde{D}_{N_s,23}Y^r_2 \end{bmatrix}\cdot \\
&\quad\cdot \left(\begin{bmatrix} \tilde{A}_{N_s} \\ \tilde{C}_{N_s,2} \\ 0 \end{bmatrix} x(k) + \begin{bmatrix} \tilde{B}_{N_s,1} \\ \tilde{D}_{N_s,21} \\ 0 \end{bmatrix} e(k) + \begin{bmatrix} \tilde{B}_{N_s,2} \\ \tilde{D}_{N_s,22} \\ -\mathcal{S}^{\ell\perp}_v\tilde{D}_{ssv} \end{bmatrix}\tilde{w}(k) \right) \\
&= \begin{bmatrix} Z_1 + Z_3Y^r_1\bar{T}_u & Z_2 & (Z_3-I)Y^r_2 \end{bmatrix}\cdot \\
&\quad\cdot \left(\begin{bmatrix} \tilde{A}_{N_s} \\ \tilde{C}_{N_s,2} \\ 0 \end{bmatrix} x(k) + \begin{bmatrix} \tilde{B}_{N_s,1} \\ \tilde{D}_{N_s,21} \\ 0 \end{bmatrix} e(k) + \begin{bmatrix} \tilde{B}_{N_s,2} \\ \tilde{D}_{N_s,22} \\ -\mathcal{S}^{\ell\perp}_v\tilde{D}_{ssv} \end{bmatrix}\tilde{w}(k) \right) \\
&= -\tilde{F}x(k) + \tilde{D}_e e(k) + \tilde{D}_w\tilde{w}(k)
\end{aligned}
$$

The optimal input becomes

$$
\begin{aligned}
v(k) &= E_v\tilde{v}(k) \\
&= -E_v\tilde{F}x(k) + E_v\tilde{D}_e e(k) + E_v\tilde{D}_w\tilde{w}(k) \\
&= -Fx(k) + D_e e(k) + D_w\tilde{w}(k)
\end{aligned}
$$

which is equal to control law (5.110).

Inequality constraints:
Now we will concentrate on the inequality constraints. The following derivation is an extension of the result, given in Rawlings & Muske [96]. Consider the inequality constraints:

$$
\tilde{\psi}(k) = \tilde{C}_{N_s,4}x(k) + \tilde{D}_{N_s,41}e(k) + \tilde{D}_{N_s,42}\tilde{w}(k) + \tilde{D}_{N_s,43}\tilde{v}(k) \le \tilde{\Psi}(k)
$$

where we assume that $\tilde{\psi}(k)$ has finite dimension. To allow inequality constraints beyond the switching horizon N_s we have introduced an extra inequality constraint

$$
\psi_\infty(k+j) \le \Psi_\infty \qquad , \; j \ge N_s
$$

where $\Psi_\infty \in \mathbb{R}^{n_{c5}\times 1}$ is a constant vector and $\psi_\infty(k)$ is given by

$$
\psi_\infty(k) \;=\; C_5x(k) + D_{51}e(k) + D_{52}w(k) + D_{53}v(k)
$$

Define ϵ, E, C_ψ and W_m according to (5.94), (5.95), (5.96) and (5.97). Let constant n_E be the smallest integer such that the matrix W_{n_E} has full column-rank, and define the left inverse $W^l_{n_E} = (W^T_{n_E} W_{n_E})^{-1} W^T_{n_E}$. Let ρ_{max} be the spectral radius of diagonal matrix $\bar{A}_s$ ($\rho_{max} = \max(|\lambda_i|)$ = largest magnitude of eigenvalues). Further let n_ψ be defined as in theorem 21 and let

$$\psi_\infty(k+N_s+j|k) \leq \Psi_\infty \quad \text{for} \quad j = 1, \ldots, n_\psi \tag{5.126}$$

We derive for $j > 0$:

$$\begin{aligned}
\hat{\psi}_\infty(k+N_s+j|k) &= C_5\hat{x}(k+N_s+j|k) + D_{52}w(k+N_s+j|k) + D_{53}v(k+N_s+j|k) \\
&= C_5\hat{x}_T(k+N_s+j|k) + D_{53}C_v\, x_v(k+N_s+j|k)) + \\
&\quad +D_{53}F_v(x(k+N_s+j|k) - x_{ss})C_5x_{ss} + D_{52}w_{ss} + D_{53}v_{ss} \\
&= (C_5 + D_{53}F_v)\hat{x}_\Delta(k+N_s+j|k) + D_{53}C_v\, x_v(k+N_s+j) + \\
&\quad +(C_5D_{ssx} + D_{52} + D_{53}D_{ssv})w_{ss} \\
&= C_\psi\hat{x}_T(k+N_s+j|k) + (C_5D_{ssx} + D_{52} + D_{53}D_{ssv})w_{ss}
\end{aligned}$$

From equation (5.126) we know that for $j = 1, \ldots, n_\psi$

$$\hat{\psi}_\infty(k+N_s+j|k) - (C_5D_{ssx} + D_{52} + D_{53}D_{ssv})w_{ss} < \Psi_\infty - (C_5D_{ssx} + D_{52} + D_{53}D_{ssv})w_{ss}$$

and so

$$C_\psi\hat{x}_T(k+N_s+j|k) \leq \epsilon$$

Multiplied by E^{-1} we obtain for $j = 1, \ldots, n_\psi$:

$$E^{-1}\hat{\psi}_\infty(k+N_s+j|k) = E^{-1}C_\psi\hat{x}_T(k+N_s+j|k) \leq 1$$

For $n_E \leq n_\psi$ we find that

$$E^{-1}C_\psi\hat{x}_T(k+N_s+j|k) \leq 1 \qquad j = 1, \ldots, n_E$$

or

$$\begin{aligned}
&\begin{bmatrix} E^{-1}\psi_\infty(k+N_s+1|k) \\ E^{-1}\psi_\infty(k+N_s+2|k) \\ \vdots \\ E^{-1}\psi_\infty(k+N_s+n_E|k) \end{bmatrix} = \\
&\quad = \begin{bmatrix} E^{-1}C_\psi A_T \\ E^{-1}C_\psi A_T^2 \\ \vdots \\ E^{-1}C_\psi A_T^{n_E} \end{bmatrix} \hat{x}_T(k+N_s|k) \\
&\quad = W_{n_E}\hat{x}_T(k+N_s|k) \leq 1
\end{aligned}$$

The signal $W_{n_E}\hat{x}_T(k+N_s|k)$ is a $n_E\, n_{c5} \times 1$ vector and so a bound on the Euclidean norm is given by:

$$\|W_{n_E}\hat{x}_T(k+N_s|k)\| \leq \sqrt{n_E\, n_{c5}}$$

Using equation (5.98) we find for $j \geq n_\psi + 1$:

$$\rho_{max}^{j} < \rho_{max}^{n_\psi} \leq 1/\left(\|E^{-1}C_\psi T_s\|\|\bar{T}_s W_{n_E}^{l}\|\sqrt{n_E\, n_{c5}}\right)$$

and for the norm of the diagonal matrix $\bar{A}_s^j$ we find

$$\|\bar{A}_s^j\| \leq \rho_{max}^{j} < \rho_{max}^{n_\psi}$$

Combining the results give for $j \geq n_\psi + 1$:

$$\begin{aligned}
&\|E^{-1}C_\psi \hat{x}_T(k+j+N_s|k)\| = \\
&\quad = \|E^{-1}C_\psi A^j \hat{x}_T(k|k)\| \\
&\quad = \|E^{-1}C_\psi T_s \bar{A}_s^j \bar{T}_s W_{n_E}^{l} W_{n_E}\hat{x}_T(k+N_s|k)\| \\
&\quad \leq \|E^{-1}C_\psi T_s\| \cdot \|\bar{A}_s^j\| \cdot \|\bar{T}_s W_{n_E}^{l}\| \cdot \|W_{n_E}\hat{x}_T(k+N_s|k)\| \\
&\quad < \|E^{-1}C_\psi T_s\| \cdot \rho_{max}^{n_\psi} \cdot \|\bar{T}_s W_{n_E}^{l}\| \cdot \sqrt{n_E\, n_{c5}} \\
&\quad = 1
\end{aligned}$$

and so starting from (5.126) we derived

$$\psi_\infty(k+N_s+j|k) \leq \Psi_\infty \quad \text{for } j = n_\psi+1,\ldots,\infty \tag{5.127}$$

This means that (5.126) implies (5.127). Equation (5.126) can be rewritten as

$$W_{n_\Psi}\, \hat{x}_T(k+N_s|k) \leq \underline{1}$$

Together with constraint (5.89) this results in the overall inequality constraint

$$\begin{bmatrix} \tilde{\psi}(k) - \tilde{\Psi}(k) \\ W_{n_\Psi}\, \hat{x}_T(k+N_s|k) - \underline{1} \end{bmatrix} \leq 0$$

Consider performance index (5.118) in which the optimization vector $\tilde{\mu}(k)$ is written as

$$\begin{aligned}
\tilde{\mu}(k) &= \tilde{\mu}_E(k) + \tilde{\mu}_I(k) && (5.128)\\
&= -(H_1+H_2)^{-1}(f_1(k)+f_2(k)) + \tilde{\mu}_I(k) && (5.129)\\
&= -H^{-1}(f(k)) + \tilde{\mu}_I(k) && (5.130)\\
& && (5.131)
\end{aligned}$$

where $\tilde{\mu}_E(k)$ is the equality constrained solution given in the previous section, and $\tilde{\mu}_i(k)$ is an additional term to take the inequality constraints into account. The performance index now becomes

$$\begin{aligned}
&\frac{1}{2}\tilde{\mu}^T(k)\,H\,\tilde{\mu}(k) + f^T(k)\tilde{\mu}(k) + c(k) = \\
&\quad = \frac{1}{2}\Big(- H^{-1}f(k) + \tilde{\mu}_I(k)\Big)^T H\,\Big(- H^{-1}f(k) + \tilde{\mu}_I(k)\Big) \\
&\qquad + f^T(k)\Big(- H^{-1}f(k) + \tilde{\mu}_I(k)\Big) + c(k) = \\
&\quad = \frac{1}{2}\tilde{\mu}_I^T(k)\,H\,\tilde{\mu}_I(k) - f^T(k)H^{-1}H\,\tilde{\mu}_I(k) \\
&\qquad + \frac{1}{2}f^T(k)H^{-1}\,H\,H^{-1}f(k) - f^T(k)H^{-1}f(k) + f^T(k)\tilde{\mu}_I(k) + c(k) \\
&\quad = \frac{1}{2}\tilde{\mu}_I^T(k)\,H\,\tilde{\mu}_I(k) - \frac{1}{2}f^T(k)H^{-1}f(k) + c(k) \\
&\quad = \frac{1}{2}\tilde{\mu}_I^T(k)\,H\,\tilde{\mu}_I(k) + c'(k)
\end{aligned} \tag{5.132}$$

Now consider the input signal

$$\begin{aligned}
\tilde{v}(k) &= \tilde{v}_E(k) + Y^{r\perp}\tilde{\mu}(k) \\
&= \tilde{v}_E(k) - Y^{r\perp}H^{-1}f(k) + Y^{r\perp}\tilde{\mu}_I(k) \\
&= \tilde{v}_I(k) + Y^{r\perp}\tilde{\mu}_I(k)
\end{aligned}$$

where

$$\tilde{v}_I(k) = \tilde{v}_E(k) + Y^{r\perp}\tilde{\mu}(k) = -\tilde{F}x(k) + \tilde{D}_e e(k) + \tilde{D}_w\tilde{w}(k)$$

and define the signals

$$\begin{aligned}
\tilde{\psi}_I(k) &= \tilde{\psi}_0(k) + \tilde{D}_{N_s,43}\tilde{v}_I(k) \\
&= (\tilde{C}_{N_s,4} - \tilde{D}_{N_s,43}\tilde{F})x(k) + (\tilde{D}_{N_s,41} + \tilde{D}_{N_s,43}\tilde{D}_e)e(k) + (\tilde{D}_{N_s,42} + \tilde{D}_{N_s,43}\tilde{D}_w)\tilde{w}(k) && (5.133) \\
\hat{x}_{T,I}(k+N_s|k) &= \hat{x}_{T,0}(k+N_s|k) + \tilde{B}_{N_s,3}\tilde{v}_I(k) && (5.134) \\
&= (\tilde{A}_{N_s} - \tilde{B}_{N_s,3}\tilde{F})x(k) + (\tilde{B}_{N_s,1} + \tilde{B}_{N_s,3}\tilde{D}_e)e(k) + (\tilde{B}_{N_s,2} + \tilde{B}_{N_s,3}\tilde{D}_w)\tilde{w}(k) && (5.135)
\end{aligned}$$

then

$$\begin{aligned}
\tilde{\psi}(k) &= \tilde{\psi}_I(k) + \tilde{D}_{N_s,43}Y^{r\perp}\tilde{\mu}_I(k) \\
\hat{x}_T(k+N_s|k) &= \hat{x}_{T,I}(k+N_s|k) + \tilde{B}_{N_s,3}Y^{r\perp}\tilde{\mu}_I(k)
\end{aligned}$$

and we can rewrite the constraint as:

$$\begin{bmatrix} \tilde{\psi}(k) - \tilde{\Psi}(k) \\ W_{n_\Psi}\,\hat{x}_T(k+N_s|k) - \underline{1} \end{bmatrix} = A_\psi\tilde{\mu}_I(k) + b_\psi(k) \le 0 \tag{5.136}$$

where

$$
\begin{aligned}
b_\psi(k) &= \begin{bmatrix} \tilde{\psi}_I(k) - \tilde{\Psi}(k) \\ W_{n_\Psi}\,\hat{x}_{T,I}(k+N_s|k) - \underline{1} \end{bmatrix} && (5.137)\\
A_\psi &= \begin{bmatrix} \tilde{D}_{N_s,43} \\ W_{n_\psi}\tilde{B}_{N_s,3} \end{bmatrix} Y^{r\perp} && (5.138)
\end{aligned}
$$

Substitution of (5.133) and (5.135) in (5.137) gives

$$
\begin{aligned}
b_\psi(k) &= \begin{bmatrix} \tilde{C}_{N_s,4} - \tilde{D}_{N_s,43}\tilde{F} \\ W_{n_\Psi}(\tilde{A}_{N_s} - \tilde{B}_{N_s,3}\tilde{F}) \end{bmatrix} x(k) + \begin{bmatrix} \tilde{D}_{N_s,41} + \tilde{D}_{N_s,43}\tilde{D}_e \\ W_{n_\Psi}(\tilde{B}_{N_s,1} + \tilde{B}_{N_s,3}\tilde{D}_e) \end{bmatrix} e(k) + \\
&= \begin{bmatrix} \tilde{D}_{N_s,42} + \tilde{D}_{N_s,43}\tilde{D}_w \\ W_{n_\Psi}(\tilde{B}_{N_s,2} + \tilde{B}_{N_s,3}\tilde{D}_w) \end{bmatrix} \tilde{w}(k) - \begin{bmatrix} \tilde{\Psi}(k) \\ \underline{1} \end{bmatrix}
\end{aligned}
$$

The constrained infinite horizon MPC problem is now equivalent to a Quadratic Programming problem of minimizing (5.132) subject to (5.136). Note that the term $c'(k)$ in (5.132) does not play any role in the optimization and can therefore be skipped.

□ End Proof

5.4 Implementation

5.4.1 Implementation and computation in LTI case

In this chapter the predictive control problem was solved for various settings (unconstrained, equality/inequality constrained, finite/infinite horizon). In the absence of inequality constraints, the resulting controller is given by the control law:

$$
v(k) = -F\,x(k) + D_e\,e(k) + D_w\,\tilde{w}(k) \qquad (5.139)
$$

with E_v such that $v(k) = E_v\tilde{v}(k)$.
Unfortunately, the state x of the plant and the true value of $e(k)$ are unknown. We therefore introduce a controller state $x_c(k)$ and an estimate e_c. By substitution of x_c and e_c in system equations (5.1)-(5.2) and control law (5.139) we obtain:

$$
\begin{aligned}
x_c(k+1) &= A\,x_c(k) + B_1\,e_c(k) + B_2\,w(k) + B_3\,v(k) && (5.140)\\
e_c(k) &= D_{11}^{-1}\big(y(k) - C_1\,x_c(k) - D_{12}\,w(k)\big) && (5.141)\\
v(k) &= -F\,x_c(k) + D_e\,e_c(k) + D_w\,\tilde{w}(k) && (5.142)
\end{aligned}
$$

Elimination of $e_c(k)$ and $v(k)$ from the equations results in the closed loop form for the controller:

$$
\begin{aligned}
x_c(k+1) &= (A - B_3F - B_1D_{11}^{-1}C_1 - B_3D_eD_{11}^{-1}C_1)\,x_c(k) \\
&\quad +(B_1D_{11}^{-1} + B_3D_eD_{11}^{-1})\,y(k) \\
&\quad +(B_3D_w - B_3D_e\,D_{11}^{-1}D_{12}E_w + B_2E_w - B_1D_{11}^{-1}D_{12}E_w)\,\tilde{w}(k) && (5.143)\\
v(k) &= (-F - D_e\,D_{11}^{-1}\,C_1)\,x_c(k) + D_e\,D_{11}^{-1}y(k) + \\
&\quad +(D_w - D_e\,D_{11}^{-1}D_{12}E_w)\,\tilde{w}(k) && (5.144)
\end{aligned}
$$

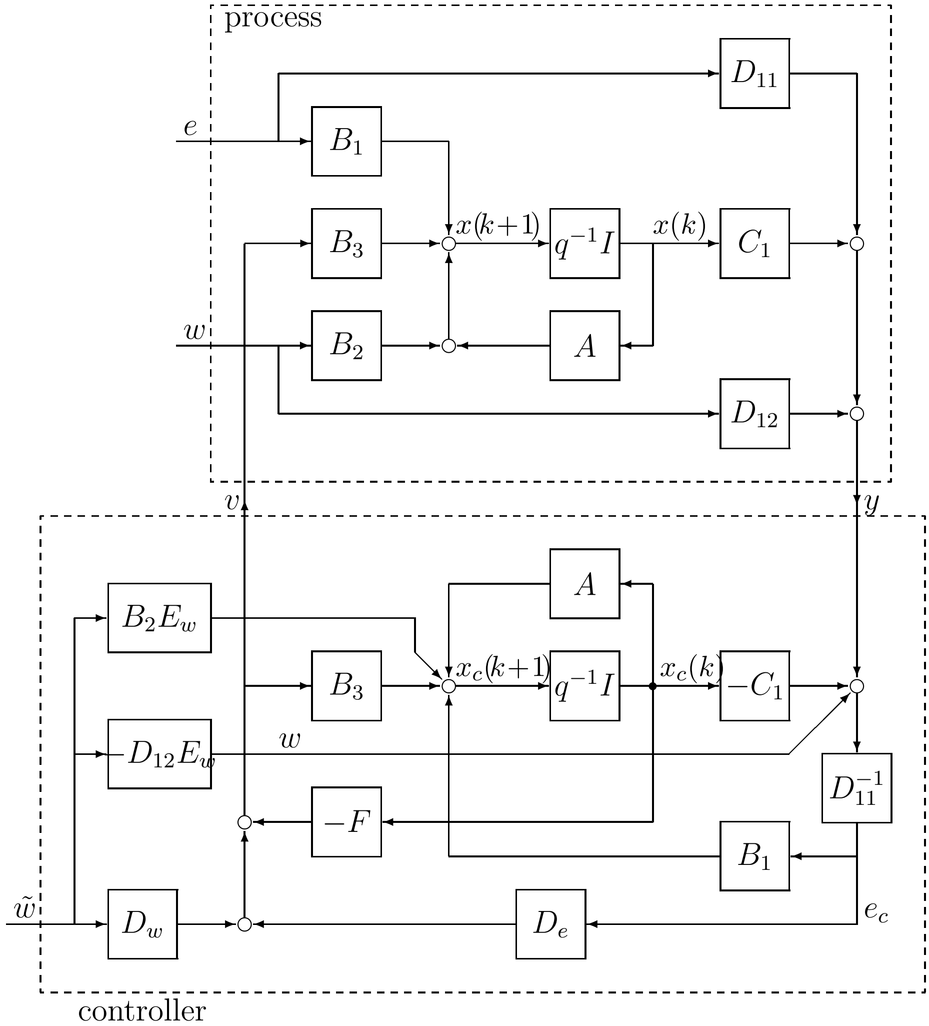

Figure 5.1: Realization of the LTI SPCP controller

with $E_w = \begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix}$ is such that $w(k) = E_w\tilde{w}(k)$.

Note that this is an linear time-invariant (LTI) controller.

Figure 5.1 visualizes the controller, where the upper block is the true process, satisfying equations (5.1) and (5.2), and the lower block is the controller, satisfying equations (5.140), (5.141) and (5.142). Note that the states of the process and controller are denoted by x and x_c, respectively. When the controller is stabilizing and there is no model error, the state x_c and the noise estimate e_c will converge to the true state x and true noise signal e respectively. After a transient the states of plant and controller will become the same. Stability of the closed loop is discussed in chapter 6.

Computational aspects:

Consider the expression (5.26)-(5.28). The matrices F, D_e and D_w can also be found by solving

$$(\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23})\begin{bmatrix} \tilde{F} & \tilde{D}_e & \tilde{D}_w \end{bmatrix} = \tilde{D}_{23}^T\bar{\Gamma}\begin{bmatrix} \tilde{C}_2 & \tilde{D}_{21} & \tilde{D}_{22} \end{bmatrix} \tag{5.145}$$

where we define $F = E_v\tilde{F}$, $D_e = E_v\tilde{D}_e$ and $D_w = E_v\tilde{D}_w$. Inversion of $(\tilde{D}_{23}^T\bar{\Gamma}\tilde{D}_{23})$ may be badly conditioned and should be avoided. As defined in chapter 4, $\Gamma(j)$ is a diagonal selection matrix with ones and zeros on the diagonal. This means that $\bar{\Gamma} = \bar{\Gamma}^2$. Define $M = \bar{\Gamma}\tilde{D}_{23}$, then we can do a singular value decomposition of M:

$$M = \begin{bmatrix} U_1 & U_2 \end{bmatrix}\begin{bmatrix} \Sigma \\ 0 \end{bmatrix}V^T$$

and the equation 5.145 becomes:

$$(V\Sigma^2V^T)\begin{bmatrix} \tilde{F} & \tilde{D}_e & \tilde{D}_w \end{bmatrix} = V\Sigma U^T\begin{bmatrix} \tilde{C}_2 & \tilde{D}_{21} & \tilde{D}_{22} \end{bmatrix} \tag{5.146}$$

and so the solution becomes:

$$\begin{aligned}
\begin{bmatrix} \tilde{F} & \tilde{D}_e & \tilde{D}_w \end{bmatrix} &= (V\Sigma^2V^T)^{-1}V\Sigma U_1^T\begin{bmatrix} \tilde{C}_2 & \tilde{D}_{21} & \tilde{D}_{22} \end{bmatrix} = \\
&= V\Sigma^{-1}U_1^T\begin{bmatrix} \tilde{C}_2 & \tilde{D}_{21} & \tilde{D}_{22} \end{bmatrix}
\end{aligned}$$

and so

$$\begin{bmatrix} F & D_e & D_w \end{bmatrix} = E_v V\Sigma^{-1}U_1^T\begin{bmatrix} \tilde{C}_2 & \tilde{D}_{21} & \tilde{D}_{22} \end{bmatrix}$$

For singular value decomposition robust and reliable algorithms are available. Note that the inversion of Σ is simple, because it is a diagonal matrix and all elements are positive (we assumes $\tilde{D}_{23}\bar{\Gamma}\tilde{D}_{23}^T$ to have full rank).

5.4.2 Implementation and computation in full SPCP case

In this section we discuss the implementation of the full SPCP controller. For both the finite and infinite horizon the full SPCP can be solved using an optimal control law

$$v(k) = v_0(k) + M_2\tilde{\mu}(k) \tag{5.147}$$

where $\tilde{\mu}$ is the solution for the quadratic programming problem

$$\min_{\tilde{\mu}} \frac{1}{2}\tilde{\mu}^T H \tilde{\mu} \tag{5.148}$$

subject to:

$$A_\psi \tilde{\mu} + b_\psi(k) \leq 0 \tag{5.149}$$

This optimization problem (5.147)-(5.149) can be seen as a nonlinear mapping with input variables $x(k)$, $e(k)$ and $\tilde{w}(k)$ and output $v(k)$. As in the LTI-case (section 5.4.1), estimations $x_c(k)$ and $e_c(k)$ from the state $x(k)$ and noise signal $e(k)$, can be obtained using an observer.
Figure 5.2 visualizes the nonlinear controller, where the upper block is the true process, satisfying equations (5.1) and (5.2), and the lower block is the controller, where the nonlinear mapping has inputs $x_c(k)$, $e_c(k)$, $\tilde{w}(k)$ and $\tilde{\Psi}(k)$ and output $v(k)$. Again, when the controller is stabilizing and there is no model error, the state x_c and the noise estimate e_c will converge to the true state x and true noise signal e respectively. After a transient the states of plant and controller will become the same.
An advantage of predictive control is that, by writing the control law as the result of a constrained optimization problem (5.148)-(5.149), it can effectively deal with constraints. An important disadvantage is that every time step a computationally expensive optimization problem has to be solved [138]. The time required for the optimization makes model predictive control not suitable for fast systems and/or complex problems.
In this section we discuss the work of Bemporad *et al.* [11], who formulate the linear model predictive control problem as multi-parametric quadratic programs. The control variables are treated as optimization variables and the state variables as parameters. The optimal control action is a continuous and piecewise affine function of the state under the assumption that the active constraints are linearly independent. The key advantage of this approach is that the control actions are computed off-line: the on-line computation simply reduces to a function evaluation problem.
The Kuhn-Tucker conditions for the quadratic programming problem (5.148)-(5.149) are given by:

$$\begin{aligned} H\,\tilde{\mu} + A_\psi^T \lambda &= 0 & (5.150)\\ \lambda^T\Big(A_\psi \tilde{\mu} + b_\psi(k)\Big) &= 0 & (5.151)\\ \lambda &\geq 0 & (5.152)\\ A_\psi \tilde{\mu} + b_\psi(k) &\leq 0 & (5.153) \end{aligned}$$

Equation (5.150) leads to optimal value for $\tilde{\mu}$:

$$\tilde{\mu} = -H^{-1} A_\psi^T \lambda \tag{5.154}$$

substitution into equation (5.151) gives condition

$$\lambda^T\Big(- A_\psi H^{-1} A_\psi^T \lambda + b_\psi(k)\Big) = 0 \tag{5.155}$$

Let λ_a and λ_i denote the Lagrangian multipliers corresponding the active and inactive constraints respectively. For inactive constraint there holds $\lambda_i = 0$, for active constraint there holds $\lambda_a > 0$. Let S_i and S_a be selection matrices such that

$$\begin{bmatrix} \lambda_a \\ \lambda_i \end{bmatrix} = \begin{bmatrix} S_a \\ S_i \end{bmatrix} \lambda$$

then

$$\lambda = S_a^T \lambda_a$$

Substitution in (5.154) and (5.155) gives:

$$\tilde{\mu} = -H^{-1} A_\psi^T S_a^T \lambda_a \tag{5.156}$$

$$\lambda_a^T S_a \Big(- A_\psi H^{-1} A_\psi^T S_a \lambda_a + b_\psi(k) \Big) = 0 \tag{5.157}$$

(5.157) leads to the condition:

$$-S_a A_\psi H^{-1} A_\psi^T S_a \lambda_a + S_a b_\psi(k) = 0$$

so

$$\lambda_a = \Big(S_a A_\psi H^{-1} A_\psi^T S_a \Big)^{-1} S_a b_\psi(k) \tag{5.158}$$

Substitution in (5.156) gives

$$\tilde{\mu} = -H^{-1} A_\psi^T S_a^T \Big(S_a A_\psi H^{-1} A_\psi^T S_a \Big)^{-1} S_a b_\psi(k) \tag{5.159}$$

inequality (5.153) now becomes

$$\begin{aligned} & A_\psi \tilde{\mu} + b_\psi(k) = \\ &= \; b_\psi(k) - A_\psi H^{-1} A_\psi^T S_a^T \Big(S_a A_\psi H^{-1} A_\psi^T S_a \Big)^{-1} S_a b_\psi(k) \\ &= \; \Big[\; -A_\psi H^{-1} A_\psi^T S_a^T \Big(S_a A_\psi H^{-1} A_\psi^T S_a \Big)^{-1} S_a \; \Big] b_\psi(k) \\ &\leq \; 0 \end{aligned} \tag{5.160}$$

The solution $\tilde{\mu}$ of (5.159) for a given choice of S_a is the optimal solution of the quadratic programming problem as long as (5.160) holds and $\lambda_a \geq 0$ Combining (5.158) and (5.160) gives:

$$\begin{bmatrix} M_1(S_a) \\ M_2(S_a) \end{bmatrix} b_\psi(k) \leq 0 \tag{5.161}$$

where

$$\begin{aligned} M_1(S_a) &= -A_\psi H^{-1} A_\psi^T S_a^T \Big(S_a A_\psi H^{-1} A_\psi^T S_a \Big)^{-1} S_a \\ M_2(S_a) &= -\Big(S_a A_\psi H^{-1} A_\psi^T S_a \Big)^{-1} S_a \end{aligned}$$

Let

$$b_\psi(k) = N \begin{bmatrix} x_c(k) \\ e_c(k) \\ \tilde{w}(k) \\ \tilde{\Psi}(k) \end{bmatrix}$$

Consider the singular value decomposition

$$N = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

where Σ is square and has full-rank n_N, and define

$$\chi(k) = \Sigma\, V_1^T \begin{bmatrix} x_c(k) \\ e_c(k) \\ \tilde{w}(k) \\ \tilde{\Psi}(k) \end{bmatrix} \tag{5.162}$$

Condition (5.161) now becomes

$$\begin{bmatrix} M_1(S_a) \\ M_2(S_a) \end{bmatrix} U_1^T \chi(k) \leq 0 \tag{5.163}$$

Inequality (5.163) describes a polyhedra $\mathcal{P}(S_a)$ in the $\mathbb{R}^{n_N}$, and represent a set of all χ, that leads to the set of active constraints described by matrix S_a, and thus corresponding to the solution $\tilde{\mu}$, described by (5.159).
For all combinations of active constraints (for all S_a) one can compute the set $\mathcal{P}(S_a)$ and in this way a part of the $\mathbb{R}^{n_N}$ will be covered. For some values of χ (or in other words, for some $\big(x_c(k), e_c(k), \tilde{w}(k), \tilde{\Psi}(k)\big)$), there is no possible combination of active constraints, and so no feasible solution to the QP-problem is found. In this case we have a feasibility problem, for these values of χ we have to relax the predictive control problem in some way. (this will be discussed in the next section).
All sets $\mathcal{P}(S_a)$ can be computed off-line. The on-line optimization has now become a search-problem. At time k we can compute $\chi(k)$ following equation (5.162), and determine to which set $\mathcal{P}(S_a)$ the vector $\chi(k)$ belongs. $\tilde{\mu}$ can be computed using the corresponding S_a in equation (5.159).
Resuming, the method of Bemporad *et al.* [11], proposes an algorithm that partitions the state space into polyhedral sets and computes the coefficients of the affine function for every set. The result is a search tree that determines the set to which a state belongs. The tree is computed off-line, and the classification and computation of the control signal from the state and the coefficients is done on-line. The method works very well if the number of inequality constraints is small. However, the number of polyhedral sets grows dramatically with the number of inequality constraints.
Another way to avoid the cumbersome optimization, is described by Hoekstra *et al.* [59], where a feed-forward neural network is used to approximate the nonlinear mapping $v(k) =$

$h(x_c(k), e_c(k), \tilde{w}(k), \tilde{\Psi}(k))$, which maps the state, the reference signal, the noise and the disturbances to the control signal. The existence of a function is proved, and it is shown that this function is continuous. Contrary to [11], linear independency of the active constraints is not required, and the approach is applicable to control problems with many constraints.

5.5 Feasibility

At the end of this chapter, a remark has to be made concerning feasibility and stability. A drawback of using hard constraints is that it may lead to infeasibility: There is no possible control action without violation of the constraints. This can happen when the prediction and control horizon are chosen too small, around setpoint changes or when noise and disturbance levels are high. When there is no feasibility guarantee, stability can not be proven.

Two algorithms to handle infeasibility are discussed in this section:

- soft-constraint approach
- minimal time approach

Soft-constraint algorithm

In contrast to hard constraints one can define soft constraints [100],[140], given by an additional term to the performance index and only penalizing constraint violations. For example, consider the problem with the hard constraint

$$\min_{v} J \qquad \text{subject to} \qquad \tilde{\psi} \leq \tilde{\Psi} \tag{5.164}$$

This output constraint can be softened by considering the following problem:

$$\min_{v\,,\,\alpha} J + c\,\alpha^2 \qquad \text{subject to} \qquad \tilde{\psi} \leq \tilde{\Psi} + \alpha\ , \quad \alpha \geq 0\ , \quad c \gg 1 \tag{5.165}$$

In this way, violation of the constraints is allowed, but on the cost of a higher performance index.

minimal time algorithm

In the minimal time approach [96] consider the inequality constraints $\phi(k+j) \leq \Phi(k+j)$ for $j = 1, \ldots, N$. (where $\tilde{\phi}$ is the stacked vector with all $\phi(k+j)$, see section 4.2.2). Now we disregard the constraints at the start of the prediction interval up to some sample number j_{mt} and determine the minimal j_{mt} for which the problem becomes feasible:

$$\min_{v, j_{mt}} j_{mt} \qquad \text{subject to} \qquad \psi(k+j) \leq \Psi(k+j) \ \text{ for } \ j = j_{mt}+1, \ldots, N \tag{5.166}$$

Now we can calculate the optimal input sequence $\tilde{v}(k)$ by optimizing the following problem:

$$\min_{v} J \qquad \text{subject to} \qquad \psi(k+j) \leq \Psi(k+j) \ \text{ for } \ j = j_{mt}+1, \ldots, N \tag{5.167}$$

where j_{mt} is the minimum value of problem (5.166).

5.6 Examples

Example 22 : computation of controller matrices (unconstrained case)

Consider the system of example 10 on page 82. When we compute the optimal predictive control for the unconstrained case, with $N = N = N_c = 5$, we obtain the controller matrices:

$$\begin{aligned} F &= \begin{bmatrix} 0 & 1 & -0.4 & 0.3 & 1 \end{bmatrix} \\ D_e &= \begin{bmatrix} -0.6 \end{bmatrix} \\ D_w &= \begin{bmatrix} -0.1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

Example 23 : computation of controller matrices (equality constrained case)

We add a control horizon for $N_c = 1$ and use the results for the equality constrained case to obtain the controller matrices:

$$\begin{aligned} F &= \begin{bmatrix} 0 & 0.8624 & -0.2985 & 0.3677 & 1 \end{bmatrix} \\ D_e &= \begin{bmatrix} -0.6339 \end{bmatrix} \\ D_w &= \begin{bmatrix} -0.0831 & 0.8624 & -0.0232 & 0.2587 & -0.0132 & 0.1639 & -0.0158 & 0.1578 \end{bmatrix} \end{aligned}$$

Example 24 : MPC of an elevator

A predictive controller design problem with constraints will be illustrated using a triple integrator chain

$$\dddot{y}\,(t) = u(t) + \epsilon(t)$$

with constraints on y, $\dot{y}$, $\ddot{y}$ and $\dddot{y}$. This is a model of a rigid body approximation of an elevator for position control, subject to constraints on speed, acceleration and jerk (time derivative of accelaration).
Define the state

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} \ddot{y}(t) \\ \dot{y}(t) \\ y(t) \end{bmatrix}$$

The noise model is given by:

$$\epsilon(t) = \dddot{e}\,(t) + 0.5\ddot{e}(t) + \dot{e}(t) + 0.1e(t)$$

where e is given as zero-mean white noise. Then the continuous-time state space description of this system can be given

$$\begin{aligned} \dot{x}_1(t) &= u(t) + 0.1e(t) \\ \dot{x}_2(t) &= x_1(t) + e(t) \\ \dot{x}_3(t) &= x_2(t) + 0.5 * e(t) \\ y(t) &= x_3(t) + e(t) \end{aligned}$$

where the noise is now acting on the states $\dddot{y}\,(t)$ *and output* $y(t)$*. In matrix form we get:*

$$\begin{aligned} \dot{x}(t) &= A_c\, x(t) + K_c\, e(t) + B_c\, u(t) \\ y(t) &= C_c\, x(t) + e(t) \end{aligned}$$

where

$$A_c = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad B_c = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad K_c = \begin{bmatrix} 0.1 \\ 1 \\ 0.5 \end{bmatrix} \quad C_c = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

A zero-order hold transformation with sampling-time T *gives a discrete-time IO model:*

$$\begin{aligned} x_o(k+1) &= A_o\, x_o(k) + K_o\, e(k) + B_o\, u(k) \\ y(k) &= C_o\, x_o(k) \end{aligned}$$

where

$$A_o = \begin{bmatrix} 1 & 0 & 0 \\ T & 1 & 0 \\ T^2/2 & T & 1 \end{bmatrix} \quad B_o = \begin{bmatrix} T \\ T^2/2 \\ T^3/6 \end{bmatrix} \quad K_o = \begin{bmatrix} 0.1\, T \\ T + T^2/20 \\ T/2 + T^2/6 + T^3/60 \end{bmatrix}$$

$$C_o = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

The prediction model is built up as given in section 3. Note that we use $u(k)$ *rather than* $\Delta u(k)$*. This because the system already consists of a triple integrator, and another integrator in the controller is not desired. The aim of this example is to direct the elevator from position* $y = 0$ *at time* $t = 0$ *to position* $y = 1$ *as fast as possible. For a comfortable and safe operation, control is done subject to constraints on the jerk, acceleration, speed and overshoot, given by*

$$\begin{aligned} |\dddot{y}\,(t)| &\le 0.4 && \textit{(jerk)} \\ |\ddot{y}(t)| &\le 0.4 && \textit{(acceleration)} \\ |\dot{y}(t)| &\le 0.4 && \textit{(speed)} \\ y(t) &\le 1.01 && \textit{(overshoot)} \end{aligned}$$

These constraints can be translated into linear constraints on the control signal $u(k)$ *and prediction of the state* $\hat{x}(k)$*:*

$$\begin{aligned} u(k+j-1) &\le 0.4 && \textit{(positive jerk)} \\ -u(k+j-1) &\le 0.4 && \textit{(negative jerk)} \\ \hat{x}_1(k+j) &\le 0.4 && \textit{(positive acceleration)} \\ -\hat{x}_1(k+j) &\le 0.4 && \textit{(negative acceleration)} \\ \hat{x}_2(k+j) &\le 0.4 && \textit{(positive speed)} \\ -\hat{x}_2(k+j) &\le 0.4 && \textit{(negative speed)} \\ \hat{x}_3(k+j) &\le 1.01 && \textit{(overshoot)} \end{aligned}$$

for all $j = 1, \ldots, N$.
We choose sampling-time $T = 0.1$, *and prediction and control horizon* $N = N_c = 30$, *and minimum cost-horizon* $N_m = 1$. *Further we consider a constant reference signal* $r(k) = 1$ *for all* k, *and the weightings parameters* $P(q) = 1$, $\lambda = 0.1$. *The predictive control problem is solved by minimizing the GPC performance index (for an IO model)*

$$J(u,k) = \sum_{j=N_m}^{N} \Big(\hat{\phi}(k+j|k) - r(k+j|k)\Big)^T \Big(\hat{\phi}(k+j|k) - r(k+j|k)\Big) +$$

$$+\lambda^2 \sum_{j=1}^{N_c} u^T(k+j-1)u(k+j-1)$$

subject to the above linear constraints.
The optimal control sequence is computed and implemented. In figure 5.3 the results are given. It can be seen that the jerk (solid line), acceleration (dotted line), speed (dashed line) and the position (dashed-dotted line) are all bounded by the constraints.

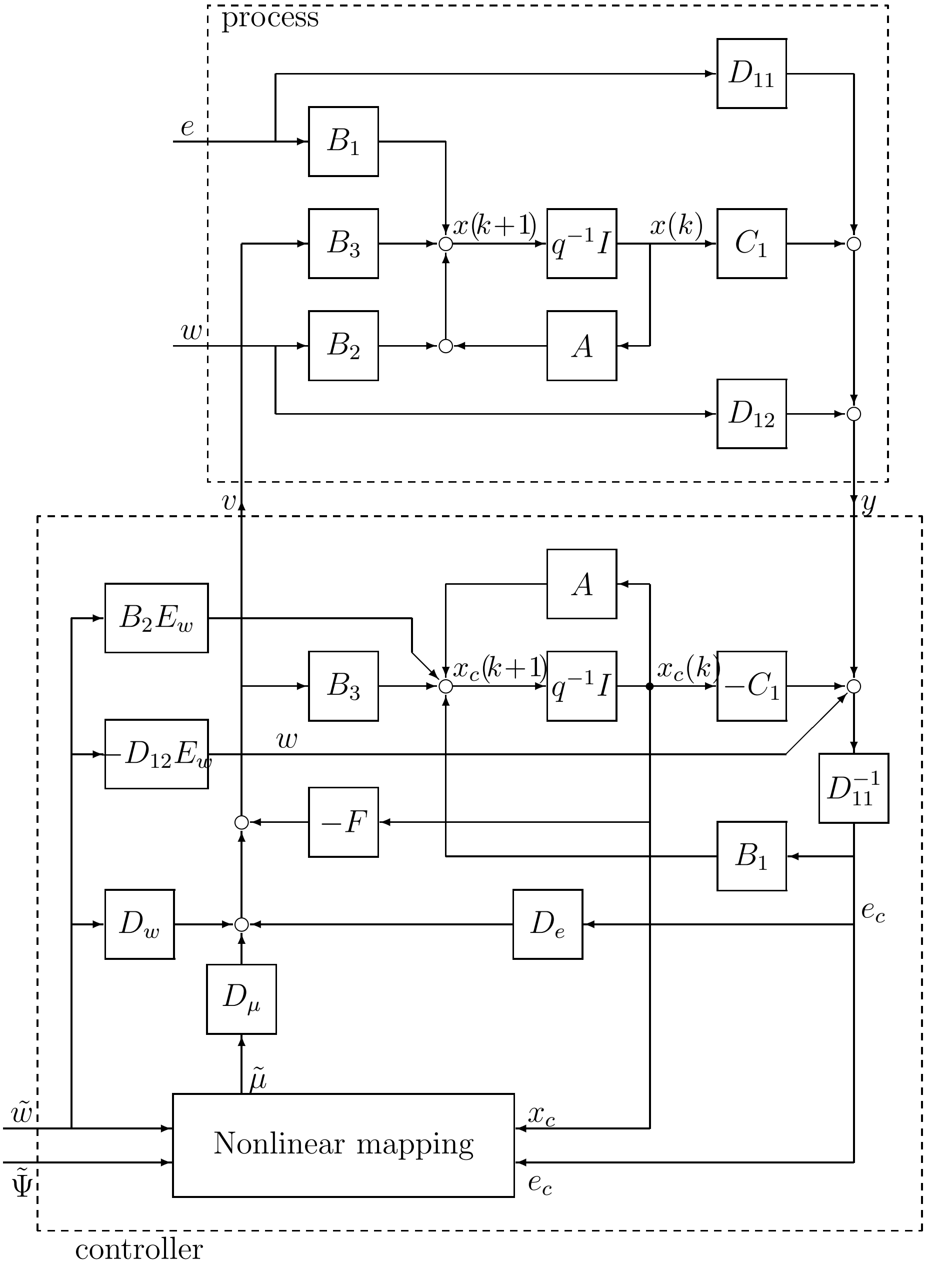

Figure 5.2: Realization of the FULL SPCP controller

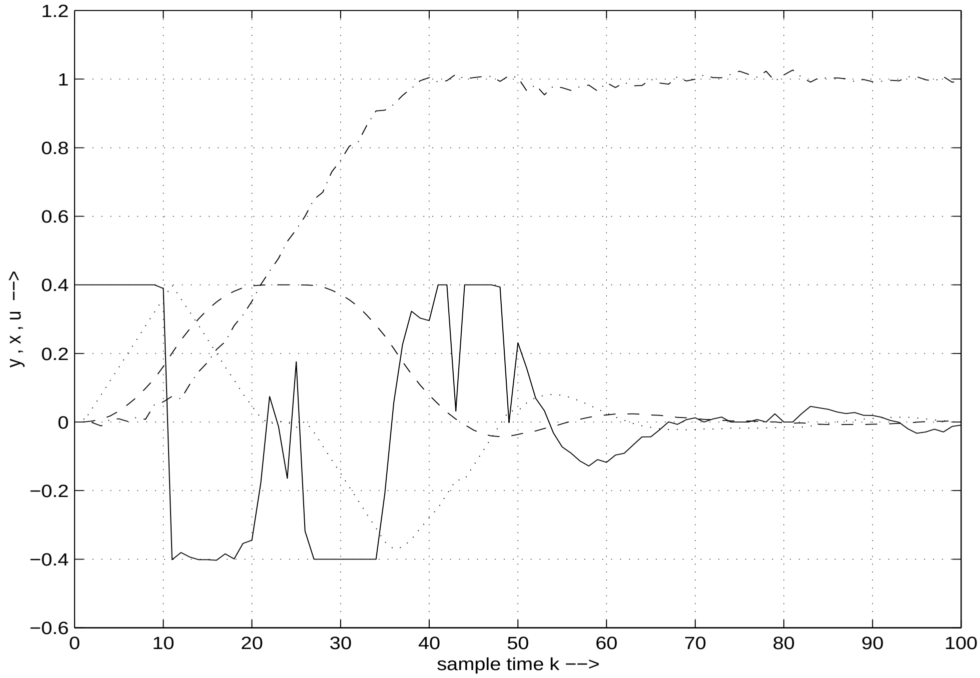

Figure 5.3: Predictive control of an elevator

Chapter 6

Stability

Predictive control design does not give an a priori guaranteed stabilizing controller. Methods like LQ, H_∞, ℓ_1-control have built-in stability properties. In this chapter we analyse the stability of a predictive controller. We distinguish two different cases, namely the unconstrained and constrained case. It is very important to notice that a predictive controller that is stable for the unconstrained case is not automatically stable for the constrained case.

6.1 Stability for the LTI case

In the case where inequality constraints are absent, the problem of stability is not too difficult, because the controller is linear and time-invariant (LTI).

6.1.1 Analysis

Let the process be given by:

$$\begin{aligned} x(k+1) &= A\,x(k) + B_1\,e(k) + B_2\,w(k) + B_3\,v(k) \\ y(k) &= C_1\,x(k) + D_{11}\,e(k) + D_{12}\,w(k) \end{aligned} \tag{6.1}$$

and the controller by (5.143) and (5.144):

$$\begin{aligned} x_c(k+1) &= (A - B_3F - B_1D_{11}^{-1}C_1 - B_3D_eD_{11}^{-1}C_1)\,x_c(k) \\ &\quad +(B_1D_{11}^{-1} + B_3D_eD_{11}^{-1})\,y(k) + \\ &\quad +(B_3D_w - B_3D_eD_{11}^{-1}D_{12}E_w + B_2E_w - B_1D_{11}^{-1}D_{12}E_w)\,\tilde{w}(k) \\ v(k) &= (-F - D_eD_{11}^{-1}C_1)\,x_c(k) + D_eD_{11}^{-1}\,y(k) + (D_w - D_eD_{11}^{-1}D_{12}E_w)\,\tilde{w}(k) \end{aligned}$$

with $E_w = \begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix}$ is such that $w(k) = E_w\tilde{w}(k)$. We make one closed loop state space representation by substitution:

$$v(k) = -(F + D_eD_{11}^{-1}C_1)\,x_c(k) + D_eD_{11}^{-1}\,y(k) + (D_w - D_eD_{11}^{-1}D_{12}E_w)\,\tilde{w}(k)$$

$$\begin{aligned}
&= -(F + D_e D_{11}^{-1} C_1)\, x_c(k) + D_e D_{11}^{-1} C_1\, x(k) + D_e e(k) + D_w\, \tilde{w}(k)\\
x(k+1) &= A\, x(k) + B_1\, e(k) + B_2 E_w\, \tilde{w}(k) + B_3\, v(k)\\
&= A\, x(k) + B_1\, e(k) + B_2 E_w \tilde{w}(k) - (B_3 F + B_3 D_e D_{11}^{-1} C_1)\, x_c(k)\\
&\quad + B_3 D_e D_{11}^{-1}\, C_1 x(k) + B_3 D_e e(k) + B_3 D_w\, \tilde{w}(k)\\
&= (A + B_3 D_e D_{11}^{-1} C_1)\, x(k) - (B_3 F + B_3 D_e D_{11}^{-1} C_1)\, x_c(k)\\
&\quad + (B_1 + B_3 D_e)\, e(k) + (B_3 D_w + B_2 E_w)\, \tilde{w}(k)\\
x_c(k+1) &= (A - B_3 F - B_1 D_{11}^{-1} C_1 - B_3 D_e D_{11}^{-1} C_1)\, x_c(k)\\
&\quad + (B_1 D_{11}^{-1} + B_3 D_e D_{11}^{-1})\, y(k)\\
&\quad + (B_3 D_w - B_3 D_e D_{11}^{-1} D_{12} E_w + B_2 E_w - B_1 D_{11}^{-1} D_{12} E_w) \tilde{w}(k)\\
&= (A - B_3 F - B_1 D_{11}^{-1} C_1 - B_3 D_e D_{11}^{-1} C_1)\, x_c(k)\\
&\quad + (B_1 D_{11}^{-1} C_1 + B_3 D_e D_{11}^{-1} C_1)\, x(k) + (B_1 + B_3 D_e)\, e(k)\\
&\quad + (B_3 D_w + B_2 E_w)\, \tilde{w}(k)
\end{aligned}$$

or in matrix form:

$$\begin{aligned}
\begin{bmatrix} x(k+1) \\ x_c(k+1) \end{bmatrix} &= \begin{bmatrix} A + B_3 D_e D_{11}^{-1} C_1 & -B_3 F - B_3 D_e D_{11}^{-1} C_1 \\ B_1 D_{11}^{-1} C_1 + B_3 D_e D_{11}^{-1} C_1 & A - B_3 F - B_1 D_{11}^{-1} C_1 - B_3 D_e D_{11}^{-1} C_1 \end{bmatrix} \begin{bmatrix} x(k) \\ x_c(k) \end{bmatrix}\\
&\quad + \begin{bmatrix} B_1 + B_3 D_e \\ B_1 + B_3 D_e \end{bmatrix} e(k) + \begin{bmatrix} B_3 D_w + B_2 E_w \\ B_3 D_w + B_2 E_w \end{bmatrix} \tilde{w}(k)\\
&= A_{cl} \begin{bmatrix} x(k) \\ x_c(k) \end{bmatrix} + B_{1,cl}\, e(k) + B_{2,cl}\, \tilde{w}(k)\\
\begin{bmatrix} v(k) \\ y(k) \end{bmatrix} &= \begin{bmatrix} D_e D_{11}^{-1} C_1 & -F - D_e D_{11}^{-1} C_1 \\ C_1 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x_c(k) \end{bmatrix} + \begin{bmatrix} D_e \\ D_{11} \end{bmatrix} e(k) + \begin{bmatrix} D_w \\ D_{12} \end{bmatrix} \tilde{w}(k)\\
&= C_{cl} \begin{bmatrix} x(k) \\ x_c(k) \end{bmatrix} + D_{1,cl}\, e(k) + D_{2,cl}\, \tilde{w}(k)
\end{aligned}$$

Next we choose a new state:

$$\begin{bmatrix} x(k) \\ x_c(k) - x(k) \end{bmatrix} = \begin{bmatrix} I & 0 \\ -I & I \end{bmatrix} \begin{bmatrix} x(k) \\ x_c(k) \end{bmatrix} = T \begin{bmatrix} x(k) \\ x_c(k) \end{bmatrix}$$

This state transformation with

$$T = \begin{bmatrix} I & 0 \\ -I & I \end{bmatrix} \qquad \text{and} \qquad T^{-1} = \begin{bmatrix} I & 0 \\ I & I \end{bmatrix}$$

leads to a new realization

$$\begin{aligned}
\begin{bmatrix} x(k) \\ x_c(k) - x(k) \end{bmatrix} &= \bar{A}_{cl} \begin{bmatrix} x(k) \\ x_c(k) - x(k) \end{bmatrix} + \bar{B}_{1,cl}\, e(k) + B_{2,cl}\, \tilde{w}(k)\\
\begin{bmatrix} v(k) \\ y(k) \end{bmatrix} &= \bar{C}_{cl} \begin{bmatrix} x(k) \\ x_c(k) \end{bmatrix} + \bar{D}_{1,cl}\, e(k) + \bar{D}_{2,cl}\, \tilde{w}(k)
\end{aligned}$$

with system matrices (see appendix B):

$$\begin{aligned}
\bar{A}_{cl} &= T\,A_{cl}\,T^{-1} \\
&= \begin{bmatrix} I & 0 \\ -I & I \end{bmatrix} \begin{bmatrix} A+B_3D_eD_{11}^{-1}C_1 & -B_3F-B_3D_eD_{11}^{-1}C_1 \\ B_1D_{11}^{-1}C_1+B_3D_eD_{11}^{-1}C_1 & A-B_3F-B_1D_{11}^{-1}C_1-B_3D_eD_{11}^{-1}C_1 \end{bmatrix} \begin{bmatrix} I & 0 \\ I & I \end{bmatrix} \\
&= \begin{bmatrix} A-B_3F & -B_3F-B_3D_eD_{11}^{-1}C_1 \\ 0 & A-B_1D_{11}^{-1}C_1 \end{bmatrix} \\
\bar{B}_{1,cl} &= T\,B_{1,cl} \\
&= \begin{bmatrix} I & 0 \\ -I & I \end{bmatrix} \begin{bmatrix} B_1+B_3D_e \\ B_1+B_3D_e \end{bmatrix} \\
&= \begin{bmatrix} B_1+B_3D_e \\ 0 \end{bmatrix} \\
\bar{B}_{2,cl} &= T\,B_{2,cl} \\
&= \begin{bmatrix} I & 0 \\ -I & I \end{bmatrix} \begin{bmatrix} B_3D_w+B_2E_w \\ B_3D_w+B_2E_w \end{bmatrix} \\
&= \begin{bmatrix} B_3D_w+B_2E_w \\ 0 \end{bmatrix} \\
\bar{C}_{cl} &= C_{cl}T^{-1} \\
&= \begin{bmatrix} D_eD_{11}^{-1}C_1 & -F-D_eD_{11}^{-1}C_1 \\ C_1 & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ I & I \end{bmatrix} \\
&= \begin{bmatrix} -F & -F-D_eD_{11}^{-1}C_1 \\ C_1 & 0 \end{bmatrix} \\
\bar{D}_{1,cl} &= D_{1,cl} \\
\bar{D}_{2,cl} &= D_{2,cl}
\end{aligned}$$

The eigenvalues of the matrix $\bar{A}_{cl}$ are equal to eigenvalues of $(A-B_3F)$ and the eigenvalues of $(A-B_1D_{11}^{-1}C_1)$. This means that in the unconstrained case necessary and sufficient conditions for closed-loop stability are:

1. the eigenvalues of $(A-B_3F)$ are stricly inside the unit circle.
2. the eigenvalues of $(A-B_1D_{11}^{-1}C_1)$ are strictly inside the unit circle.

Condition (1) can be satisfied by choosing appropriate tuning parameters such that the feedback matrix F makes $|\lambda_{(A-B_3F)}|<1$. We can obtain that by a careful tuning (chapter 7), by introducing the end-point constraint (section 6.1.2) or by extending the prediction horizon to infinity (section 6.1.3).

Condition (2) is related with the choice of the noise model $H(q)$, given by the input output relation (compare 6.1)

$$\begin{aligned}
x(k+1) &= A\,x(k)+B_1\,e(k) \\
y(k) &= C_1\,x(k)+D_{11}\,e(k)
\end{aligned}$$

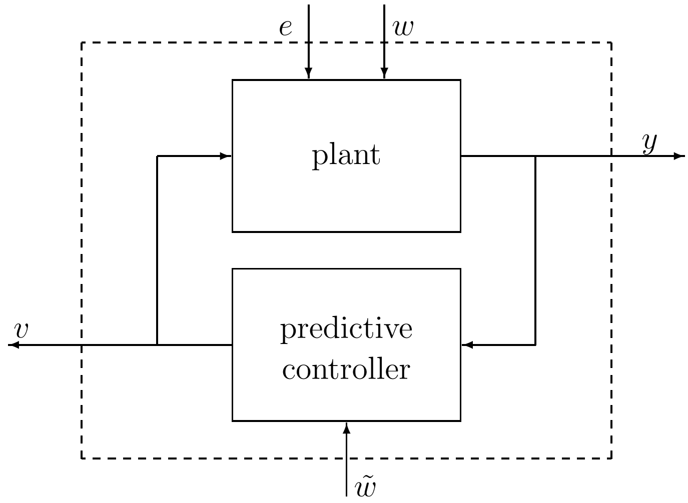

Figure 6.1: Closed loop configuration

with transfer function:

$$H(q) = C_1(qI - A)^{-1}B_1 + D_{11}$$

From appendix B we learn that for

$$H(q) \equiv \left[\begin{array}{c|c} A & B_1 \\ \hline C_1 & D_{11} \end{array}\right]$$

we find

$$H^{-1}(q) \equiv \left[\begin{array}{c|c} A - B_1 D_{11}^{-1} C_1 & B_1 D_{11}^{-1} \\ \hline -D_{11}^{-1} C_1 & D_{11}^{-1} \end{array}\right]$$

so the inverse noise model is given (for $v = 0$ and $w = 0$) by the state space equations:

$$\begin{aligned} x(k+1) &= (A - B_1 D_{11}^{-1} C_1)\, x(k) + B_1 D_{11}^{-1}\, y(k) \\ e(k) &= -D_{11}^{-1} C_1\, x(k) + D_{11}^{-1}\, y(k) \end{aligned}$$

with transfer function:

$$H^{-1}(q) = -D_{11}^{-1} C_1 (qI - A + B_1 D_{11}^{-1} C_1)^{-1} B_1 D_{11}^{-1} + D_{11}^{-1}$$

From these equations it is clear that the eigenvalues of $(A - B_1 D_{11}^{-1} C_1)$ are exactly equal to the poles of the system $H^{-1}(q)$, and thus a necessary condition for closed loop stability is that the inverse of the noise model is stable.

If condition (2) is not satisfied, it means that the inverse of the noise model is not stable. We will have to find a new observer-gain $B_{1,new}$ such that $A - B_{1,new} D_{11}^{-1} C_1$ is stable, without changing the stochastic properties of the noise model. This can be done by a spectral factorization

$$H(q)H^T(q^{-1}) = H_{new}(q)\, H_{new}^T(q^{-1})$$

where H_{new} and H_{new}^{-1} are both stable. $H(q)$ and $H_{new}(q)$ have the same stochastic properties and so replacing H by H_{new} will not affect the properties of the process. The new H_{new} is given as follows:

$$H_{new}(q) = C_1(qI - A)^{-1}B_{1,new} + D_{11}$$

The new observer-gain $B_{1,new}$ is given by:

$$B_{1,new} = (AXC_1^T + B_1D_{11}^{-1})(I + C_1XC_1^T)^{-1}D_{11}$$

where X is the positive definite solution of the discrete algebraic Riccati equation:

$$(A - B_1D_{11}^{-1}C_1)^TX(I + C_1^TC_1X)(A - B_1D_{11}^{-1}C_1) - X = 0$$

6.1.2 End-point constraint

By introducing an end-point constraint, the closed-loop system becomes asymptotically stable. Since the end-point constraint is an equality constraint, it still holds that the resulting controller is linear and time-invariant. The concept is based on the work of Kwon & Pearson ([65]) and the monotonicity of the performance index is used to prove stability. The main idea is to force the system to its steady state at the end of the prediction interval. Consider the steady state, given by the quadruple $(v_{ss}, x_{ss}, w_{ss}, z_{ss})$, satisfying

$$\begin{aligned}
x_{ss} &= A\,x_{ss} + B_2\,w_{ss} + B_3\,v_{ss}\\
z_{ss} &= C_2\,x_{ss} + D_{22}\,w_{ss} + D_{23}\,v_{ss}
\end{aligned}$$

(see section 5.1 for more details). The end-point constraint is given for systems with $z_{ss} = 0$, and so $(v_{ss}, x_{ss}, w_{ss}, z_{ss}) = (v_{ss}, x_{ss}, w_{ss}, 0)$ is a steady state of the system.

Theorem 25 *Consider a LTI system given in the state space description*

$$\begin{aligned}
x(k+1) &= A\,x(k) + B_2\,w(k) + B_3\,v(k)\\
z(k) &= C_2\,x(k) + D_{22}\,w(k) + D_{23}\,v(k)
\end{aligned}$$

with $z_{ss} = 0$ in steady state. A performance index is defined as

$$\min_{\tilde{v}(k)} J(\tilde{v}, k) = \min_{\tilde{v}(k)} \sum_{j=0}^{N-1} \Big(\hat{z}(k+j|k)\Big)^T \Gamma(j) \Big(\hat{z}(k+j|k)\Big) \tag{6.2}$$

where $\Gamma(i) \geq \Gamma(i+1)$ is semi-positive definite for $i = 0, \ldots, N-1$ and an additional constraint is given by:

$$x(k+N) = x_{ss} \tag{6.3}$$

Finally, let $w(k) = w_{ss}$ for $k \geq 0$. Then, the predictive control law, minimizing (6.2), results in a stable closed loop.

proof:
Define the function

$$V(k) = \min_{\tilde{v}(k)} J(\tilde{v}, k)$$

and the optimal vector

$$\tilde{v}^*(k) = \begin{bmatrix} v(^*k|k) \\ \vdots \\ v^*(k+N_c-2|k) \\ v^*(k+N_c-1|k) \\ v_{ss} \\ \vdots \\ v_{ss} \end{bmatrix} = \arg\min_{\tilde{v}(k)} J(\tilde{v}, k)$$

where $v^*(k+j|k)$ means the optimal input value to be applied a time $k+j$ as calculated at time k so

$$V(k) = \sum_{j=0}^{\infty} \Big(\hat{z}^*(k+j|k)\Big)^T \Gamma(j) \Big(\hat{z}^*(k+j|k)\Big)$$

where $\hat{z}^*(k+j|k)$ is the performance signal at time $k+j$ when the optimal input sequence $\tilde{v}^*(k)$ computed at time k is applied. Now define

$$\tilde{v}_{sub}(k+1) = \begin{bmatrix} v^*(k+1|k) \\ \vdots \\ v^*(k+N_c-1|k) \\ v_{ss} \\ v_{ss} \\ \vdots \\ v_{ss} \end{bmatrix}$$

and compute

$$V_{sub}(k+1) = J(\tilde{v}_{sub}, k+1)$$

Applying this above input sequence $\tilde{v}_{sub}(k+1)$ to the system results in a $\hat{z}_{sub}(k+j|k+1) = \hat{z}^*(k+j|k)$ for $j = 1, \ldots, N$ and $\hat{z}_{sub}(k+N+j|k+1) = 0$ for $j > 0$. With this we derive

$$\begin{aligned} V_{sub}(k+1) &= \sum_{j=1}^{N} \Big(\hat{z}_{sub}(k+j+1|k+1)\Big)^T \Gamma(j+1) \Big(\hat{z}_{sub}(k+j+1|k+1)\Big) = \\ &\leq \sum_{j=1}^{N} \Big(\hat{z}_{sub}(k+j+1|k+1)\Big)^T \Gamma(j) \Big(\hat{z}_{sub}(k+j+1|k+1)\Big) = \\ &= \sum_{j=2}^{N} \Big(\hat{z}^*(k+j|k)\Big)^T \Gamma(j) \Big(\hat{z}^*(k+j|k)\Big) = \\ &= V(k) - \Big(\hat{z}^*(k|k)^T \Gamma(1) \hat{z}^*(k|k)\Big) \\ &\leq V(k) \end{aligned}$$

where $\Big(\hat{z}^*(k|k)^T\Gamma(1)\hat{z}^*(k|k)\Big) \geq 0$ because $\Gamma(1)$ is semi-positive definite. Further, because of the receding horizon strategy, we do a new optimization

$$V(k+1) = \min_{\tilde{v}(k+1)} J(\tilde{v}, k+1) \leq J(\tilde{v}_{sub}, k+1)$$

Therefore, $V(k+1) \leq V(k)$ and so $V(k)$ is a Lyapunov function for this system, which proves stability. □ End Proof

Remark 1:
The constraint

$$W\, x(k+N) = W\, x_{ss}$$

where W is a $m \times n$-matrix with full column-rank (so $m \geq n$), is equivalent to the end-point constraint (6.3), because $x(k+N) = x_{ss}$ if $W\, x(k+N) = W\, x_{ss}$. Therefore, the constraint

$$\begin{bmatrix} \hat{y}(k+N|k) \\ \hat{y}(k+N+1) \\ \vdots \\ \hat{y}(k+N+n-1|k) \end{bmatrix} - \begin{bmatrix} y_{ss} \\ \vdots \\ y_{ss} \end{bmatrix} = \begin{bmatrix} C_1 \\ C_1 A \\ \vdots \\ C_1 A^{n-1} \end{bmatrix} x(k+N) - \begin{bmatrix} C_1 \\ C_1 A \\ \vdots \\ C_1 A^{n-1} \end{bmatrix} x_{ss} = 0$$

also results in a stable closed loop if the matrix

$$\begin{bmatrix} C_1 \\ C_1 A \\ \vdots \\ C_1 A^{n-1} \end{bmatrix}$$

has full rank. For SISO systems with pair (C_1, A) observable, we choose $n = dim(A)$, so equal to the number of states. For MIMO systems, there may be an integer value $n < dim(A)$ for which W has full column-rank.

The end-point constraint was first introduced by (Clarke & Scattolini [23]), (Mosca & Zhang [85]) and forces the output $y(k)$ to match the reference signal $r(k) = r_{ss}$ at the end of the prediction horizon:

$$y(k+N+j) = r_{ss} \;\text{ for }\; j = 1, \ldots, n$$

With this additional constraint the closed-loop system is asymptotically stable.

Remark 2:
Although stability is guaranteed, the use of an end-point constraint has some disadvantages: The end-point constraint is only a sufficient, but not a necessary condition for stability. The constraint may be too restrictive and can lead to infeasibility, especially if the horizon N is chosen too small. This will be discussed in section 6.2.

The tuning rules, as will discussed in the next chapter, change. This has to do with the fact that because of the control horizon N_c the output is already forced to its steady state at time N_c and so $N = N_c$ will give the same result as any choice of $N > N_c$. The choice of the prediction horizon N is overruled by the choice of control horizon N_c and we loose a degree of freedom.

6.1.3 Infinite prediction horizon

A straight forward way to guarantee stability in predictive control is to choose an infinite prediction horizon ($N = \infty$). It is easy to show that the controller is stabilizing for the unconstrained case, which is formulated in the following theorem:

Theorem 26 *Consider a LTI system given in the state space description*

$$\begin{aligned} x(k+1) &= A\,x(k) + B_2\,w(k) + B_3\,v(k) && (6.4) \\ z(k) &= C_2\,x(k) + D_{22}\,w(k) + D_{23}\,v(k) && (6.5) \end{aligned}$$

with $z_{ss} = 0$ in steady state. Define the set

$$\tilde{v}(k) = \begin{bmatrix} v(k|k) \\ \vdots \\ v(k+N_c-1|k) \end{bmatrix}$$

where $N_c \in \mathbb{Z}^+ \cup \{\infty\}$. A performance index is defined as

$$\min_{\tilde{v}(k)} J(\tilde{v},k) = \min_{\tilde{v}(k)} \sum_{j=0}^{\infty} \Big(\hat{z}(k+j|k)\Big)^T \Gamma(j) \Big(\hat{z}(k+j|k)\Big)$$

where $\Gamma(i) \geq \Gamma(i+j) \geq 0$ for $j \geq 0$. The predictive control law minimizing this performance index results in a stabilizing controller.

Proof:
Define the function

$$V(k) = \min_{\tilde{v}(k)} J(\tilde{v},k)$$

and the optimal vector

$$\tilde{v}^*(k) = \arg\min_{\tilde{v}(k)} J(k)$$

where

$$\tilde{v}^*(k) = \begin{bmatrix} v^*(k|k) \\ v^*(k+1|k) \\ \vdots \\ v^*(k+N_c-1|k) \end{bmatrix} \quad \text{for } N_c < \infty$$

or

$$\tilde{v}^*(k) = \begin{bmatrix} v^*(k|k) \\ v^*(k+1|k) \\ \vdots \end{bmatrix} \text{ for } N_c = \infty$$

where $v^*(k+j|k)$ means the optimal input value to be applied a time $k+j$ as calculated at time k, so

$$V(k) = \sum_{j=0}^{\infty} \Big(\hat{z}^*(k+j|k)\Big)^T \Gamma(j) \Big(\hat{z}^*(k+j|k)\Big)$$

where $\hat{z}^*(k+j|k)$ is the performance signal at time $k+j$ when the optimal input sequence $\tilde{v}^*(k)$ computed at time k is applied. Now define for $N_c < \infty$:

$$\tilde{v}_{sub}(k+1) = \begin{bmatrix} v^*(k+1|k) \\ v^*(k+2|k) \\ \vdots \\ v^*(k+N_c-1|k) \\ v_{ss} \end{bmatrix} \text{ for } N_c < \infty$$

or

$$\tilde{v}_{sub}(k+1) = \begin{bmatrix} v^*(k+1|k) \\ v^*(k+2|k) \\ \vdots \end{bmatrix} \text{ for } N_c = \infty$$

and compute

$$V_{sub}(k+1) = J(\tilde{v}_{sub}, k+1)$$

then this value is equal to

$$V_{sub}(k+1) = \sum_{j=1}^{\infty} \Big(\hat{z}^*(k+j|k)\Big)^T \Gamma(j) \Big(\hat{z}^*(k+j|k)\Big)$$

and so

$$V_{sub}(k+1) \leq V(k)$$

further

$$V(k+1) = \min_{\tilde{v}(k+1)} J(\tilde{v}, k+1) \leq J(\tilde{v}_{sub}, k+1)$$

therefore $V(k+1) \leq V(k)$ and so $V(k)$ is a Lyapunov function for this system, which proves stability. □ End Proof

As is shown above, for stability it is not important if $N_c = \infty$ or $N_c < \infty$. For $N_c = \infty$ the predictive controller becomes equal to the optimal LQ-solution (Bitmead, Gevers & Wertz [13], see section 5.3.2). The main reason not to choose an infinite control horizon is the fact that constraint-handling on an infinite horizon is an extremely hard problem. Rawlings & Muske ([96]) studied the case where the prediction horizon is equal to infinity ($N = \infty$), but the control horizon is finite ($N_c < \infty$), see the sections 5.3.3 and 5.3.4. Because of the finite number of control actions that can be applied, the constraint-handling has become a finite-dimensional problem instead of an infinte-dimensional problem for $N_c = \infty$.

6.2 Inequality constrained case

The issue of stability becomes even more complicated in the case where constraints have to be satisfied. When there are only constraints on the control signal, a stability guarantee can be given if the process itself is stable (Sontag [111]), (Balakrishnan *et al.* [10]).
In the general case with constraints on input, output and states, the main problem is feasibility. The existence of a stabilizing control law is not at all trivial. The two main results on stability in constrained predictive control are the papers by Rawlings & Muske [96] and Kothare *et al.* [63].
The work of Rawlings & Muske is based on an infinite prediction horizon as discussed in the sections 5.3.3 and 5.3.4. They show that for fixed constraints on states, input signals and output signals, and a constant reference signal $r(k) = r_{ss}$, stability and feasibility can be proven. The work of Kothare et al. [63] is based on solving the predictive control problem with a state feedback law using Linear Matrix inequalities. This will be discussed in section 8.3.
An important issue in stability of inequality constrained MPC is what the influence of the present input sequence $\tilde{v}(k)$ will have on future inequality constraints. An useful property of inequality constraints is shift-invariance:

Definition 27 ***shift-invariant constraint***
Let $\tilde{v}_{sub}(k+1)$ be the shifted version of $\tilde{v}(k)$, completed with v_{ss} as defined in theorem 25 (in the end-point constrained case) or theorem 26 (for the infinite horizon case) for the noiseless case (so $e(k) = 0$). Consider constraint

$$\tilde{\psi}(k) = C_4 x(k) + D_{42}\tilde{w}(k) + D_{43}\tilde{v}(k) \leq \tilde{\Psi}(k) \qquad (6.6)$$

This constraint is shift-invariant if, for any $\tilde{v}(k)$ satisfying (6.6), there holds

$$\tilde{\psi}(k+1) = C_4 x(k+1) + D_{42}\tilde{w}(k+1) + D_{43}\tilde{v}_{sub}(k+1) \leq \tilde{\Psi}(k+1)$$

where $\tilde{w}(k+1)$ is the shifted version of $\tilde{w}(k)$, completed with w_{ss}.

Based on the definition of a shift-invariant constraint, we can formulate the following lemma:

Lemma 28 *Consider SPCP problem as in theorem 25 and 26. Let constraint (6.6) be shift-invariant. Then, the predictive control law, minimizing (6.2) subject to (6.6), results in a stable closed loop.*

The proof to the problem follows the proofs of theorem 25 and 26, where we proof that if $v^*(k)$ is an optimal solution at time k, then $v_{sub}(k+1)$ satisfies constraint (6.6) at time $k+1$ and gives a lower value of the performance index $J_{sub}(k+1)$.

6.3 Relation to IMC scheme and Youla parametrization

6.3.1 The IMC scheme

The Internal Model Control (IMC) scheme provides a practical structure to analyze properties of closed-loop behaviour of control systems and is closely related to model predictive control [48], [83]. In this section we will show that for stable processes, the solution of the SPCP (as derived in chapter 5) can be rewritten in the IMC configuration.
The IMC structure is given in figure 6.2 and consists of the process $G(q)$, the model $\hat{G}(q)$, the (stable) inverse of the noise model $\hat{H}^{-1}$ and the two-degree-of-freedom controller Q. For simplicity reasons we will consider $F(q) = 0$ (known disturbances are not taken into account), but the extension is straight forward. The blocks in the IMC scheme satisfy the

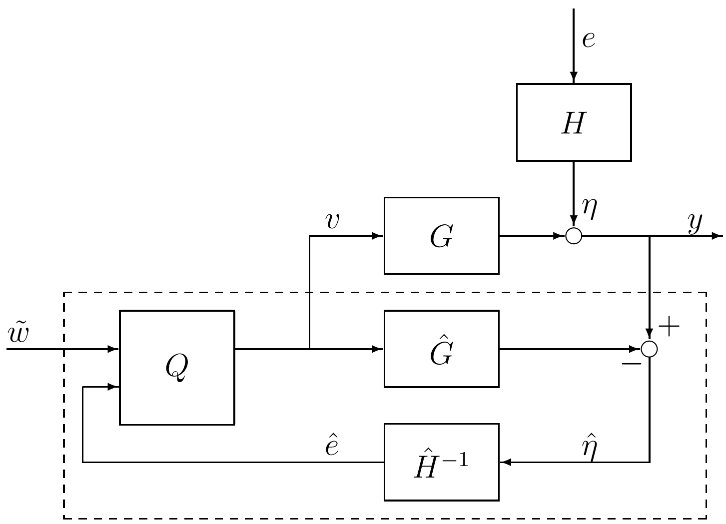

Figure 6.2: Internal Model Control Scheme

following relations:

Process:

$$y(k) = G(q)v(k) + H(q)e(k) \tag{6.7}$$

Process model:

$$\hat{y}(k) = \hat{G}(q)v(k) \tag{6.8}$$

Pre-filter:

$$\hat{e}(k) = \hat{H}^{-1}(q)(y(k) - \hat{y}(k)) \tag{6.9}$$

Controller:

$$v(k) = Q(q)\begin{bmatrix} \tilde{w}(k) \\ \hat{e}(k) \end{bmatrix} = Q_1(q)\tilde{w}(k) + Q_2(q)\,\hat{e}(k) \tag{6.10}$$

Proposition 29 *Consider the IMC scheme, given in figure 6.2, satisfying equations (6.7)-(6.10), where G, H and H^{-1} are stable. If the process and noise models are perfect,so if $\hat{G}(q) = G(q)$ and $\hat{H}(q) = H(q)$, the closed loop description is given by:*

$$\begin{aligned} v(k) &= Q_1(q)\tilde{w}(k) + Q_2(q)e(k) && (6.11) \\ y(k) &= G(q)Q_1(q)\tilde{w}(k) + \Big(G(q)Q_2(q) + H(q)\Big)e(k) && (6.12) \end{aligned}$$

The closed loop is stable if and only if Q is stable.

Proof:
The description of the closed loop system is given by substitution of (6.7)-(6.9) in (6.7) and (6.10):

$$\begin{aligned} v(k) &= \Big(I - Q_2(q)\hat{H}^{-1}(q)(G(q) - \hat{G}(q))\Big)^{-1}\Big(Q_1(q)\tilde{w}(k) + Q_2(q)\hat{H}^{-1}(q)H(q)e(k)\Big) \\ y(k) &= G(q)\Big(I - Q_2(q)\hat{H}^{-1}(q)(G(q) - \hat{G}(q))\Big)^{-1}\Big(Q_1(q)\tilde{w}(k) \\ &\quad + Q_2(q)\hat{H}^{-1}(q)H(q)e(k)\Big) + H(q)e(k) \end{aligned}$$

For $G = \hat{G}$ and $H = \hat{H}$ this simplifies to equations (6.11) and (6.12). It is clear that from equations (6.11) and (6.12) that if Q is stable, the closed loop system is stable. On the other hand, if the closed loop is stable, we can read from equation (6.11) that Q_1 and Q_2 are stable.

□ End Proof

The MPC controller in an IMC configuration

The next step is to derive the function Q for our predictive controller. To do so, we need to find the relation between the IMC controller and the common two-degree of freedom controller $C(q)$, given by:

$$v(k) = C_1(q)\tilde{w}(k) + C_2(q)y(k) \tag{6.13}$$

$$\tag{6.14}$$

From equations (6.8)-(6.10) we derive:

$$\begin{aligned} v(k) &= Q_1(q)\tilde{w}(k) + Q_2(q)\,\hat{e}(k) \\ &= Q_1(q)\tilde{w}(k) + Q_2(q)\,\hat{H}^{-1}(q)(y(k) - \hat{y}(k)) \\ &= Q_1(q)\tilde{w}(k) + Q_2(q)\,\hat{H}^{-1}(q)y(k) - Q_2(q)\,\hat{H}^{-1}(q)\hat{G}(q)v(k) \\ \Big(I + Q_2(q)\,\hat{H}^{-1}(q)\hat{G}(q)\Big)v(k) &= Q_1(q)\tilde{w}(k) + Q_2(q)\,\hat{H}^{-1}(q)y(k) \end{aligned}$$

so

$$\begin{aligned} v(k) &= \Big(I + Q_2\,\hat{H}^{-1}(q)\hat{G}(q)\Big)^{-1} Q_1(q)\tilde{w}(k) \\ &\quad + \Big(I + Q_2\,\hat{H}^{-1}(q)\hat{G}(q)\Big)^{-1} Q_2(q)\,\hat{H}^{-1}(q)y(k) \end{aligned}$$

This results in:

$$\begin{aligned} C_1(q) &= \Big(I + Q_2(q)\,\hat{H}^{-1}(q)\hat{G}(q)\Big)^{-1} Q_1(q) \\ C_2(q) &= \Big(I + Q_2(q)\,\hat{H}^{-1}(q)\hat{G}(q)\Big)^{-1} Q_2(q)\,\hat{H}^{-1}(q) \\ &= Q_2(q)\Big(\hat{H}(q) + \hat{G}(q)Q_2(q)\Big)^{-1} \end{aligned}$$

with the inverse relation:

$$\begin{aligned} Q_1(q) &= \Big(I - C_2(q)\hat{G}(q)\Big)^{-1} C_1(q) \\ Q_2(q) &= \Big(I - C_2(q)\hat{G}(q)\Big)^{-1} C_2(q)\hat{H}(q) \end{aligned}$$

In state space we can describe the systems $\hat{G}$, $\hat{H}^{-1}$ and $Q(q)$ as follows:

Process model $\hat{G}(q)$:

$$\begin{aligned} x_1(k+1) &= A\,x_1(k) + B_3\,v(k) \\ \hat{y}(k) &= C_1\,x_1(k) \end{aligned}$$

Pre-filter $\hat{H}^{-1}(q)$:

$$\begin{aligned} x_2(k+1) &= (A - B_1\,D_{11}^{-1}\,C_1)\,x_2(k) + B_1\,D_{11}^{-1}\,(y(k) - \hat{y}(k)) \\ \hat{e}(k) &= -D_{11}^{-1}\,C_1\,x_2(k) + D_{11}^{-1}(y(k) - \hat{y}(k)) \end{aligned}$$

Controller $Q(q)$:

$$x_3(k+1) = (A - B_3\,F)\,x_3(k) + B_3 D_w \tilde{w}(k) + (B_1 + B_3\,D_e)\,\hat{e}(k) \tag{6.15}$$

$$v(k) = -F\,x_3(k) + D_w\,\tilde{w}(k) + D_e\,\hat{e}(k) \tag{6.16}$$

The state equations of x_1 and x_2 follow from chapter 2. We will now show that the above choice of $Q(q)$ will indeed lead to the same control action as the controller derived in chapter 5.
Combining the three state equations in one gives:

$$\begin{aligned} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} &= \begin{bmatrix} A - B_3 D_e D_{11}^{-1} C_1 & -B_3 D_e D_{11}^{-1} C_1 & -B_3 F \\ -B_1 D_{11}^{-1} C_1 & A - B_1 D_{11}^{-1} C_1 & 0 \\ -B_1 D_{11}^{-1} C_1 - B_3 D_e D_{11}^{-1} C_1 & -B_1 D_{11}^{-1} C_1 - B_3 D_e D_{11}^{-1} C_1 & A - B_3 F \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} B_3 D_e D_{11}^{-1} \\ B_1 D_{11}^{-1} \\ B_1 D_{11}^{-1} + B_3 D_e D_{11}^{-1} \end{bmatrix} y(k) + \begin{bmatrix} B_3 D_w \\ 0 \\ B_3 D_w \end{bmatrix} \tilde{w}(k) \end{aligned}$$

$$v(k) = \begin{bmatrix} -D_e D_{11}^{-1} C_1 & -D_e D_{11}^{-1} C_1 & -F \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} D_e D_{11}^{-1} \end{bmatrix} y(k) + \begin{bmatrix} D_w \end{bmatrix} \tilde{w}(k)$$

We apply a state transformation:

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} x_1 \\ x_1 + x_2 - x_3 \\ x_3 \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ I & I & -I \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

This results in

$$\begin{bmatrix} x_1'(k+1) \\ x_2'(k+1) \\ x_3'(k+1) \end{bmatrix} = \begin{bmatrix} A & -B_3 D_e D_{11}^{-1} C_1 & -B_3 F - B_3 D_e D_{11}^{-1} C_1 \\ 0 & A & 0 \\ 0 & -B_1 D_{11}^{-1} C_1 - B_3 D_e D_{11}^{-1} C_1 & A - B_3 F - B_1 D_{11}^{-1} C_1 - B_3 D_e D_{11}^{-1} C_1 \end{bmatrix} \begin{bmatrix} x_1'(k) \\ x_2'(k) \\ x_3'(k) \end{bmatrix}$$
$$+ \begin{bmatrix} B_3 D_e D_{11}^{-1} \\ 0 \\ B_1 D_{11}^{-1} + B_3 D_e D_{11}^{-1} \end{bmatrix} y(k) + \begin{bmatrix} B_3 D_w \\ 0 \\ B_3 D_w \end{bmatrix} \tilde{w}(k)$$
$$v(k) = \begin{bmatrix} 0 & -D_e D_{11}^{-1} C_1 & -F - D_e D_{11}^{-1} C_1 \end{bmatrix} \begin{bmatrix} x_1'(k) \\ x_2'(k) \\ x_3'(k) \end{bmatrix} + \begin{bmatrix} D_e \end{bmatrix} y(k) + \begin{bmatrix} D_w \end{bmatrix} \tilde{w}(k)$$

It is clear that state x_1' is not observable and state x_2' is not controlable. Since A has all eigenvalues inside the unit disc, we can do model reduction by deleting these states and we obtain the reduced controller:

$$\begin{aligned} x_3'(k+1) &= (A - B_3 F - B_1 D_{11}^{-1} C_1 - B_3 D_e D_{11}^{-1} C_1)\, x_3'(k) \\ &\quad + (B_1 D_{11}^{-1} + B_3 D_e D_{11}^{-1})\, y(k) + B_3 D_w\, \tilde{w}(k) \\ v(k) &= (-F - D_e D_{11}^{-1} C_1)\, x_3(k) + D_e D_{11}^{-1}\, y(k) + D_w\, \tilde{w}(k) \end{aligned}$$

This is exactly equal to the optimal predictive controller, derived in the equations (5.143) and (5.144).

Remark: Note that a condition of a stable Q is equivalent the condition that all eigenvalues of $(A - B_3 F)$ are strictly inside the unit circle. (The condition a stable inverse H^{-1} is equivalent to the condition that all eigenvalues of $(A - B_1 D_{11}^{-1} C_1)$ are strictly inside the unit circle.)

Example 30 : computation of function $Q_1(q)$ and $Q_2(q)$ for a given MPC controller

Consider the system of example 10 on page 82 with the MPC controller computed in example 23 on page 126. Using equations 6.15 and 6.16 we can give the functions $Q_1(q)$ and $Q_2(q)$ as follows:

$$Q_1(q) \equiv \left[\begin{array}{c|c} A - B_3\, F & B_3 D_w \\ \hline -F & D_w \end{array}\right]$$

$$Q_2(q) \equiv \left[\begin{array}{c|c} A - B_3\, F & (B_1 + B_3\, D_e) \\ \hline -F & D_e \end{array}\right]$$

or in polynomial form:

$$Q_1(q) = \frac{-0.0831(z-1)(z-0.5)(z-0.2)}{(z^3 - 1.3323\, z^2 + 0.4477\, z)}$$

$$Q_2(q) = \frac{-0.6339(z^2 - 1.0534\, z + 0.3265)(z-0.2)}{(z^3 - 1.3323\, z^2 + 0.4477\, z)}$$

6.3.2 The Youla parametrization

The IMC scheme provides a nice framework for the case where the process is stable. For the unstable case we can use the Youla parametrization [117], [87], [31]. We consider process $G(q)$, noise filter $H(q)$, satisfying

$$y(k) = G(q)\, v(k) + H(q)\, e(k) \tag{6.17}$$

For simplicity reasons we will consider $F(q) = 0$ (known disturbances are not taken into account). The extension for $F(q) \neq 0$ is straight forward.

Consider square transfer matrices

$$\Lambda(q) = \begin{bmatrix} M(q) & -N(q) \\ X(q) & Y(q) \end{bmatrix} \qquad \Lambda^{-1}(q) = \tilde{\Lambda} = \begin{bmatrix} \tilde{Y}(q) & -\tilde{X}(q) \\ \tilde{N}(q) & \tilde{M}(q) \end{bmatrix}$$

satisfying the following properties:

1. The system $\Lambda(q)$ is stable.
2. The inverse system $\Lambda^{-1}(q)$ is stable.
3. $M(q)$, $\tilde{M}(q)$ $Y(q)$ and $\tilde{Y}(q)$ are square and invertible systems.
4. $G(q) = M^{-1}(q)N(q) = \tilde{N}(q)\tilde{M}^{-1}(q)$.

The above factorization is denoted as a doubly coprime factorization of $G(q)$. Now consider a controller given by the following parametrization, denoted as the Youla parametrization:

$$\begin{aligned} v(k) &= \Big(Y(q) + Q_2(q)N(q)\Big)^{-1}\Big(X(q) + Q_2(q)M(q)\Big)y(k) + \\ &\quad + \Big(Y(q) + Q_2(q)N(q)\Big)^{-1} Q_1(q)\tilde{w}(k) \end{aligned} \tag{6.18}$$

where $M(q)$, $N(q)$, $X(q)$ and $Y(q)$ are found from a doubly coprime factorization of plant model $G(q)$ and $Q_1(q)$ and $Q_2(q)$ are transfer functions with the appropriate dimensions. For this controller we can make the following proposition:

Proposition 31 *Consider the process (6.17) and the Youla parametrization (6.18). The closed loop description is then given by:*

$$v(k) = \tilde{M}(q)Q_1(q)\tilde{w}(k) + \tilde{M}(q)Q_2(q)X(q)H(q)e(k) \tag{6.19}$$
$$y(k) = \tilde{N}(q)Q_1(q)\tilde{w}(k) + \Big(\tilde{N}(q)Q_2(q)X(q) + I\Big)H(q)e(k) \tag{6.20}$$

The closed loop is stable if and only if both Q_1 and Q_2 are stable.

The closed loop of plant model $G(q)$, noise model $H(q)$ together with the Youla-based controller is visualised in figure 6.3.

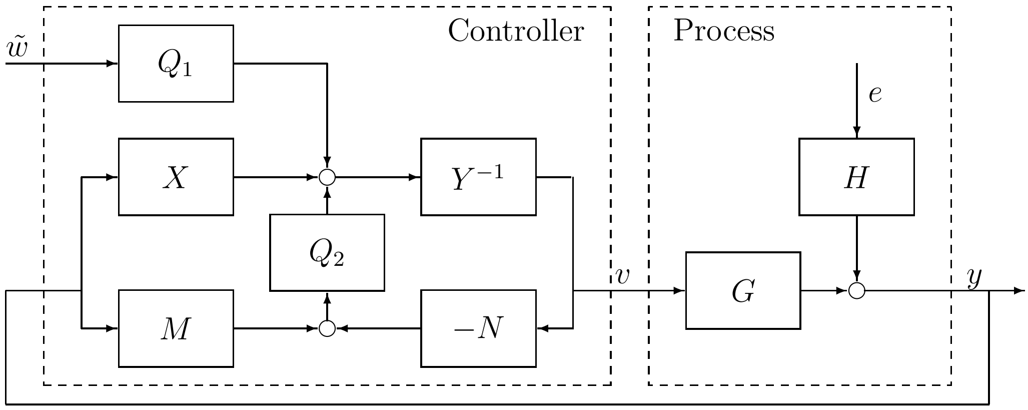

Figure 6.3: Youla parametrization

The Youla parametrization (6.18) gives all controllers stabilizing $G(q)$ for stable $Q_1(q)$ and $Q_2(q)$. For a stabilizing MPC controller we can compute the corresponding functions $Q_1(q)$ and $Q_2(q)$:

The MPC controller in a Youla configuration

The decomposition of $G(q)$ in two stable functions $M(q)$ and $N(q)$ (or coprime factorization) is not unique. We will consider two possible choices for the factorization:

Factorization, based on the inverse noise model:
A possible state-space realization of $\Lambda(q)$ is the following

$$\Lambda_{ij}(q) = C_{\Lambda,i}(qI - A_\Lambda)^{-1}B_{\Lambda,j} + D_{\Lambda,ij}$$

for

$$\left[\begin{array}{c|cc} A_\Lambda & B_{\Lambda,1} & B_{\Lambda,2} \\ \hline C_{\Lambda,1} & D_{\Lambda,11} & D_{\Lambda,12} \\ C_{\Lambda,2} & D_{\Lambda,21} & D_{\Lambda,22} \end{array}\right] = \left[\begin{array}{c|cc} A - B_1D_{11}^{-1}C_1 & B_1D_{11}^{-1} & B_3 \\ \hline -C_1D_{11}^{-1} & I & 0 \\ -F & 0 & I \end{array}\right]$$

And we choose

$$\begin{aligned} Q_1 &= D_w \\ Q_2 &= D_e \end{aligned}$$

The poles of $\Lambda(q)$ are equal to the eigenvalues of the matrix $A - B_1 D_{11}^{-1} C_1$, so equal to the poles of the inverse noise model $H^{-1}(q)$. The poles of $\Lambda^{-1}(q)$ are equal to the eigenvalues of $A - B_3 F$. So, if the eigenvalues of $(A - B_1 D_{11}^{-1} C_1)$ and $(A - B_3 F)$ are inside the unit circle, the controller will be stabilizing for any stable Q_w and Q_e. In our case $Q_1 = D_w$ and $Q_2 = D_e$ are constant matrices and thus stable.

Factorization, based on a stable plant model:
If G, H and H^{-1} are stable, we can also make another choice for Λ:

$$\Lambda(q) = \begin{bmatrix} M(q) & -N(q) \\ X(q) & Y(q) \end{bmatrix} = \begin{bmatrix} H^{-1}(q) & -H^{-1}(q)G(q) \\ 0 & I \end{bmatrix}$$

$\Lambda(q)$ is stable, and so is $\Lambda^{-1}(q)$, given by:

$$\Lambda^{-1}(q) = \begin{bmatrix} H(q) & G(q) \\ 0 & I \end{bmatrix}$$

By comparing Figure 6.2 and Figure 6.3 we find that for the above choices the Youla parametrization boils down to the IMC scheme by choosing $Q_{YOULA} = Q_{IMC}$. We find that the IMC scheme is a special case of the Youla parametrization.

Chapter 7

Tuning

As was already discussed in the introduction, one of the main reasons for the popularity of predictive control is the easy tuning. The purpose of tuning the parameters is to acquire good signal tracking, sufficient disturbance rejection and robustness against model mismatch.

Consider the LQPC-performance index

$$J(u,k) = \sum_{j=N_m}^{N} \hat{x}^T(k+j|k) Q \hat{x}(k+j|k) + \sum_{j=1}^{N} u^T(k+j-1) R u(k+j-1)$$

and GPC-performance index

$$J(u,k) = \sum_{j=N_m}^{N} \Big(\hat{y}_p(k+j|k) - r(k+j)\Big)^T \Big(\hat{y}_p(k+j|k) - r(k+j)\Big) +$$

$$+\lambda^2 \sum_{j=1}^{N_c} \Delta u^T(k+j-1) \Delta u(k+j-1)$$

The tuning parameters can easily be recognized. They are:

$$\begin{array}{rcl}
N_m & = & \text{minimum-cost horizon} \\
N & = & \text{prediction horizon} \\
N_c & = & \text{control horizon} \\
\lambda & = & \text{weighting factor (GPC)} \\
P(q) & = & \text{tracking filter (GPC)} \\
Q & = & \text{state weighting matrix (LQPC)} \\
R & = & \text{control weighting matrix (LQPC)}
\end{array}$$

This chapter discusses the tuning of a predictive controller. In the section 7.1 some rules of thumb are given for the initial parameter settings. In section 7.2 we look at the case where the initial controller does not meet the desired specifications. An advanced tuning procedure may provide some tools that lead to a better and suitable controller. The final fine-tuning of a predictive controller is usually done on-line when the controller is already running. Section 7.3 discusses some special settings of the tuning parameters, leading to well known controllers.

In this chapter we will assume that the model is perfect. Robustness against model mismatch will be discussed in chapter 9.

7.1 Initial settings for the parameters

For GPC the parameters N, N_c and λ are the three basic tuning parameters, and most papers on GPC consider the tuning of these parameters. The rules-of-thumb for the settings of a GPC-controller were first discussed by Clarke & Mothadi ([20]) and later reformulated for the more extended UPC-controller in Soeterboek ([108]). In LQPC the parameters N and N_c are present, but instead of λ and $P(q)$, we have the weighting matrices R and Q. In Lee & Yu ([70]), the tuning of the LQPC parameters is considered.
In the next sections the initial setting for the summation parameters (N_m, N, N_c) and the signal weighting parameters ($P(q)$, λ / Q, R) are discussed.

7.1.1 Initial settings for the summation parameters

In both the LQPC-performance index and the GPC-performance index, three summation parameters, N_m, N and N_c are recognized, which can be used to tune the predictive controller. Often we choose $N_m = 1$, which is the best choice most of the time, but may be chosen larger in the case of a dead-time or an inverse response.
The parameter N is mostly related to the length of the step response of the process, and the prediction interval N should contain the crucial dynamics of $x(k)$ due to the response on $u(k)$ (in the LQPC-case) or the crucial dynamics of $y(k)$ due to the response on $\Delta u(k)$ (in the GPC-case).
The number $N_c \leq N$ is called the control horizon which forces the control signal to become constant

$$u(k+j) = constant \quad \text{for} \quad j \geq N_c$$

or equivalently the increment input signal to become zero

$$\Delta u(k+j) = 0 \quad \text{for} \quad j \geq N_c$$

An important effect of a small control horizon ($N_c \ll N$) is the smoothing of the control signal (which can become very wild if $N_c = N$) and the control signal is forced towards its steady-state value, which is important for stability-properties. Another important consequence of decreasing N_c is the reduction in computational effort. Let the control signal $v(k)$ be a $p \times 1$ vector. Then the control vector $\tilde{v}(k)$ is a $pN \times 1$ vector, which means we have pN degrees of freedom in the optimization. By introducing the control horizon $N_c < N$, these degrees of freedom decrease because of the equality constraint, as was shown in chapter 4 and 5. If the control horizon is the only equality constraint, the new optimization parameter is $\tilde{\mu}$, which is a $pN_c \times 1$ vector. Thus the degrees of freedom reduces to pN_c. This can be seen by observing that the matrix $\tilde{D}_{33}^{\perp}$ is an $pN \times pN_c$ matrix, and so the matrix $H = 2\,(\tilde{D}_{33}^{\perp})^T \tilde{D}_{23}^T \bar{\Gamma} \tilde{D}_{23} \tilde{D}_{33}^{\perp}$ becomes an $pN_c \times pN_c$ matrix. For the case without further inequality constraints, this matrix H has to be inverted, which is easier to invert for smaller N_c.

If we add inequality constraints we have to minimize

$$\min_{\tilde{\mu}} \frac{1}{2} \tilde{\mu}^T \, H \, \tilde{\mu}$$

subject to these constraints. The number of parameters in the optimization reduces from pN to pN_c, which will speed up the optimization.

Consider a process where

$$\begin{aligned}
d &= \text{the dead time of the process}\\
n &= \text{the number of poles of the model}\\
&= \text{the dimension of the matrix } A_o \text{ for an IO model}\\
&= \text{the order of polynomial } a_o(q) \text{ for an IO model}\\
&= \text{the dimension of the matrix } A_i \text{ for an IIO model}\\
&= \text{the order of polynomial } a_i(q) \text{ for an IIO model}\\
t_s &= \text{the 5\% settling time of the process } G_o(q)\\
\omega_s &= \text{the sampling-frequency}\\
\omega_b &= \text{the bandwidth of the process } G_{c,o}(s)
\end{aligned}$$

where $G_{c,o}(s)$ is the original IO process model in continuous-time (corresponding to the discrete-time IO process model $G_o(q)$). Now the following initial setting are recommended (Soeterboek [81]):

$$\begin{aligned}
N_m &= 1 + d\\
N_c &= n\\
N &= \text{int}(\alpha_N \, t_s) \;\; \text{for well-damped systems}\\
N &= \text{int}(\beta_N \, \omega_s/\omega_b) \;\; \text{for badly-damped and unstable systems}
\end{aligned}$$

where $\alpha_N \in [1.5, 2]$ and $\beta_N \in [4, 25]$. Resuming, N_m is equal to 1 + the process dead-time estimate and N should be chosen larger than the length of the step response of the open-loop process. When we use an IIO model, the value of N should be based on the original IO-model ($G_o(q)$ or $G_{c,o}(s)$).

The bandwidth ω_b of a system with transfer function $G_{c,o}(s)$ is the frequency at which the magnitude ratio drops to $1/\sqrt{2}$ of its zero-frequency level (given that the gain characteristic is 'flat' at zero frequency). So:

$$\omega_b \;=\; \max_{\omega in \mathbb{R}} \left\{ \omega \;\middle|\; |G_{c,o}(j\omega)|^2/|G_{c,o}(0)|^2 \geq 0.5 \right\}$$

For badly-damped and unstable systems, the above choice of $N = 2\,\mathrm{int}(2\omega_s/\omega_b)$ is a lower bound. Often increasing N may improve the stability and the performance of the closed loop.

In the constrained case N_c will usually be chosen bigger, to introduce extra degrees of freedom in the optimization. A reasonable choice is

$$n \leq N_c \leq N/2$$

If signal constraints get very tight, N_c can be even bigger than $N/2$. (see elevator example in chapter 5).

7.1.2 Initial settings for the signal weighting parameters

The GPC signal weighting parameters:

λ should be chosen as small as possible, 0 in most cases. In the case of non-minimum phase systems, $\lambda = 0$ will lead to stability problems and so λ should be chosen as small as possible. The parameters p_i, $i = 1, \ldots, n_p$ of filter

$$P(q) = 1 + p_1 q^{-1} + p_2 q^{-2} + \cdots + p_{n_p} q^{-n_p}$$

are chosen such that the roots of the polynomial $P(q)$ are the desired poles of the closed loop. $P(q)$ makes $y(k)$ to track the low-pass filtered reference signal $P^{-1}(q)r(k)$.

The LQPC signal weighting parameters

In the SISO case, the matrix Q is a $n \times n$ matrix and R will be a scalar. When the purpose is to get the output signal $y(k)$ to zero as fast as possible we can choose

$$Q = C_1^T C_1 \qquad \text{and} \qquad R = \lambda^2 I$$

which makes the LQPC performance index equivalent to the GPC index for an IO model, $r(k) = 0$ and $P(q) = 1$. This can be seen by substitution:

$$\begin{aligned} J(u,k) &= \sum_{j=N_m}^{N} \hat{x}^T(k+j|k) Q \hat{x}(k+j|k) + \sum_{j=1}^{N_c} u^T(k+j-1) R u(k+j-1) \\ &= \sum_{j=N_m}^{N} \hat{x}^T(k+j|k) C_1^T C_1 \hat{x}(k+j|k) + \lambda^2 \sum_{j=1}^{N_c} u^T(k+j-1) u(k+j-1) \end{aligned}$$

$$= \sum_{j=N_m}^{N} \hat{y}^T(k+j|k)\hat{y}(k+j|k) + \lambda^2 \sum_{j=1}^{N_c} u^T(k+j-1)u(k+j-1)$$

The rules for the tuning of λ are the same as for GPC. So, $\lambda = 0$ is a obvious choice for minimum phase systems, λ is small for non-minimum phase systems. By adding an term Q_1 to the weighting matrix

$$Q = C_1^T C_1 + Q_1$$

we can introduce an additional weighting $\hat{x}^T(k+j|k)Q_1\hat{x}(k+j|k)$ on the states.
In the MIMO case it is important to scale the input and output signals. To make this clear, we take an example from the paper industry where we want to produce a paper roll with a specific thickness (in the order of 80 μm) and width (for A1-paper in the order of 84 cm), Let $y_1(k)$ represent the thickness-error (defined in meters) and $y_2(k)$ the width-error (also defined in meters). If we define an error like

$$J = \sum |y_1(k)|^2 + |y_2(k)|^2$$

it is clear that the contribution of the thickness-error to the performance index can be neglected in comparison to the contribution due to the width-error. This means that the error in paper thickness will be neglected in the control procedure and all the effort will be put in controlling the width of the paper. We should not be surprised that using the above criterion could result in paper with variation in thickness from, say 0 to 200 μm. The introduction of scaling gives a better balance in the effort of minimizing both width-error and thickness-error. In this example let us require a precision of 1% for both outputs, then we can introduce the performance index

$$J = \sum |y_1(k)/8 \cdot 10^{-7}|^2 + |y_2(k)/8.4 \cdot 10^{-3}|^2$$

Now the relative error of both thickness and width are measured in the same way.

In general, let the required precision for the i-th output be given by d_i, so $|\, y_i \,| \leq d_i$, for $i = 1, \ldots, m$. Then a scaling matrix can be given by

$$S = \mathrm{diag}(d_1, d_2, \ldots, d_m)$$

By choosing

$$Q = C_1^T S^{-2} C_1$$

the first term of the LQPC performance index will consist of

$$\hat{x}^T(k+j|k)Q\hat{x}(k+j|k) = \hat{x}^T(k+j|k)C_1^T S^{-2} C_1 \hat{x}(k+j|k) = \sum_{i=1}^{m} |\hat{y}_i(k+j|k)|^2/d_i^2$$

We can do the same for the input, where we can choose a matrix

$$R = \lambda^2 \,\mathrm{diag}(r_1^2, r_2^2, \ldots, r_p^2)$$

The variables $r_i^2 > 0$ are a measure how much the costs should increase if the i-th input $u_i(k)$ increases by 1.

Example 32 ***: Tuning of GPC controller***

In this example we tune a GPC controller for the following system:

$$a_o(q)\, y(k) = b_o(q)\, u(k) + f_o(q)\, d_o(k) + c_o(q)\, e_o(k)$$

where

$$\begin{aligned} a_o(q) &= (1-0.9q^{-1}) \\ b_o(q) &= 0.03\, q^{-1} \\ c_o(q) &= (1-0.7q^{-1}) \\ f_o(q) &= 0 \end{aligned}$$

For the design we consider $e_o(k)$ *to be integrated ZMWN (so* $e_i(k) = \Delta e_o(k)$ *is ZMWN). First we transform the above polynomial IO model into a polynomial IIO model and we obtain the following system:*

$$a_i(q)\, y(k) = b_i(q)\, \Delta u(k) + f_i(q)\, d_i(k) + c_i(q)\, e_i(k)$$

where

$$\begin{aligned} a_i(q) &= (1-q^{-1})a_o(q) &&= (1-q^{-1})(1-0.9q^{-1}) \\ b_i(q) &= b_o(q) &&= 0.03\, q^{-1} \\ c_i(q) &= c_o(q) &&= (1-0.7q^{-1}) \\ f_i(q) &= f_o(q) &&= 0 \end{aligned}$$

Now we want to find the initial values of the tuning parameters following the tuning rules in section 7.1. To start with we plot the impulse response of the IO-model (!!!) in Figure 7.1. We see in the figure that system is well-damped. The 5% settling time is equal to $t_s = 30$. *The order of the IIO model is* $n = 2$ *and the dead time* $d = 0$ *.*
Now the following initial setting are recommended

$$\begin{aligned} N_m &= 1 + d = 1 \\ N_c &= n = 2 \\ N &= int(\alpha_N\, t_s) = int(\alpha_N\, 30) \end{aligned}$$

where $\alpha_N \in [1.5-2]$. *Changing* α_N *from 1.5 to 2 does not have much influence, and we therefore choose for the smallest value* $\alpha_N = 1.5$, *leading to* $N = 45$.
To verify the tuning we apply a step to the system ($r(k) = 1$ *for* $k \geq 0$*) and evaluate the output signal* $y(k)$. *The result is given in 7.2.*

Example 33 ***: Tuning of LQPC controller***

In this example we tune a LQPC controller for the following system:

$$A_o = \begin{bmatrix} 2.3000 & 1.0000 \\ -1.2000 & 0 \end{bmatrix} \quad C_o = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

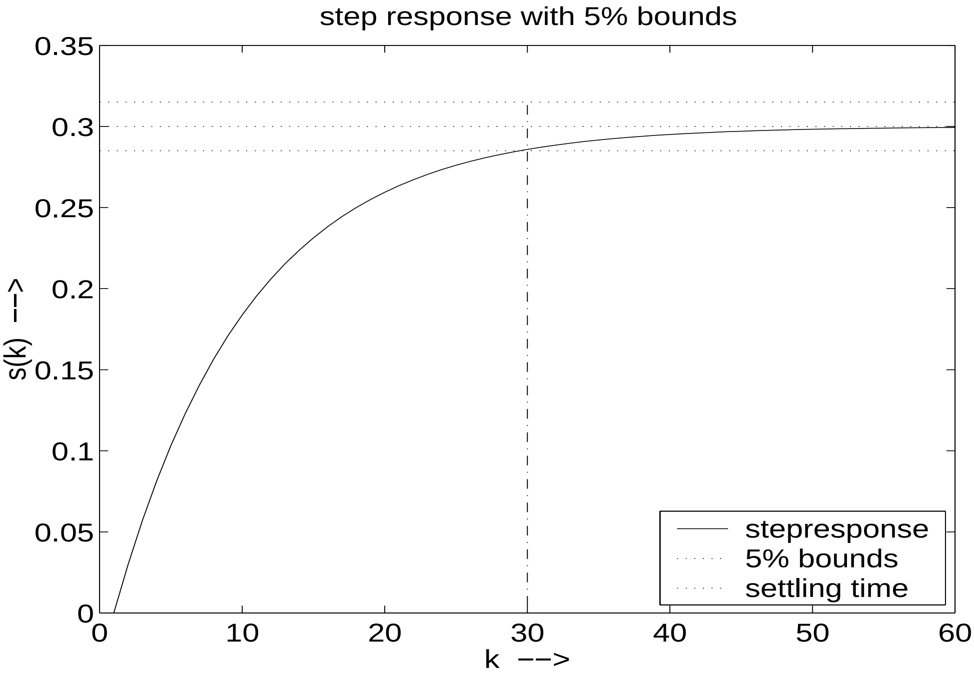

Figure 7.1: Impulse response of IO model

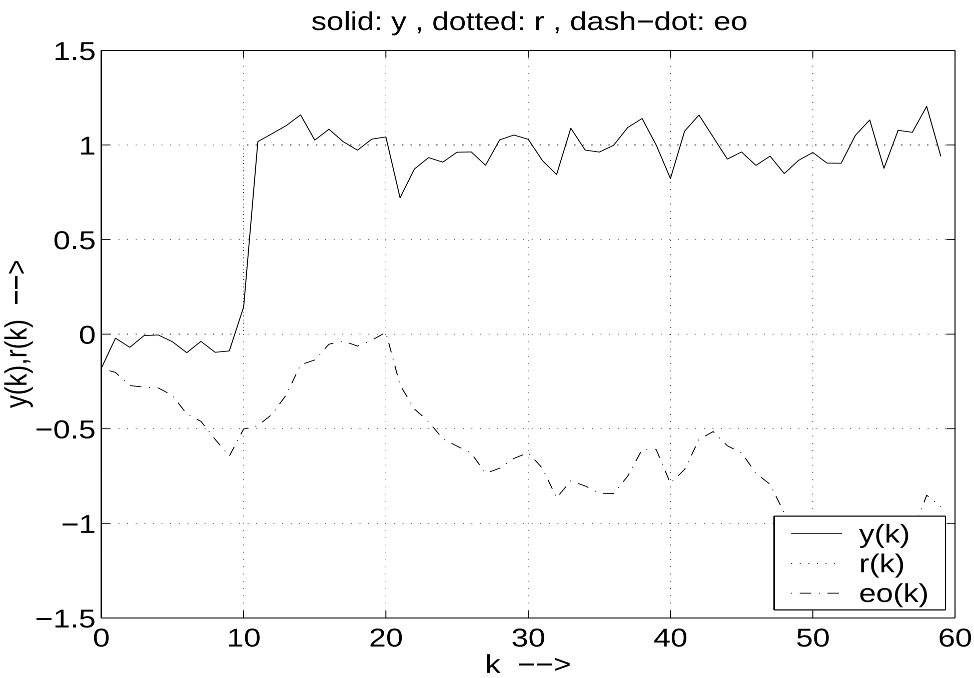

Figure 7.2: Closed loop response on a step reference signal

$$B_o = \begin{bmatrix} 0.1 \\ 0.09 \end{bmatrix} \quad K_o = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad L_o = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

which is found by sampling a continuous-time system with sampling-frequency $\omega_s = 100$ rad/s. Now we want to find the initial values of the tuning parameters following the tuning rules in section 7.1. To start with compute the eigenvalues of A_o and we find $\lambda_1 = 0.8$ and $\lambda_2 = 1.5$. We observe that the the system is unstable and so we have to determine the bandwidth of the original continuous-time system. we plot the bode-diagram of the continuous-time model in Figure 7.3. We see in the figure that $\omega_b = 2\pi 2.89 = 18.15$ rad/s.

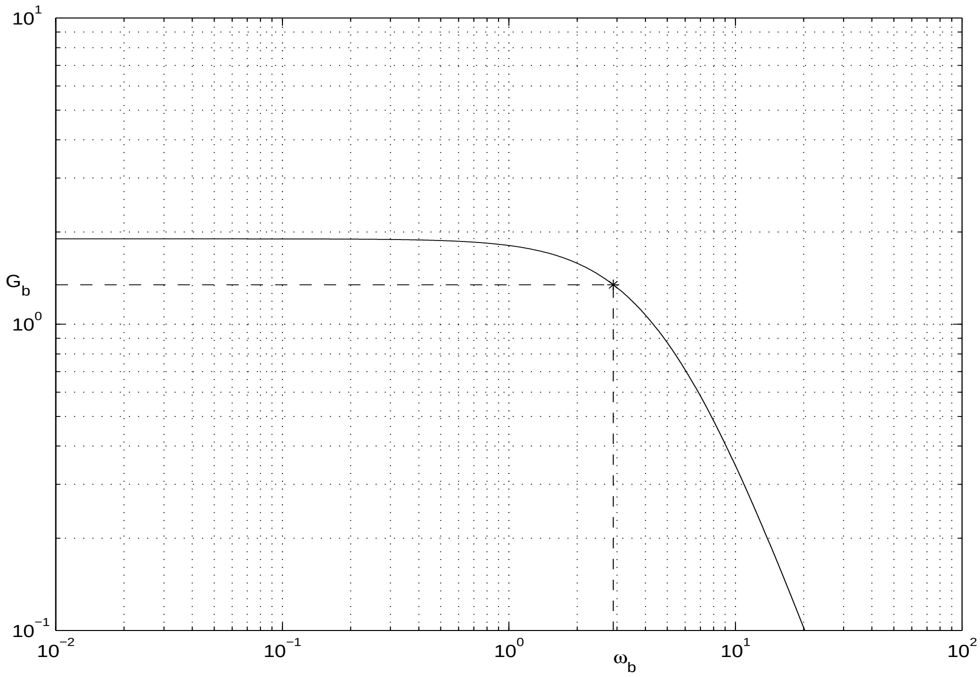

Figure 7.3: Bode plot of IO continuous-time model

The order of the IO model is $n = 2$ and the dead time $d = 0$ (there is no delay in the system).
Now the following initial setting are recommended

$$\begin{aligned} N_m &= 1 + d = 1 \\ N_c &= n = 2 \\ N &= int(\beta_N \, \omega_s/\omega_b) = int(\beta_N \, 5.51) \end{aligned}$$

where $\beta_N \in [4 - 25]$. Changing β_N from 4 to 25 does not have much influence, and we therefore choose for the smallest value $\beta_N = 4$, leading to $N = 22$.
To verify the influence of β_N, we start the system in initial state $x = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$, and evaluate the output signal $y(k)$. The result is given in 7.4. It is clear that for $\beta = 10$ the best response is found. We therefore obtain the optimal prediction horizon $N = int(\beta_N \cdot 5.51) = int(10 \cdot 5.51) = 55$.

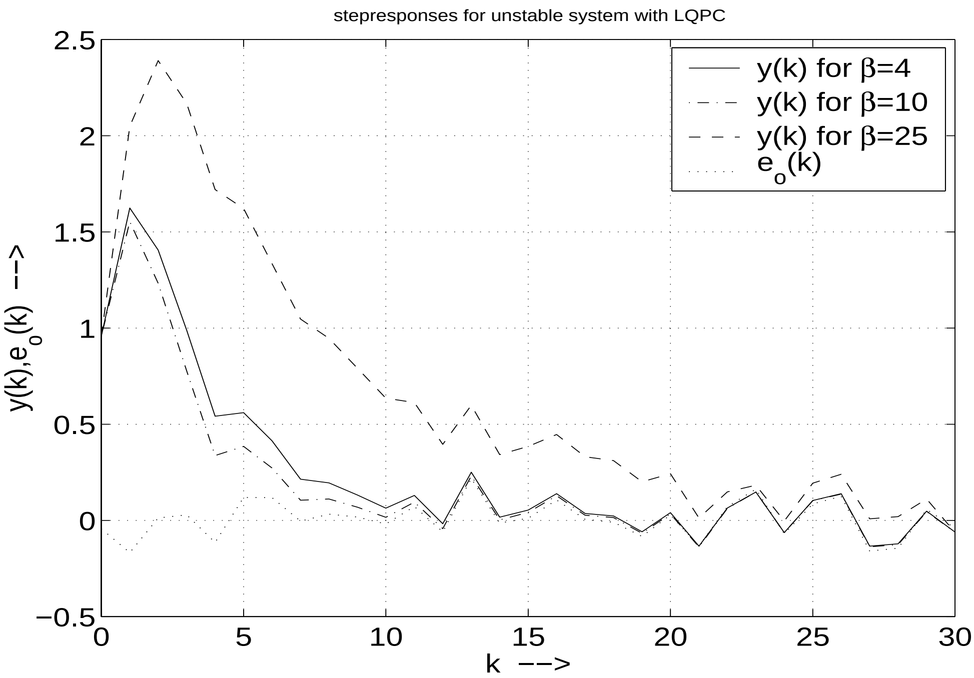

Figure 7.4: Closed loop response for initial state $x = [1\ 1]^T$.

7.2 Tuning as an optimization problem

In the ideal case the performance index and constraints reflect the mathematical formulation of the specifications from the original engineering problem. However, in practice the desired behaviour of the closed loop system is usually expressed in performance specifications, such as step-response criteria (overshoot, rise-time and settling-time) and response criteria on noise and disturbances. If the specifications are not too tight, the initial tuning, using the rules of thumb of the previous section, may result in a feasible controller. If the desired specifications are not met, an advanced tuning is necessary. In that case the relation between the performance specifications and the predictive control tuning parameters (N_m, N, N_c, for GPC: λ, $P(q)$, and for LQPC: Q, R) should be understood.
In this section we will study this relation and consider various techniques to improve or optimize our predictive controller.

We start with the closed loop system as derived in chapter 6:

$$\begin{bmatrix} v(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \begin{bmatrix} e(k) \\ \tilde{w}(k) \end{bmatrix}$$

where $\tilde{w}(k)$ consists of the present and future values of reference and disturbance signal. Let $\tilde{w}(k) = \tilde{w}_r(k)$ represent a step reference signal and let $\tilde{w}(k) = \tilde{w}_d(k)$ represent a step disturbance signal. Further define the output signal on a step reference signal as

$$s_{yr}(k) = M_{22}(q)\, \tilde{w}_r(k)$$

the output signal on a step disturbance signal as

$$s_{yd}(k) = M_{22}(q)\,\tilde{w}_d(k)$$

the input signal on a step reference signal as

$$s_{vr}(k) = M_{12}(q)\,\tilde{w}_r(k)$$

We will now consider the performance specifications in more detail, and look at responses criteria on step-reference signals (criteria 1 to 4), responses criteria on noise and disturbances (criteria 5 to 7) and finally at the stability of the closed loop (criterion 8):

1. **Overshoot of output signal on step reference signal:**
   The overshoot on the output is defined as the peak-value of the output signal $y(k)-1$ for $r(k)$ is a unit step, or equivalently, $\tilde{w}(k) = \tilde{w}_r(k)$. This value is equal to:

   $$\chi_{os} = \max_k(\ s_{yr}(k) - 1\ )$$

2. **Rise time of output signal on step reference signal:**
   The rise time is the time required for the output to rise to 80% of the final value. This value is equal to:

   $$\chi_{rt} = \min_k \left\{ k \,\middle|\, (\ s_{yr}(k) > 0.8\ ) \right\}$$

   The rise-time $k_o = \chi_{rt}$ is an integer and the derivative cannot be computed directly. Therefore we introduce the interpolated rise-time $\hat{\chi}_{rt}$:

   $$\hat{\chi}_{rt} = \frac{0.8 + (k_o - 1)s_{yr}(k_o) - k_o s_{yr}(k_o - 1)}{s_{yr}(k_o) - s_{yr}(k_o - 1)}$$

   The interpolated rise-time is the (real-valued) time-instant where the first-order interpolated step-response reach the value 0.8.

3. **Settling time of output signal on step reference signal:**
   The settling time is defined as the time the output signal $y(k)$ needs to settle within 5% of its final value.

   $$\chi_{st} = \min_{k_o} \left\{ k \geq k_o \,\middle|\, |\ s_{yr}(k) - 1\ | < 0.05 \right\}$$

   The settling-time $k_o = \chi_{st}$ is an integer and the derivative cannot be computed directly. Therefore we introduce the interpolated rise-time $\hat{\chi}_{st}$:

   $$\hat{\chi}_{st} = \frac{0.05 + (k_o - 1)|s_{yr}(k_o)| - k_o|s_{yr}(k_o - 1)|}{|s_{yr}(k_o)| - |s_{yr}(k_o - 1)|}$$

   The interpolated settling-time is the (real-valued) time-instant where the first-order interpolated tracking-error on a step reference signal is settled within its 5% region.

4. Peak value of input signal on step reference signal
of the output signal $v(k)$ for $r(k)$ is a step, is given by

$$\chi_{pvr} = \max_k |\, s_{vr}(k)\, |$$

5. Peak value of output signal on step disturbance signal:
The peak-value of the output signal $y(k)$ for $d(k)$ is a step, or equivalently $\tilde{w}(k) = \tilde{w}_d(k)$, is given by

$$\chi_{pyd} = \max_k |\, s_{yd}(k)\, |$$

6. RMS mistracking on zero-mean white noise signal:
The RMS value of the mistracking of the output due to a noise signal with spectrum $\delta(k)\, I$ is given by

$$\chi_{rm} = \|\, M_{21}(e^{j\omega})\, \|_2 = \left(\int_{\pi}^{\pi} |\, M_{21}(e^{j\omega})\, |\, d\omega \right)^{1/2}$$

7. Bandwidth of closed loop system:
The Bandwidth is the largest frequency below which the transfer function from noise to output signal is lower than -20 dB.

$$\chi_{bw} = \min_\omega \left\{ \omega \,\middle|\, |\, M_{21}(e^{j\omega})\, | < 0.1 \right\}$$

8. Stability radius of closed loop system:
The stability radius is maximum modulus value of the closed loop poles, which is equal to the spectral radius ρ of the closed loop system matrix A_T.

$$\chi_{sm} = \rho(A_t) = \max_i |\, \lambda_i(A_T)\, |$$

where λ_i are the eigenvalues of closed loop system matrix A_T.

(Of course many other performance specifications can be formulated.)

While tuning the predictive controller, some of the above performance specifications will be crucial in the design.
Define the tuning vectors η and θ as follows:

$$\eta = \begin{bmatrix} N_m \\ N \\ N_c \end{bmatrix} \qquad \text{for GPC: } \theta = \begin{bmatrix} \lambda \\ \mathrm{vec}(A_p) \\ \mathrm{vec}(B_p) \\ \mathrm{vec}(C_p) \\ \mathrm{vec}(D_p) \end{bmatrix} \qquad \text{for LQPC: } \theta = \begin{bmatrix} \mathrm{vec}(Q^{1/2}) \\ \mathrm{vec}(R^{1/2}) \end{bmatrix}$$

It is clear that the criteria χ depend on the chosen values of θ and η, so:

$$\chi = \chi(\theta, \eta)$$

where η consists of integer values and θ consists of real values. The tuning of η and θ can be done sequentially or simultaneously. In a sequential procedure we first choose the integer vector η and tune the parameters θ for these fixed η. In a simultaneous procedure the vectors η and θ are tuned at the same time, using mixed-integer search algorithms.

The tuning of the integer values η is not too difficult, if we can limit the size of the search space $\mathcal{S}_\eta$. A reasonable choice for bounds on N_m, N and N_c is as follows:

$$\begin{array}{ccccc} 1 & \leq & N_m & \leq & N \\ 1 & \leq & N & \leq & \alpha\, N_{2,o} \\ 1 & \leq & N_c & \leq & N \end{array}$$

where $N_{2,o}$ is the prediction horizon from the initial tuning, and $\alpha = 2$ if $N_{2,o}$ is big and $\alpha = 3$ or $\alpha = 4$ for small $N_{2,o}$. By varying $\alpha \in \mathbb{Z}$ we can increase or decrease the search space $\mathcal{S}_\eta$.
For the tuning of θ it is important to know the sensitivity of the criteria χ to changes in θ. A good measure for this sensitivity is the derivative $\dfrac{\partial \chi}{\partial \theta_i}$. The derivatives for the above mentioned performance specification 1 to 7 are given in appendix D. Note that these derivatives only gives the local sensitivity and are valid for small changes in θ. Furthermore, the criteria χ are not necessarily continuous functions of θ and so the derivatives may only give a local one-sided measure for the sensitivity.

Parameter optimization

Tuning the parameters θ and η can be formulated in two ways:

1. **Feasibility problem:** Find parameters θ and η such that some selected criteria $\chi_1, \ldots, \chi_p$ satisfy specific pre-described bounds:

$$\begin{array}{ccc} \chi_1 & \leq & c_1 \\ & \vdots & \\ \chi_p & \leq & c_p \end{array}$$

   For example: find parameters θ and η such that the overshoot is smaller than 1.05, the rise-time is smaller than 10 samples and the bandwidth of the closed loop system is larger than $\pi/2$ (this can be written as $-\chi_{bw} \leq -\pi/2$).

2. **Optimization problem:** Find parameters θ and η such that criterion χ_o is minimized, subject to some selected criteria $\chi_1, \ldots, \chi_p$ satisfying specific pre-described bounds:

$$\begin{array}{rcl} \chi_1 & \leq & c_1 \\ & \vdots & \\ \chi_p & \leq & c_p \end{array}$$

For example: find parameters θ and η such that the settling-time is minimized, subject to an overshoot smaller than 1.1 and actuator effort smaller than 10.

Since η has integer values, an integer feasibility/optimization problem arises. The criteria $\chi_i(x)$ are nonlinear functions of η and θ, and a general solution technique does not exist. Methods like Mixed Integer Linear Programming (MILP), dynamic programming, Constraint Logic Propagation (CLP), genetic algorithms, randomized algorithms and heuristic search algorithms can yield a solution. In general, these methods require large amounts of calculation time, because of the high complexity and they are mathematically classified as NP-hard problems. This means that the search space (the number of potential solutions) grows exponentially as a function of the problem size.

7.3 Some particular parameter settings

Predictive control is an open method, that means that many well-known controllers can be found by solving a specific predictive control problem with special parameter settings (Soeterboek [108]). In this section we will look at some of these special cases. We consider only the SISO case in polynomial setting, and

$$y(k) = \frac{q^{-d}b(q)}{a(q)\Delta(q)}\,\Delta u(k) + \frac{c(q)}{a(q)\Delta(q)}\,e(k)$$

where

$$\begin{array}{rcl} a(q) & = & 1 + a_1\,q^{-1} + \ldots + a_{n_a}\,q^{-n_a} \\ b(q) & = & b_1\,q^{-1} + \ldots + b_{n_b}\,q^{-n_b} \\ c(q) & = & 1 + c_1\,q^{-1} + \ldots + c_{n_c}\,q^{-n_c} \\ \Delta(q) & = & 1 - q^{-1} \end{array}$$

First we define the following parameters:

n_a = the degree of polynomial $a(q)$
n_b = the degree of polynomial $b(q)$
d = time-delay of the process.

Note that $d \geq 0$ is the first non-zero impulse response element $h(d+1) \neq 0$, or equivalently in a state space representation d is equal to the smallest number for which $C_1 A^d B_3 \neq 0$.

Minimum variance control:

In a minimum variance control (MVC) setting, the following performance index is minimized:

$$J(u,k) = |\hat{y}(k+d|k) - r(k+d)|^2$$

This means that if we use the GPC performance with setting $N_m = N = d$, $N_c = 1$, $P(q) = 1$ and $\lambda = 0$ we obtain the minimum variance controller. An important remark is that a MV controller only gives a stable closed loop for minimum phase plants.

Generalized minimum variance control:

In a generalized minimum variance control (GMVC) setting, the performance index is extended with a weighting on the input:

$$J(u,k) = |\hat{y}(k+d|k) - r(k+d)|^2 + \lambda^2 |\Delta u(k)|^2$$

where $y(k)$ is the output signal of an IIO-model. This means that if we use the GPC performance, with setting $N_m = N = d$, $N_c = d$, $P(q) = 1$ and some $\lambda > 0$ we obtain the generalized minimum variance controller. A generalized minimum variance controller also may give a stable closed loop for nonminimum phase plants, if λ is well tuned.

Dead-beat control:

In a dead-beat control setting, the following performance index is minimized:

$$J(u,k) = \sum_{j=n_b+d}^{n_a+n_b+d} |\hat{y}(k+j|k) - r(k+j)|^2$$

$$\Delta u(k+j) = 0 \;\text{ for }\; j \geq n_a + 1$$

This means that if we use the GPC performance, with setting $N_m = n_b + d$, $N_c = n_a + 1$, $P(q) = 1$ and $N = n_a + n_b + d$ and $\lambda = 0$ we obtain the dead-beat controller.

Mean-level control:

In a mean-level control (MLC) setting, the following performance index is minimized:

$$J(u,k) = \sum_{j=1}^{\infty} |\hat{y}(k+j|k) - r(k+j)|^2$$

$$\Delta u(k+j) = 0 \;\text{ for }\; j \geq 1$$

This means that if we use the GPC performance with setting $N_m = 1$, $N_c = 1$, $P(q) = 1$ and $N \longrightarrow \infty$. we obtain the mean-level controller.

Pole-placement control:

In a pole-placement control setting, the following performance index is minimized:

$$J(u,k) = \sum_{j=n_b+d}^{n_a+n_b+d} |\hat{y}_p(k+j|k) - r(k+j)|^2$$

$$\Delta u(k+j) = 0 \;\text{ for }\; j \geq n_a + 1$$

where $y_p(k) = P(q)y(k)$ is the weighted output signal. This means that if we use the GPC performance with setting $N_m = n_b + d$, $N_c = n_a + 1$, $P(q) = P_d(q)$ and $N = n_a + n_b + d$. In state space representation, the eigenvalues of $(A - B_3F)$ will become equal to the roots of polynomial $P_d(q)$.

Chapter 8

MPC using a feedback law, based on linear matrix inequalities

In chapter 2 to chapter 5 the framework of finite horizon predictive control was elaborated. In this chapter a different approach to predictive control is presented, using linear matrix inequalities (LMIs). The control strategy is focussed on computing an optimal feedback law in the receding horizon framework. A controller can be derived by solving convex optimization problems using fast and reliable tecniques. LMI problems can be solved in polynomial time, which means that they have low computational complexity.
In fact, the field of application of Linear Matrix Inequalities (LMIs) in system and control engineering is enormous, and more and more engineering problems are solved using LMIs. Some examples of application are stability theory, model and controller reduction, robust control, system identification and (last but not least) predictive control.

The main reasons for using LMIs in predictive control are the following:

- Stability is easily guaranteed.
- Feasibility results are easy to obtain.
- Extension to systems with model uncertainty can be made (see chapter 9).
- Convex properties are preserved.

Of course there are also some drawbacks:

- Although solving LMIs can be done using convex techniques, the optimization problem is more complex than a quadratic programming problem, as discussed in chapter 5.
- Feasibility in the noisy case still can not be guaranteed.

In the first section of this chapter we will discuss the main features of linear matrix inequalities, in section 2 we will show how LMIs can be used to solve the MPC problem, and in section 3 we will look at the inequality constrained case. Robustness issues, concerning stability in the case of model uncertainty, are discussed in chapter 9.

8.1 Linear matrix inequalities

Consider the linear matrix expression

$$F(\theta) = F_0 + \sum_{i=1}^{m} \theta_i F_i \tag{8.1}$$

where $\theta \in \mathrm{IR}^{m\times 1}$, with elements θ_i, is the variable and the symmetric matrices $F_i = F_i^T \in \mathrm{IR}^{n\times n}$, $i = 0, \ldots, m$ are given. A *strict* Linear Matrix Inequality (LMI) has the form

$$F(\theta) > 0 \tag{8.2}$$

The inequality symbol in (8.2) means that $F(\theta)$ is positive definite (i.e. $x^T F(\theta) x > 0$ for all nonzero $x \in \mathrm{IR}^{n\times 1}$). A *nonstrict* LMI has the form

$$F(\theta) \geq 0$$

Properties of LMIs:

- Convexity:
  The set $C = \{\theta | F(\theta) > 0\}$ is convex in θ, which means that for each pair $\theta_1, \theta_2 \in C$ and for all $\lambda \in [0,1]$ the next property holds

$$\lambda\theta_1 + (1-\lambda)\theta_2 \in C$$

- Multiple LMIs:
  Multiple LMIs can be expressed as a single LMI, since

$$F^{(1)} > 0 \ , \ F^{(2)} > 0 \ , \ldots , \ F^{(p)} > 0$$

  is equaivalent to:

$$\begin{bmatrix} F^{(1)} & 0 & \cdots & 0 \\ 0 & F^{(2)} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & F^{(p)} \end{bmatrix} > 0$$

- Schur complements:
  Consider the following LMI:

$$\begin{bmatrix} Q(\theta) & S(\theta) \\ S^T(\theta) & R(\theta) \end{bmatrix} > 0$$

where $Q(\theta) = Q^T(\theta)$, $R(\theta) = R^T(\theta)$ and $S(\theta)$ depend affinely on θ. This is equaivalent to

$$R(\theta) > 0 \ , \ Q(\theta) - S(\theta)R^{-1}(\theta)S^T(\theta) > 0$$

Finally note that any symmetric matrix $M \in \mathbb{R}^{n\times n}$ can be written as

$$M = \sum_{i=1}^{n(n+1)/2} M_i \, m_i$$

where m_i are the scalar entries of the upper-right triangle part of M, and M_i are symmetric matrices with entries 0 or 1. For example:

$$M = \begin{bmatrix} m_1 & m_2 \\ m_2 & m_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} m_1 + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} m_2 + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} m_3$$

This means that a matrix inequality, that is linear in a symmetric matrix M, can easily be transformed into a linear matrix inequality with a parameter vector

$$\theta = \begin{bmatrix} m_1 & m_2 & \ldots & m_{n(n+1)/2} \end{bmatrix}^T$$

Before we look at MPC using LMIs, we consider the more simple cases of Lyapunov stability of an autonomous system and the properties of a stabilizing state-feedback.

Lyapunov theory:

Probably the most elementary LMI is related to the Lyapunov theory. The difference state equation

$$x(k+1) = A\,x(k)$$

is stable (i.e. all state trajectories $x(k)$ go to zero) if and only if there exists a positive-definite matrix P such that

$$A^T PA - P < 0$$

For this system the Lyapunov function $V(k) = x^T(k)Px(k)$ is positive definite for $P > 0$, and the increment $\Delta V(k) = V(k+1) - V(k)$ is negative definite because

$$\begin{aligned} V(k+1) - V(k) &= x^T(k+1)Px(k+1) - x^T(k)Px(k) \\ &= (Ax(k))^T P(Ax(k)) - x^T(k)Px(k) \\ &= x^T(k)(A^T PA - P)x(k) \\ &< 0 \end{aligned}$$

for $(A^T PA - P) < 0$.

Stabilizing state-feedback:

Consider the difference state equation:

$$x(k+1) = A\,x(k) + B\,v(k)$$

where $x(k)$ is the state and $v(k)$ is the input. When we apply a state feedback $v(k) = F\,x(k)$, we obtain:

$$x(k+1) = A\,x(k) + B\,Fx(k) = (A + BF)\,x(k)$$

and so the corresponding Lyapunov equation becomes:

$$(A^T + F^T B^T)P(A + BF) - P < 0 \quad , \quad P > 0$$

Or by choosing $P = S^{-1}$ this is equivalent to

$$(A^T + F^T B^T)S^{-1}(A + BF) - S^{-1} < 0 \quad , \quad S > 0$$

pre- and post-multiplying with S result in

$$S(A^T + F^T B^T)S^{-1}(A + BF)S - S < 0 \quad , \quad S > 0$$

By choosing $F = YS^{-1}$ we obtain the condition:

$$(SA^T + Y^T B^T)S^{-1}(AS + BY) - S < 0 \quad , \quad S > 0$$

which can be rewritten (using the Schur complement transformation) as

$$\begin{bmatrix} S & SA^T + Y^T B^T \\ AS + BY & S \end{bmatrix} > 0 \ \ , S > 0 \tag{8.3}$$

which are LMIs in S and Y. So any S and Y satisfying the above LMI result in a stabilizing state-feedback $F = YS^{-1}$.

8.2 Unconstrained MPC using linear matrix inequalities

Consider the system

$$x(k+1) = A\,x(k) + B_2\,w(k) + B_3\,v(k) \tag{8.4}$$
$$y(k) = C_1\,x(k) + D_{12}\,w(k) \tag{8.5}$$
$$z(k) = C_2\,x(k) + D_{22}\,w(k) + D_{23}\,v(k) \tag{8.6}$$

For simplicity reasons, in this chapter, we consider $e(k) = 0$. Like in chapter 5, we consider constant external signal w, a zero steady-state performance signal z_{ss} and a weighting matrix $\Gamma(j)$ which is equal to identity, so

$$\begin{aligned} w(k+j) &= w_{ss} \quad \text{for all} \quad j \geq 0 \\ z_{ss} &= 0 \\ \Gamma(j) &= I \quad \text{for all} \quad j \geq 0 \end{aligned}$$

Let the system have a steady-state, given by $(v_{ss}, x_{ss}, w_{ss}, z_{ss}) = (D_{ssv}w_{ss}, D_{ssx}w_{ss}, w_{ss}, 0)$. We define the shifted versions of the input and state which are zero in steady-state:

$$\begin{aligned} v_\Delta(k) &= v(k) - v_{ss} \\ x_\Delta(k) &= x(k) - x_{ss} \end{aligned}$$

Now we consider the problem of finding an optimal constant state feedback $v_\Delta(k) = F\,x_\Delta(k)$ to minimize the quadratic objective

$$\min_F J(k)$$

where $J(k)$ is given by ($\Gamma > 0$):

$$J(k) = \sum_{j=0}^{\infty} \hat{z}^T(k+j|k)\Gamma\hat{z}(k+j|k)$$

in which $\hat{z}(k+j|k)$ is the prediction of $z(k+j)$, based on knowledge up to time k.

$$\begin{aligned} J(k) &= \sum_{j=0}^{\infty} \hat{z}(k+j|k)^T\Gamma\hat{z}(k+j|k) \\ &= \sum_{j=0}^{\infty}(C_2\hat{x}(k+j|k) + D_{22}w(k+j) + D_{23}v(k+j))^T\Gamma(C_2\hat{x}(k+j|k) \\ &\quad +D_{22}w(k+j) + D_{23}v(k+j)) \\ &= \sum_{j=0}^{\infty}(C_2x_\Delta(k+j|k) + D_{22}v_\Delta(k+j))^T\Gamma(C_2x_\Delta(k+j|k) + D_{22}v_\Delta(k+j)) \\ &= \sum_{j=0}^{\infty} {x_\Delta}^T(k+j|k)(C_2^T + F^TD_{22}^T)\Gamma(C_2 + D_{22}F)x_\Delta(k+j|k) \end{aligned}$$

This problem can be translated to a minimization problem with LMI-constraints. Consider the quadratic function $V(k) = x_\Delta(k)^TPx_\Delta(k)$, $P > 0$, that satisfies

$$\Delta V(k) = V(k+1) - V(k) < -{x_\Delta}^T(k)(C^T + F^TD^T)\Gamma(C + DF)x_\Delta(k) \tag{8.7}$$

for every trajectory. Note that because $\Delta V(k) < 0$ and $V(k) > 0$, it follows that $\Delta V(\infty) = 0$. This implies that $x_\Delta(\infty) = 0$ and therefore $V(\infty) = 0$. Now there holds:

$$\begin{aligned} V(k) &= V(k) - V(\infty) \\ &= -\sum_{j=0}^{\infty} \Delta V(k+j) \\ &> \sum_{j=0}^{\infty} {x_\Delta}^T(k+j)(C^T + F^T D^T)\Gamma(C+DF)x_\Delta(k+j) \\ &= J(k) \end{aligned}$$

so ${x_\Delta}^T(k)Px_\Delta(k)$ is a lyapunov function and at the same time an upper bound on $J(k)$. Now derive

$$\begin{aligned} \Delta V(k) &= V(k+1) - V(k) = \\ &= {x_\Delta}^T(k+1)Px_\Delta(k+1) - {x_\Delta}^T(k)Px_\Delta(k) = \\ &= {x_\Delta}^T(k)\Big((A+BF)^T P(A+BF)\Big)x_\Delta(k) - {x_\Delta}^T(k)Px_\Delta(k) = \\ &= {x_\Delta}^T(k)\Big((A+BF)^T P(A+BF) - P\Big)x_\Delta(k) \end{aligned}$$

And so condition (8.7) is the same as

$$(A+BF)^T P(A+BF) - P + (C+DF)^T\Gamma(C+DF) < 0$$

By setting $P = \gamma S^{-1}$ and $F = YS^{-1}$ we obtain the following condition:

$$\gamma(A^T + S^{-1}Y^T B^T)S^{-1}(A+BYS^{-1}) - \gamma S^{-1} + (C^T + S^{-1}Y^T D^T)\Gamma(C+DYS^{-1}) < 0$$

pre- and post-multiplying with S and division by γ result in

$$(SA^T + Y^T B^T)S^{-1}(AS+BY) - S + \gamma^{-1}(SC^T + Y^T D^T)\Gamma(CS+DY) < 0 \qquad (8.8)$$

or

$$S \;-\; \begin{bmatrix} SA^T + Y^T B^T & (SC^T + Y^T D^T)\Gamma^{1/2} \end{bmatrix} \begin{bmatrix} S^{-1} & 0 \\ 0 & \gamma^{-1} I \end{bmatrix} \begin{bmatrix} AS+BY \\ \Gamma^{1/2}(CS+DY) \end{bmatrix} > 0$$

which can be rewritten (using the Schur complement transformation) as

$$\begin{bmatrix} S & SA^T + Y^T B^T & (SC^T + Y^T D^T)\Gamma^{1/2} \\ AS+BY & S & 0 \\ \Gamma^{1/2}(CS+DY) & 0 & \gamma I \end{bmatrix} > 0 \;\;, S > 0 \qquad (8.9)$$

So any γ, S and Y satisfying the above LMI gives an upper bound

$$V(k) = \gamma\, x_\Delta(k)^T S^{-1} x_\Delta(k) > J(k)$$

Finally introduce an additional constraint that

$${x_\Delta}^T(k)S^{-1}x_\Delta(k) \leq 1 \qquad (8.10)$$

which, using the Schur complement property, is equivalent to

$$\begin{bmatrix} 1 & {x_\Delta}^T(k) \\ x_\Delta(k) & S \end{bmatrix} \geq 0 \tag{8.11}$$

Now

$$\gamma > J$$

and so a minimization of γ subject to LMI constraints (8.9) and (8.11) is equivalent to a miminization of the upper bound of the performance index.
Note that the LMI constraint (8.9) looks more complicated than expression (8.8). However (8.9) is linear in S, Y and γ. A convex optimization algorithm, denoted as the interior point algorithm, can solve convex optimization problems with many LMI constraints (up to a 1000 constraints) in a fast and robust way.

8.3 Constrained MPC using linear matrix inequalities

In the previous section an unconstrained predictive controller in the LMI-setting was derived. In this section we incorporate constraints. The most important tool we use is the notion of invariant ellipsoid:

Invariant ellipsoid:

Lemma: *Consider the system (8.4) and (8.6). At sampling time k, consider*

$$x_\Delta(k|k)^T S^{-1} x_\Delta(k|k) \leq 1$$

and let S also satisfy (8.9), then there will hold for all $j > 0$:

$$x_\Delta(k+j|k)^T S^{-1} x_\Delta(k+j|k) \leq 1$$

Proof:
From the previous section we know that when (8.9) is satisfied, there holds:

$$V(k+j) < V(k) \quad \text{for} \;\; j > 0$$

Using $V(k) = {x_\Delta}^T(k) P x_\Delta(k) = \gamma {x_\Delta}^T(k) S^{-1} x_\Delta(k)$ we derive:

$$x_\Delta(k+j|k)^T S^{-1} x_\Delta(k+j|k) = \gamma^{-1} V(k+j) < \gamma^{-1} V(k) = x_\Delta(k|k)^T S^{-1} x_\Delta(k|k) \leq 1$$

□ End Proof

$\mathcal{E} = \Big\{ \eta \,|\, \eta^T S^{-1} \eta \leq 1 \Big\} = \Big\{ \eta \,|\, \eta^T P \eta \leq \gamma \Big\}$ is an invariant ellipsoid for the predicted states of the system, and so

$$x_\Delta(k|k) \in \mathcal{E} \;\; \Rightarrow \;\; x_\Delta(k+j|k) \in \mathcal{E} \quad \forall\, j > 0$$

The condition $x_\Delta(k|k)^T S^{-1} x_\Delta(k|k) \leq 1$ of the above lemma is equivalent to the LMI

$$\begin{bmatrix} 1 & {x_\Delta}^T(k|k) \\ x_\Delta(k|k) & S \end{bmatrix} \geq 0 \tag{8.12}$$

Signal constraints

Lemma 34 *Consider the scalar signal*

$$\psi(k) = C_4 x(k) + D_{42} w(k) + D_{43} v(k) \tag{8.13}$$

and the signal constraint

$$|\psi(k)| \leq \psi_{max} \tag{8.14}$$

where $D_{43} \neq 0$ *and* $\psi_{max} > 0$. *Further let condition (8.11) hold and assume*

$$\psi_{ss}(k) = C_4 x_{ss}(k) + D_{42} w_{ss}(k) + D_{43} v_{ss}(k) < \psi_{max}$$

Then condition (8.14) will be satisfied if

$$\begin{bmatrix} S & (C_4 S + D_{43} Y)^T \\ C_4 S + D_{43} Y & (\psi_{max} - \psi_{ss})^2 \end{bmatrix} \geq 0 \tag{8.15}$$

Remark:
Note that for an input constraint on the i-th input v_i

$$|v_{\Delta i}(k+j)| \leq v_{\Delta max,i} \quad \text{for} \quad j \geq 0$$

is equivalent to condition (8.14) for $\psi_{max,i} = v_{\Delta max,i}$, $C_2 = 0$, $D_{42} = 0$ and $D_{43} = e_i$, where e_i is a selection vector such that $v_i(k) = e_i\, v(k)$.
An output constraint on the i-th output y_i

$$|y_i(k{+}j{+}1) - y_{ss,i}| \leq y_{max,i} \quad \text{for} \quad j \geq 0$$

where $y_{ss} = C_1 x_{ss} + D_{12} w_{ss} = (C_1 D_{ssx} + D_{12}) w_{ss}$.
By deriving

$$\begin{aligned} y_i(k{+}j{+}1) &= C_1 x(k{+}j{+}1) + D_{12} w(k{+}j{+}1) = \\ &= C_1 x_\Delta(k+j+1) + y_{ss} = \\ &= C_1 A x_\Delta(k+j) + C_1 B_3 v_\Delta(k+j) + y_{ss} \end{aligned}$$

we find that the output condition is equaivalent to condition (8.14) for $\psi_{max,i} = y_{max,i}$, $C_4 = C_1 A$, $D_{42} = 0$ and $D_{43} = C_{y,i} B$.

8.4 Summarizing MPC using linear matrix inequalities

Consider the system

$$
\begin{aligned}
x(k+1) &= A\,x(k) + B\,v(k) && (8.16)\\
y(k) &= C_1\,x(k) + D_{12}\,w(k) && (8.17)\\
z(k) &= C_2\,x(k) + D_{22}\,w(k) + D_{23}\,v(k) && (8.18)\\
\psi(k) &= C_4\,x(k) + D_{42}\,w(k) + D_{43}\,v(k) && (8.19)
\end{aligned}
$$

with a steady-state $(v_{ss}, x_{ss}, w_{ss}, z_{ss}) = (D_{ssv}w_{ss}, D_{ssx}w_{ss}, w_{ss}, 0)$ The LMI-MPC control problem is to find, at each time-instant k, a state-feedback law

$$v(k) = F\,(x(k) - x_{ss}) + v_{ss}$$

such that the criterion

$$\min_F J(k) \quad \text{where} \quad J(k) = \sum_{j=0}^{\infty} z^T(k+j)\Gamma z(k+j) \tag{8.20}$$

is optimized subject to the constraint

$$|\,\psi_\ell(k+j)\,| \le \psi_{\ell,max}\ , \quad , \text{ for } \ell = 1, \ldots, n_\psi\ , \quad \forall j \ge 0 \tag{8.21}$$

where $\psi_{\ell,max} > 0$ for $\ell = 1, \ldots, n_\psi$. The optimal solution is found by solving the following LMI problem:

Theorem 35 *Given a system with state space description (8.16)-(8.18) and a control problem of minimizing (8.20) subject to constraint (8.21), and a external signal*

$$w(k+j) = w_{ss} \quad \textit{for } \ j > 0$$

The state-feedback $F = Y\,S^{-1}$ minimizing the worst-case $J(k)$ can be found by solving:

$$\min_{\gamma,S,Y} \gamma \tag{8.22}$$

subject to

(LMI for lyapunov stability:)

$$S > 0 \tag{8.23}$$

(LMI for ellipsoid boundary:)

$$\begin{bmatrix} 1 & {x_\Delta}^T(k|k) \\ x_\Delta(k|k) & S \end{bmatrix} \ge 0 \tag{8.24}$$

(LMIs for worst case MPC performance index:)

$$\begin{bmatrix} S & SA^T + Y^T B^T & (SC^T + Y^T D^T)\Gamma^{1/2} \\ AS + BY & S & 0 \\ \Gamma^{1/2}(CS + DY) & 0 & \gamma I \end{bmatrix} > 0 \tag{8.25}$$

(LMIs for constraints on ψ:)

$$\begin{bmatrix} S & (C_4 S + D_{43} Y)^T E_\ell^T \\ E_\ell (C_4 S + D_{43} Y) & (\psi_{\ell,max} - \psi_{\ell,ss})^2 \end{bmatrix} \geq 0, \qquad \ell = 1, \ldots, n_\psi \tag{8.26}$$

where $E_\ell = \begin{bmatrix} 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \end{bmatrix}$ *with the* 1 *on the* ℓ*-th position.*

LMI (8.23) guarantees Lyapunov stability, LMI (8.24) describes an invariant ellipsoid for the state, (8.25) gives the LMI for the MPC-criterion and (8.26) correspond to the state constraint on ψ_ℓ .

Chapter 9

Robustness

9.1 Introduction

In the last decades the focus of control theory is shifting more and more towards control of uncertain systems. A disadvantage of common finite horizon predictive control techniques (as presented in the chapters 2 to 5) is their inability to deal explicitly with uncertainty in the process model. In real life engineering problems, system parameters are uncertain, because they are difficult to estimate or because they vary in time. That means that we don't know the exact values of the parameters, but only a set in which the system moves.

A controller is called robustly stable if a small change in the system parameters does not destabilize the system. It is said to give robust performance if the performance does not deteriorate significantly for small changes parameters in the system. The design and analysis of linear time-invariant (LTI) robust controllers for linear systems has been studied extensively in the last decade.

In practice a predictive controller usually can be tuned quite easily to give a stable closed-loop and to be robust with respect to model mismatch. However, in order to be able to guarantee stability, feasibility and/or robustness and to obtain better and easier tuning rules by increased insight and better algorithms, the development of a general stability and robustness theory for predictive control has become an important research topic.

One way to obtain robustness in predictive control is by careful tuning (Clarke & Mothadi [20], Soeterboek [108], Lee & Yu [70]). This method gives quite satisfactory results in the unconstrained case.

In the constrained case robustness analysis is much more difficult, resulting in more complex and/or conservative tuning rules (Zafiriou [136], Gencelli & Nikolaou [52]). One approach to guarantee robust stability is the use of an explicit contraction constraint (Zheng & Morari [141], De Vries & van den Boom [126]). The two main disadvantages of this approach are the resulting non-linear optimization problem and the large but unclear influence of the choice of the contraction constraint on the controlled system.

A second solution is using a Youla-parametrization in combination with the small-gain theorem. By optimizing the time-varying Youla parameter instead of the output of the controller, stability constraints on the Youla parameter as well as signal constraints can be satisfied (Van den Boom & De Vries [125]).

An approach to guarantee robust performance is to guarantee that the criterion function is a contraction by optimizing the maximum of the criterion function over all possible models (Zheng & Morari [141]). The main disadvantages of this method are the need to use polytopic model uncertainty descriptions, the use of less general criterion functions and, especially, the difficult *min max* optimization. Recently Kothare *et al.* ([63]) derived an LMI-based method, that circumvented all of these disadvantages, however, it may become quite conservative. This method, based on the techniques presented in chapter 8, will be discussed in section 9.4.

Concepts of Robustness

We now formalize the above given definitions of robust stability and robust performance. First the small change of the system parameters has to be described. The 'real' process G_t is assumed to live in an uncertainty set $\mathcal{G}$. In the next section we will give specific structures to describe this uncertainty set.

Robust Stability:
A controller is robustly stable, if it stabilizes all systems in the set $\mathcal{G}$.

Robust Performance:
A controller gives robust performance, if it can maintain a predescribed performance-level for all systems in the set $\mathcal{G}$.

Of course the definition of robust performance needs to be more specific. We have to define a measure of performance. In the MPC case we obviously choose the standard predictive control performance index.

9.2 Model uncertainty

The basic technique to treat model uncertainty in control is by designing a controller, based on a nominal model G, given by the equations (4.26-4.28), and keep in mind that in the description of the 'real' process G_t belongs to an uncertainty set $\mathcal{G}$. Such a set $\mathcal{G}$ can be structured or unstructured.

Unstructured model uncertainty: The difference between G and G_t is described by a tranfer matrix Ω, which is assumed to be bounded in some system norm. The unstructured uncertainty model is given by the following extended description:

$$x(k+1) = A\,x(k) + B_1\,e(k) + B_2\,w(k) + B_3\,v(k) + B_4\,\delta(k) \tag{9.1}$$

$$y(k) = C_1\,x(k) + D_{11}\,e(k) + D_{12}\,w(k) \tag{9.2}$$

$$
\begin{aligned}
z(k) &= C_2\,x(k) + D_{21}\,e(k) + D_{22}\,w(k) + D_{23}\,v(k) + D_{24}\,\delta(k) && (9.3)\\
\psi(k) &= C_4\,x(k) + D_{41}\,e(k) + D_{42}\,w(k) + D_{43}\,v(k) + D_{44}\,\delta(k) && (9.4)\\
\epsilon(k) &= C_5\,x(k) + D_{53}\,v(k) && (9.5)\\
\delta(k) &= \Omega(q)\epsilon(k) && (9.6)
\end{aligned}
$$

In this extended description an uncertainty block $\Omega(q)$ is added, representing the model uncertainty. $\Omega(q)$ itself is unknown, but is assumed to belong to the $\mathcal{O}$ consisting of all stable ∞-norm bounded LTI tranfer matrices

$$\mathcal{O} = \{\, \Omega \mid \|\Omega\|_\infty \leq 1 \,\}$$

Here the ∞-norm is given by:

$$\|\Omega\|_\infty = \max_{|z|\geq 1} \bar{\sigma}\Big(\Omega(z)\Big)$$

where $\bar{\sigma}$ stands for the maximum singular value. If Ω is a scalar transfer function we obtain:

$$\|\Omega\|_\infty = \max_{|z|\geq 1} |\Omega(z)|$$

which is equal to the gain peak-value in the Bode diagram of Ω. The unstructured model uncertainty description is important, because simple tools are available to analyse robustness in the case of such an unstructured uncertainty.

Structured model uncertainty: A finite number of parameters is not exactly known, causing the state space system matrices in the state space description to belong to an uncertainty set. We adopt the following uncertainty model:

$$
\begin{aligned}
x(k+1) &= A\,x(k) + B_1\,e(k) + B_2\,w(k) + B_3\,v(k) && (9.7)\\
y(k) &= C_1\,x(k) + D_{11}\,e(k) + D_{12}\,w(k) && (9.8)\\
z(k) &= C_2\,x(k) + D_{21}\,e(k) + D_{22}\,w(k) + D_{23}\,v(k) && (9.9)\\
\psi(k) &= C_4\,x(k) + D_{41}\,e(k) + D_{42}\,w(k) + D_{43}\,v(k) && (9.10)\\
& && (9.11)
\end{aligned}
$$

Without loss of generality we will assume C_1, D_{11} and D_{12} to be fixed. Note that C_1 can always be chosen consisting only of ones and zeros (see for example C_o and C_i in chapter 2). We assume the remaining system matrices

$$P = \begin{bmatrix} A & B_1 & B_2 & B_3 \\ C_2 & D_{21} & D_{22} & D_{23} \\ C_4 & D_{41} & D_{42} & D_{43} \end{bmatrix}$$

to belong to a set $\mathcal{P}$, given by

$$P \in \text{Co}\{P_1, P_2, \ldots, P_L\}$$

With this notation we mean that P is in the convex closure of P_1, P_2, ..., P_L and there exist variables $0 \leq \lambda_i \leq 1$ such that

$$P = \lambda_1 P_1 + \lambda_2 P_2 + \ldots + \lambda_L P_L \qquad \text{where} \quad \sum_{i=1}^{L} \lambda_i = 1$$

The set $\text{Co}\{P_1, P_2, \ldots, P_L\}$ is often denoted as a polytope with vertices in P_1, P_2, ..., P_L. For the above model the vertices P_i, $i = 1, \ldots, L$ can be obtained by modelling of the plant, and can be seen as extreme values of P.

9.3 Robustness by tuning

In this section we will consider a tuning procedure to obtain robust stability in the case of unstructured model uncertainty (equations 9.1-9.6) and model error bound $\|\Omega\|_\infty \leq 1$. We only consider the case without inequality constraints, and so the contoller is linear time invariant and has the form (5.143-5.144):

$$\begin{aligned}
x_c(k+1) &= (A - B_3 F - B_1 D_{11}^{-1} C_1 - B_3 D_e D_{11}^{-1} C_1)\, x_c(k) \\
&\quad +(B_1 D_{11}^{-1} + B_3 D_e D_{11}^{-1})\, y(k) \\
&\quad +(B_3 D_w - B_3 D_e\, D_{11}^{-1} D_{12} E_w + B_2 E_w - B_1 D_{11}^{-1} D_{12} E_w)\, \tilde{w}(k) \\
v(k) &= (-F - D_e\, D_{11}^{-1}\, C_1)\, x_c(k) + D_e\, D_{11}^{-1} y(k) + \\
&\quad +(D_w - D_e\, D_{11}^{-1} D_{12} E_w)\, \tilde{w}(k)
\end{aligned}$$

Because of the superposition principle, we can consider $\tilde{w} = 0$ and $e = 0$ without changing the closed loop stability properties. Now we can close the loop, substituting the controller equations in (9.1),(9.2),(9.5), leading to:

$$\begin{aligned}
x(k+1) &= (A + B_3 D_e D_{11}^{-1} C_1)\, x(k) + B_3\,(-F - D_e D_{11}^{-1} C_1)\, x_c(k) + B_4\, \delta(k) \\
x_c(k+1) &= (B_1 D_{11}^{-1} + B_3 D_e D_{11}^{-1}) C_1\, x(k) \\
&\quad +(A - B_3 F - B_1 D_{11}^{-1} C_1 - B_3 D_e D_{11}^{-1} C_1)\, x_c(k) \\
\epsilon(k) &= (C_5 + D_{53} D_e D_{11}^{-1} C_1)\, x(k) + D_{53}\,(-F - D_e D_{11}^{-1} C_1)\, x_c(k)
\end{aligned}$$

or in compact notation

$$\begin{aligned}
x_{\tilde{T}}(k+1) &= \begin{bmatrix} A + B_3 D_e D_{11}^{-1} C_1 & -B_3 F - B_3 D_e D_{11}^{-1} C_1 \\ B_1 D_{11}^{-1} C_1 + B_3 D_e D_{11}^{-1} C_1 & A - B_3 F - B_1 D_{11}^{-1} C_1 - B_3 D_e D_{11}^{-1} C_1 \end{bmatrix} x_{\tilde{T}}(k) \\
&\quad + \begin{bmatrix} B_4 \\ 0 \end{bmatrix} \delta(k) \\
&= A_{\tilde{T}}\, x_{\tilde{T}}(k) + B_{\tilde{T}}\, \delta(k) \\
\epsilon(k) &= \begin{bmatrix} C_5 + D_{53} D_e D_{11}^{-1} C_1 & -D_{53} F - D_{53} D_e D_{11}^{-1} C_1 \end{bmatrix} x_{\tilde{T}}(k) \\
&= C_{\tilde{T}}\, x_{\tilde{T}}(k)
\end{aligned}$$

This is a state space realization of a transfer matrix

$$\epsilon(k) = \tilde{T}(q)\,\delta(k) = \Big(\, C_{\tilde{T}}\Big(qI - A_{\tilde{T}}\Big)^{-1} B_{\tilde{T}}\,\Big)\,\delta(k)$$

In the case of model uncertainty, the function $\tilde{T}(q)$ will be in closed loop with the uncertainty Ω. This means we have a closed loop of $\tilde{T}(q)$ and $\Omega(q)$, as depicted in figure 9.1, leading to the closed loop form

$$\Big(I - \tilde{T}(q)\Omega(q)\Big)^{-1}$$

A sufficient conditions for this closed loop to be stable is:

$$\|\tilde{T}\|_\infty = \max_{|z|\geq 1} \bar{\sigma}\Big(\tilde{T}(z)\Big) < 1$$

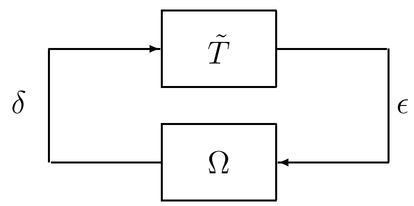

Figure 9.1: Closed loop with uncertainty block

The function $\tilde{T}$ depends on the design parameters in the tuning vectors θ and η, as defined on page 159. The sensitivity of the $\|\tilde{T}(\theta,\eta)\|_\infty$ with respect to each of the real valued parameters θ_i is derived in appendix D.
A tuning procedure for robust stability can follow the next steps: Design the controller for the initial settings θ_o and η_o of chapter 7. Compute $\|\tilde{T}\|_\infty$. If $\|\tilde{T}(\theta_o,\eta_o)\|_\infty < 1$, we already have robust stability and use the initial setting as final setting. If $\|\tilde{T}(\theta_o,\eta_o)\|_\infty \geq 1$, the controller has to be retuned to obtain robust stability:
We perform a mixed-integer optimization using $\dfrac{\partial\|\tilde{T}\|_\infty}{\partial\theta_i}$ as the gradient function for the real-valued parameters. As soon as $\|\tilde{T}\|_\infty < 1$ the optimization procedure can be terminated, or proceeded to obtain a better robustness margin ($\|\tilde{T}(\theta,\eta)\|_\infty \ll 1$).

Additive model error

We will now elaborate on the the additive model error case, so the model error is given by:

$$\Omega(q) = W_a^{-1}(q)\Big(G(q) - \hat{G}(q)\Big)$$

where $W_a(q)$ is a given weighting filter such that $\|\Omega\| \leq 1$. Note that for SISO systems this means that

$$|G(e^{j\omega}) - \hat{G}(e^{j\omega})| \leq |W_a(e^{j\omega})| \quad \forall\,\omega \in \mathbb{R}$$

and so $W_a(e^{j\omega})$ gives na upper bound on the magnitude of the model error.
We make the following assumptions:

1. $G(q)$ and $\hat{G}(q)$ are stable.

2. $H(q)$ and $\hat{H}(q)$ are stable.

3. $H^{-1}(q)$ and $\hat{H}^{-1}(q)$ are stable.

4. The MPC controller for the model $G(q)$, $H(q)$ is stabilizing.

The first three assumptions guarantee that we can study the robust control problem in the Internal Model Control (IMC) framework (see section 6.3). The fourth assumption is necessary, because it only makes sense to study robust stability if the MPC controller is stabilizing for the nominal case.
Now consider the IMC-scheme given in section by (6.7)-(6.9) in (6.7) and (6.10) in section 6.3. We assume that the nominal MPC controller is stabilizing, so $Q_1(q)$ and $Q_2(q)$ are stable. The closed loop is now given by (6.13)-(6.13):

$$\begin{aligned} v(k) &= \Big(I - Q_2(q)\hat{H}^{-1}(q)(G(q) - \hat{G}(q))\Big)^{-1}\Big(Q_1(q)\tilde{w}(k) + Q_2(q)\hat{H}^{-1}(q)H(q)e(k)\Big) \\ y(k) &= G(q)\Big(I - Q_2(q)\hat{H}^{-1}(q)(G(q) - \hat{G}(q))\Big)^{-1}\Big(Q_1(q)\tilde{w}(k) \\ &\quad + Q_2(q)\hat{H}^{-1}(q)H(q)e(k)\Big) + H(q)e(k) \end{aligned}$$

It is clear that the perturbed closed loop is stable if and only if

$$\Big(I - Q_2(q)\hat{H}^{-1}(q)(G(q) - \hat{G}(q))\Big)^{-1}$$

is stable. A necessary and sufficient condition for stability is that for all $|q| \geq 1$):

$$Q_2(q)\hat{H}^{-1}(q)(G(q) - \hat{G}(q)) \neq I$$

A sufficient condition for stability is that

$$\|Q_2\hat{H}^{-1}(G - \hat{G})\|_\infty < 1$$

This can be relaxed to

$$\|Q_2\hat{H}^{-1}W_a\|_\infty \|W_a^{-1}(G - \hat{G})\|_\infty < 1$$

We know that $\|W_a^{-1}(G - \hat{G})\|_\infty < 1$, and so a sufficient condition is

$$\|Q_2\hat{H}^{-1}W_a\|_\infty < 1$$

Tuning of the noise model

Until now we assumed that the noise model

$$H(q) = C_1(qI - A)^{-1}B_1 + I$$

was correct with a stable inverse (eigenvalues of A–B_1C_1 are all strict inside the unit circle). In the case of model uncertainty however, the state observer B_1, based on the above noise model may not work satisfactory and may even destabilize the uncertain system. In many papers, the observer matrix B_1 is therefore considered as a design variable and is tuned to obtain good robustness properties. By changing B_1, we can change the dynamics of $\tilde{T}(z)$ and thus improve the robustness of the predictive controller.
We make the following assumption:

$$D_{21} = D_{21,o}\,B_1$$

which means that matrix D_e is linear in B_1, and so there exist a matrix $D_{e,o}$ such that $D_e = D_{e,o}\,B_1$. This will make the computations much easier.
We already had the equation

$$\|Q_2\hat{H}^{-1}W_a\|_\infty < 1$$

Note that $Q_2(q)$ is given by

$$\begin{aligned} Q_2(q) &= -F(q\,I - A + B_2F)^{-1}(B_1 + B_3D_e) + D_e \\ &= \Big(-F(q\,I - A + B_2F)^{-1}(I + B_3D_{e,o}) + D_{e,o}\Big)\,B_1 \\ &= Q_{2,o}(q)\,B_1 \end{aligned}$$

So a sufficient condition for stability is

$$\|Q_{2,o}\,B_1\,\hat{H}^{-1}W_a\|_\infty < 1$$

where we now use B_1 as a tuning variable. Note that for $B_1 \longrightarrow 0$ we have that $\|Q_{2,o}\,B_1\,\hat{H}^{-1}W_a\|_\infty \longrightarrow 0$ and so by reducing the magnitude of matrix B_1 we can always find a value that will robustly stabilize the closed-loop system. A disadvantage of this procedure is that the noise model is disturbed and noise rejection is not optimal any more. To retain some disturbance rejection, we should detune the matrix B_1 not more than necessary to obtain robustness.

9.4 Robustness in LMI-based MPC

In this section we consider the problem of robust performance in MPC based on the work of Kothare *et al.* [63]. We will use the LMI-based MPC techniques as introduced in chapter 8 to guarantee robust performance. As in chapter 8, the signals $w(k)$ and $e(k)$ are assumed to be zero. How these signals can be incorporated in the design is in Kothare *et al.* [63].

We consider both unstructured and structured model uncertainty. Given the fact that w and e are zero, we get the following model uncertainty descriptions for the LMI-based case:

Unstructured uncertainty

The uncertainty is given in the following description (compare 9.1-9.6):

$$\begin{aligned} x(k+1) &= A\,x(k) + B_3\,v(k) + B_4\,\delta(k) && (9.12)\\ z(k) &= C_2\,x(k) + D_{23}\,v(k) + D_{24}\,\delta(k) && (9.13)\\ \psi(k) &= C_4\,x(k) + D_{43}\,v(k) + D_{44}\,\delta(k) && (9.14)\\ \epsilon(k) &= C_5\,x(k) + D_{53}\,v(k) && (9.15)\\ \delta(k) &= \Omega(q)\epsilon(k) && (9.16) \end{aligned}$$

where Ω has a block diagonal structure:

$$\Omega = \begin{bmatrix} \Omega_1 & & & \\ & \Omega_2 & & \\ & & \ddots & \\ & & & \Omega_r \end{bmatrix} \tag{9.17}$$

with $\Omega_i(q) : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_i}$, $i = 1, \ldots, r$ is bounded in ∞-norm:

$$\|\Omega_i\|_\infty \leq 1 \ , \qquad i = 1, \ldots, r \tag{9.18}$$

A nominal model $G_{nom} \in \mathcal{G}$ is given for $\delta(k) = 0$:

$$x(k+1) = A\,x(k) + B_3\,v(k)$$

Note that this is an extension of the definition in section 9.2, where we restricted ourselves to the 1-block case ($r = 1$).

Structured uncertainty

The uncertainty is given in the following description (compare 9.7-9.10):

$$\begin{aligned} x(k+1) &= A\,x(k) + B_3\,v(k) && (9.19)\\ z(k) &= C_2\,x(k) + D_{23}\,v(k) && (9.20)\\ \psi(k) &= C_4\,x(k) + D_{43}\,v(k) && (9.21) \end{aligned}$$

where

$$P = \begin{bmatrix} A & B_3 \\ C_2 & D_{23} \\ C_4 & D_{43} \end{bmatrix} \in \text{Co} \left\{ \begin{bmatrix} \bar{A}_1 & \bar{B}_{3,1} \\ \bar{C}_{2,1} & \bar{D}_{23,1} \\ \bar{C}_{4,1} & \bar{D}_{43,1} \end{bmatrix}, \ldots, \begin{bmatrix} \bar{A}_L & \bar{B}_{3,L} \\ \bar{C}_{2,L} & \bar{D}_{23,L} \\ \bar{C}_{4,L} & \bar{D}_{43,L} \end{bmatrix} \right\} \tag{9.22}$$

A nominal model $G_{nom} \in \mathcal{G}$ is given by

$$x(k+1) = A_{nom}\,x(k) + B_{3,nom}\,v(k)$$

Robust performance for LMI-based MPC

The robust performance MPC control problem is to find, at each time-instant k, a state-feedback law

$$v(k) = F\,x(k)$$

such that the criterion

$$\min_{F} \max_{G \in \mathcal{G}} J(k) \quad \text{where} \quad J(k) = \sum_{j=0}^{\infty} z^T(k+j)\Gamma z(k+j) \tag{9.23}$$

is optimized subject to the constraints

$$\max_{G \in \mathcal{G}} |\, \psi_\ell(k+j) \,| \le \psi_{\ell,max} \;, \quad , \text{ for } \ell = 1, \ldots, m \;, \quad \forall j \ge 0 \tag{9.24}$$

In equation (9.23) input-signal $v(k)$ is choosen such that the performance index J is minimized for the worst-case choice of G in $\mathcal{G}$, subject to constraints (9.24) for the same worst-case G. As was introduced in the previous section, the uncertainty in G can be either structured or unstructured, resulting in the theorems 36 and 37.

Theorem 36 Robust performance for unstructured model uncertainty
Given a system with uncertainty description (9.12)-(9.18) and a control problem of minimizing (9.23) subject to constraints (9.24). The state-feedback $F = Y\,S^{-1}$ minimizing the worst-case $J(k)$ can be found by solving:

$$\min_{\gamma,S,Y,T,V,\Lambda} \gamma \tag{9.25}$$

subject to

(LMI for lyapunov stability:)

$$S > 0 \tag{9.26}$$

(LMI for ellipsoid boundary:)

$$\begin{bmatrix} 1 & x^T(k|k) \\ x(k|k) & S \end{bmatrix} \ge 0 \tag{9.27}$$

(LMI for worst case MPC performance index:)

$$\begin{bmatrix} S & (AS+B_3Y)^T & (C_2S+D_{23}Y)^T\Gamma & (C_5S+D_{53}Y)^T \\ (AS+B_3Y) & S-B_4\Lambda B_4^T & -B_4\Lambda D_{24}^T\Gamma & 0 \\ \Gamma(C_2S+D_{23}Y) & -\Gamma D_{24}\Lambda B_4^T & \gamma I - \Gamma D_{24}\Lambda D_{24}^T\Gamma & 0 \\ (C_5S+D_{53}Y) & 0 & 0 & \Lambda \end{bmatrix} \ge 0 \tag{9.28}$$

$$\Lambda > 0 \tag{9.29}$$

where Λ is partitioned as Ω:

$$\Lambda = diag(\lambda_1 I_{n_1},\, \lambda_2 I_{n_2}, \ldots, \lambda_r I_{n_r}) > 0 \tag{9.30}$$

(LMIs for constraints on ψ:)

$$\begin{bmatrix} \psi^2_{\ell,max} S & (C_5 S + D_{53} Y)^T & (C_4 S + D_{43} Y)^T E_\ell^T \\ C_5 S + D_{53} Y & V^{-1} & 0 \\ E_\ell (C_4 S + D_{43} Y) & 0 & I - E_\ell D_{44} V^{-1} D_{11}^T E_\ell^T \end{bmatrix} \geq 0, \qquad \ell = 1, \ldots, m \tag{9.31}$$

$$V^{-1} > 0 \tag{9.32}$$

where V is partitioned as Ω:

$$V = diag(v_1 I_{n_1},\, v_2 I_{n_2}, \ldots, v_r I_{n_r}) > 0 \tag{9.33}$$

and $E_\ell = \begin{bmatrix} 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \end{bmatrix}$ *with the* 1 *on the* ℓ*-th position.*

LMI (9.26) guarantees Lyapunov stability, LMI (9.27) describes an invariant ellipsoid for the state, (9.28), (9.29) are the LMIs for the worst-case MPC-criterion and (9.31), (9.33) and (9.32) correspond to the state constraint on ψ .

Proof of theorem 36:

The LMI's (9.27) and (9.29) for optimizing (9.25) are given in Kothare *et al.*[63]. In the notation of the same paper (9.31)-(9.32) can be derived as follows:
For any admissible $\Delta(k)$ and $l = 1, \ldots, m$ we have

$$\begin{aligned} \psi(k) &= E_\ell^T C_4\, x(k) + E_\ell^T D_{43}\, v(k) + E_\ell^T D_{44}\, \delta(k) & (9.34) \\ &= E_\ell^T (C_4 + D_{43} F)\, x(k) + E_\ell^T D_{44}\, \delta(k) & (9.35) \end{aligned}$$

and so for $l = 1, \ldots, m$ and $j \geq 0$:

$$\begin{aligned} \max_{j\geq 0} |\, \Psi_l(k+j|k)\, | &= \max_{j\geq 0} |\, E_\ell^T (C_4 + D_{43} F)\, x(k+j|k) + E_\ell^T D_{44}\, \delta(k+j|k)\, | \\ &\leq \max_{z^T z \leq 1} |E_\ell^T (C_4 + D_{43} F) S^{\frac{1}{2}} z + E_\ell^T D_{44}\, \delta(k+j|k)| \end{aligned}$$

Further the derivation of (9.31)-(9.32) is analoguous to the derivation of the LMI for an output constraint in Kothare *et al.*[63].

In the notation of the same paper (9.28) can be derived as follows:

$$V(x(k{+}j{+}1|k)) =$$

$$= \begin{bmatrix} x(k+j|k) \\ \delta(k+j|k) \end{bmatrix}^T \begin{bmatrix} (A + B_3 F)^T P (A + B_3 F) & (A + B_3 F)^T P B_4 \\ B_4^T P (A + B_3 F) & B_4^T P B_4 \end{bmatrix} \begin{bmatrix} x(k+j|k) \\ \delta(k+j|k) \end{bmatrix}$$

$$V(x(k+j|k)) = \begin{bmatrix} x(k+j|k) \\ \delta(k+j|k) \end{bmatrix}^T \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(k+j|k) \\ \delta(k+j|k) \end{bmatrix}$$

$$\Delta J(k+j) =$$

$$\begin{aligned} &= z(k+j)^T \Gamma z(k+j) \\ &= \Big((C_2 + D_{23}F)x(k+j) + D_{24}\delta(k+j)\Big)^T R\Big((C_2 + D_{23}F)x(k+j) + D_{24}\delta(k+j)\Big) \\ &= \begin{bmatrix} x(k+j|k) \\ \delta(k+j|k) \end{bmatrix}^T \begin{bmatrix} (C_2 + D_{23}F)^T\Gamma(C_2 + D_{23}F) & (C_2 + D_{23}F)^T\Gamma D_{24} \\ D_{24}^T\Gamma(C_2 + D_{23}F) & D_{24}^T\Gamma D_{24} \end{bmatrix} \begin{bmatrix} x(k+j|k) \\ \delta(k+j|k) \end{bmatrix} \end{aligned}$$

And so the condition

$$V(x(k+j+1|k)) - V(x(k+j|k)) \leq -\Delta J(k+j)$$

result in the inequality

$$\begin{bmatrix} x(k+j|k) \\ \delta(k+j|k) \end{bmatrix}^T \begin{bmatrix} (A+B_3F)^T P(A+B_3F) - P+ & (A+B_3F)^T PB_4 + (C_2+D_{23}F)^T\Gamma D_{24} \\ +(C_2+D_{23}F)^T\Gamma(C_2+D_{23}F) & \\ B_4^T P(A+B_3F) + D_{24}^T\Gamma(C_2+D_{23}F) & B_4^T PB_4 + D_{24}^T\Gamma D_{24} \end{bmatrix} \cdot$$

$$\cdot \begin{bmatrix} x(k+j|k) \\ \delta(k+j|k) \end{bmatrix} \leq 0$$

This condition together with condition

$$\delta_i(k+j|k)^T\delta_i(k+j|k) \leq$$

$$\leq x(k+j|k)(C_{22,i} + D_{53,i}F)^T(C_{22,i} + D_{53,i}F)x(k+j|k)\ , \quad i = 1, 2, \ldots r$$

is satisfied if there exists a matrix $Z \geq 0$ such that

$$\begin{bmatrix} (A + B_3F)^T P(A + B_3F) - P+ & (A + B_3F)^T PB_4 + (C_2 + D_{23}F)^T\Gamma D_{24} \\ +(C_2 + D_{23}F)^T\Gamma(C_2 + D_{23}F)+ & \\ +(C_5 + D_{53}F)^T(C_5 + D_{53}F) & \\ B_4^T P(A + B_3F) + D_{24}^T\Gamma(C_2 + D_{23}F) & B_4^T PB_4 + D_{24}^T\Gamma D_{24} - Z \end{bmatrix} \leq 0$$

Susbtituting $P = \gamma S^{-1}$ and $F = YS^{-1}$ we find

$$\begin{bmatrix} (AS + B_3Y)^T\gamma S^{-1}(AS + B_3Y) - \gamma S+ & (AS + B_3Y)^T\gamma S^{-1}B_4+ \\ +(C_2S + D_{23}Y)^T\Gamma(C_2S + D_{23}Y) + SQS+ & +(C_2S + D_{23}Y)^T\Gamma D_{24} \\ +(C_5S + D_{53}Y)^T\Lambda(C_5S + D_{53}Y) & \\ B_4^T\gamma S^{-1}(AS + BY) + D_{24}^T\Gamma(CS + DY) & B_4^T\gamma S^{-1}B_4 + D_{24}^T\Gamma D_{24} - Z \end{bmatrix} \leq 0$$

By susbtitution of $Z = \gamma\Lambda^{-1}$ and by applying some matrix operations we obtain equation 9.28.

□ End Proof

Theorem 37 Robust performance for structured model uncertainty
Given a system with uncertainty description (9.19)-(9.22) and a control problem of minimizing (9.23) subject to constraint (9.24). The state-feedback $F = Y\,S^{-1}$ minimizing the worst-case $J(k)$ can be found by solving:

$$\min_{\gamma,S,Y} \gamma \tag{9.36}$$

subject to

(LMI for lyapunov stability:)

$$S > 0 \tag{9.37}$$

(LMI for ellipsoid boundary:)

$$\begin{bmatrix} 1 & x^T(k|k) \\ x(k|k) & S \end{bmatrix} \geq 0 \tag{9.38}$$

(LMIs for worst case MPC performance index:)

$$\begin{bmatrix} S & S\bar{A}_i^T + Y^T\bar{B}_{3,i}^T & (S\bar{C}_{2,i}^T + Y^T\bar{D}_{23,i}^T)\Gamma^{1/2} \\ \bar{A}_iS + \bar{B}_{3,i}Y & S & 0 \\ \Gamma^{1/2}(\bar{C}_{2,i}S + \bar{D}_{23,i}Y) & 0 & \gamma I \end{bmatrix} \geq 0 \quad , \quad i = 1,\dots,L \tag{9.39}$$

(LMIs for constraints on ψ:)

$$\begin{bmatrix} S & (\bar{C}_{4,i}S + \bar{D}_{43,i}Y)^T E_\ell^T \\ E_\ell(\bar{C}_{4,i}S + \bar{D}_{43,i}Y) & \psi^2_{\ell,max} \end{bmatrix} \geq 0, \quad \ell = 1,\dots,m, \; i = 1,\dots,L \tag{9.40}$$

where $E_\ell = \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{bmatrix}$ *with the* 1 *on the* ℓ*-th position.*

LMI (9.37) guarantees Lyapunov stability, LMI (9.38) describes an invariant ellipsoid for the state, (9.39) gives the LMIs for the worst-case MPC-criterion and (9.40) correspond to the state constraint on ψ .

Proof of theorem 37:

The LMI (9.38) for solving (9.36) is given in Kothare *et al.*[63]. In the notation of the same paper (9.40) can be derived as follows:
We have, for $l = 1,\dots,m$:

$$\begin{aligned} \psi_\ell(k) &= E_\ell^T C_4\, x(k) + E_\ell^T D_{43}\, v(k) &(9.41)\\ &= E_\ell^T (C_4 + D_{43}F)\, x(k) &(9.42) \end{aligned}$$

and so all possible matrices $\begin{bmatrix} C_4 & D_{43} \end{bmatrix}$ and $l = 1, \ldots, m$ there holds

$$\begin{aligned}
\max_{j \geq 0} |\, \psi_\ell(k+j|k) \,| &= \max_{j \geq 0} |\, E_\ell^T (C_4 + D_{43}F)\, x(k+j|k) \,| \\
&\leq \max_{z^T z \leq 1} |E_\ell^T (C_4 + D_{43}F) S^{\frac{1}{2}} z| \quad , \quad j \geq 0 \\
&= \bar{\sigma}\Big(E_\ell^T (C_4 + D_{43}F) S^{\frac{1}{2}} \Big) \quad , \quad j \geq 0
\end{aligned}$$

Further the derivation of (9.40) is analoguous to the derivation of the LMI for an output constraint in Kothare *et al.*[63].

In the notation of the same paper (9.39) can be derived as follows. The equation

$$(A + B_3 F)^T P (A + B_3 F) - P + Q + F^T R F \leq 0$$

is replaced by:

$$\begin{aligned}
&(A(k+j) + B_3 F)^T P (A + B_3 F) - P + \\
&\quad +(C_2 + D_{23}F)^T \Gamma (C_2 + D_{23}F) \leq 0
\end{aligned}$$

For $P = \gamma S^{-1}$ and $F = Y S^{-1}$ we find

$$\begin{bmatrix} S & SA^T + Y^T B_3^T & (SC_2^T + Y^T D_{23}^T)\Gamma^{1/2} \\ AS + B_3 Y & S & 0 \\ \Gamma^{1/2}(C_2 S + D_{23} Y) & 0 & \gamma I \end{bmatrix} \geq 0 \qquad (9.43)$$

which is affine in $[A\ B_3\ C_2\ D_{23}]$. Hence the condition becomes equation (9.39).

□ End Proof

Chapter 10

Case study

Appendix A: Quadratic Programming

In this appendix we will consider optimization problems with a quadratic object function with linear constraints, denoted as the quadratic programming problem. There are two types of QP (quadratic programming) problems:

Definition 38 The quadratic programming problem (Type 1):
Minimize the object function

$$F(\theta) = \frac{1}{2}\,\theta^T\,H\,\theta + f^T\,\theta$$

over variable θ, where H is a semi-positive definite matrix, subject to the linear inequality constraint

$$A\,\theta\,\leq\,b$$

and the non-negative constraint

$$\theta \geq 0$$

□ *End Definition*

Definition 39 The quadratic programming problem (Type 2):
Minimize the object function

$$F(\theta) = \frac{1}{2}\,\theta^T\,H\,\theta + f^T\,\theta$$

over variable θ, where H is a semi-positive definite matrix, subject to the linear equality constraint

$$A\,\theta\,=\,b$$

and the non-negative constraint

$$\theta \geq 0$$

□ *End Definition*

Remark:
If H is a non-symmetric matrix, we can define

$$H_{new} = \frac{1}{2}\,(H + H^T)$$

Substitution of this H_{new} for H does not change the object function.

Remark:
A 'Type 1' quadratic programming problem can be converted into a 'Type 2' quadratic programming problem by introducing a 'slack'-variable y such that

$$A\,\theta + y = b \quad where \quad y \geq 0$$

We define

$$\bar{H} = \begin{bmatrix} H & 0 \\ 0 & 0 \end{bmatrix} \;,\; \bar{f} = \begin{bmatrix} f \\ 0 \end{bmatrix} \;,\; \bar{A} = \begin{bmatrix} A & I \end{bmatrix} \;,\; \bar{\theta} = \begin{bmatrix} \theta \\ y \end{bmatrix}$$

and obtain a 'Type 2' problem:

$$F(\theta) = \frac{1}{2}\,\bar{\theta}^T\,\bar{H}\,\bar{\theta} + \bar{f}^T\,\bar{\theta}$$

$$\bar{A}\,\bar{\theta} \;=\; b$$

$$\bar{\theta} \geq 0$$

The unconstrained quadratic programming problem:
If the constraints are ommitted and only the quadratic object function is considered, an analytic solution exists: An extremum of the object function

$$F(\theta) = \frac{1}{2}\,\theta^T\,H\,\theta + f^T\,\theta$$

is found when the gradient ∇F is equal to zero

$$\nabla F(\theta) = H\,\theta + f = 0$$

If H is non-singular the extremum is reached for

$$\theta = -H^{-1}\,f$$

Because H is a positive definite matrix, the extremum will be a minimum.

A.1 Quadratic programming algorithm

Consider the object function

$$F(\theta) = \frac{1}{2}\,\theta^T\,H\,\theta + f^T\,\theta$$

with the linear equality constraint

$$A\,\theta \;=\; b$$

and the non-negative constraint

$$\theta \geq 0$$

for a semi-positive definite matrix H.

For an inequality/equality constrained optimization problem necessary conditions for an extremum of the function $F(\theta)$ in θ, satisfying $h(\theta) = 0$ and $g(\theta) \leq 0$, are given by the Kuhn-Tucker conditions:
There exist vectors λ and μ, such that

$$\begin{aligned}
\nabla F(\theta) + \nabla g(\theta)\,\mu + \nabla h(\theta)\,\lambda &= 0 \\
\mu^T\,g(\theta) &= 0 \\
\mu &\geq 0 \\
h(\theta) &= 0 \\
g(\theta) &\leq 0
\end{aligned}$$

The Kuhn-Tucker conditions for this quadratic programming problem are given by:

$$H\,\theta + A^T\,\lambda - \mu \;=\; -f \tag{10.1}$$
$$\theta^T\,\mu \;=\; 0 \tag{10.2}$$
$$A\,\theta \;=\; b \tag{10.3}$$
$$\theta\,,\,\mu \;\geq\; 0 \tag{10.4}$$

We recognize equality constraints (10.3),(10.1) and a non-negative constraint (10.4), two ingredients of a general linear programming problem. On the other hand, we miss the object function, and we have an additional nonlinear equality constraint (10.2). (Note that the multiplication $\theta^T\,\mu$ is nonlinear in the new parameter vector $(\theta, \mu, \lambda)^T$.) An object function can be obtained by introducing two slack variables u_1 and u_2, and defining the problem:

$$A\,\theta + u_1 \;=\; b \tag{10.5}$$
$$H\,\theta + A^T\,\lambda - \mu + u_2 \;=\; -f \tag{10.6}$$
$$\theta\,,\,\mu\,,\,u_1\,,\,u_2 \;\geq\; 0 \tag{10.7}$$
$$\theta^T\,\mu \;=\; 0 \tag{10.8}$$

while minimizing

$$\min_{\theta,\mu,\lambda,u_1,u_2} u_1 + u_2 \tag{10.9}$$

Construct the matrices

$$A_0 = \begin{bmatrix} A & 0 & 0 & I & 0 \\ H & A^T & -I & 0 & I \end{bmatrix} \quad b_0 = \begin{bmatrix} b \\ -f \end{bmatrix} \quad f_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad \theta_0 = \begin{bmatrix} \theta \\ \lambda \\ \mu \\ u_1 \\ u_2 \end{bmatrix} \tag{10.10}$$

and the general linear programming problem can be written as:

$$\min f_0^T\,\theta_0 \quad \text{where} \quad A_0\,\theta_0 = b_0 \quad \text{and} \quad \theta_0 \geq 0$$

with the additional nonlinear constraint $\theta^T\,\mu = 0$.
It may be clear from the Kuhn-Tucker conditions in (10.3-10.2) that an optimum is only reached when u_1 and u_2 in (10.5-10.7) become zero. This can be achieved by using a simplex algorithm which is modified in the sense that there is an extra condition on finding feasible basic solutions namely that $\theta^T\,\mu = 0$.

Note from (10.4) and (10.2) that

$$\theta^T\,\mu = \theta_1\,\mu_1 + \theta_2\,\mu_2 + \ldots + \theta_n\,\mu_n = 0 \;\text{ where }\; \theta_i,\, \mu_i \geq 0 \;\text{ for }\; i = 1,\ldots,n$$

means that the nonlinear constraint (10.2) can be rewritten as:

$$\theta_i\,\mu_i = 0 \;\text{ for }\; i = 1,\ldots,n$$

and thus either $\theta_i = 0$ or $\mu_i = 0$. The extra feasibility condition on a basic solution now becomes $\theta_i\,\mu_i = 0$ for $i = 1,\ldots,n$.

Remark:
In the above derivations we assumed $b \geq 0$ and $f \leq 0$. If this is not the case, we have to change the corresponding signs of u_1 and u_2 in the equations (10.5) and (10.6).

We will sketch the modified simplex algorithm on the basis of an example.

Modified simplex algorithm:
Consider the following type 2 quadratic programming problem:

$$\min_{\theta} \frac{1}{2}\,\theta^T\,H\,\theta + f^T\,\theta$$

$$A\,\theta = b$$

$$\theta \geq 0$$

where

$$H = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ 0 & 1 & 4 & 2 \\ 0 & 0 & 2 & 4 \end{bmatrix} \qquad f = \begin{bmatrix} 0 \\ -3 \\ -2 \\ 0 \end{bmatrix} \qquad A = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 2 \end{bmatrix} \qquad b = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

Before we start the modified simplex algorithm, we check if a simple solution can be found. If we minimize the unconstrained function

$$\min_{\theta} \frac{1}{2} \theta^T H \theta + f^T \theta$$

we obtain $\theta = \begin{bmatrix} 7 & 14 & -4 & 2 \end{bmatrix}^T$. This solution is not feasible, because neither the equality constraint $A\theta = b$, nor the non-negative constraint $\theta \geq 0$ are satisfied.

We construct the matrix A_0, the vectors b_0, f_0 and the parameter vector θ_0 according to equation (10.10). Note that θ_0 is a 16×1-vector with the elements θ is a 4×1-vector, λ is a 2×1-vector, μ is a 4×1-vector, u_1 is a 2×1-vector and u_2 is a 4×1-vector.
The modified simplex method starts with finding a first feasible solution. We choose B to contain the six most-right columns of A_0. We obtain the solution

$$\theta = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \lambda = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \mu = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad u_1 = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \qquad u_2 = \begin{bmatrix} 0 \\ 3 \\ 2 \\ 0 \end{bmatrix}$$

The solution is feasible because $\theta\,,\,\mu\,,\,u_1\,,\,u_2 \geq 0$ and $\theta_i\,\mu_i = 0$ for $i = 1,2,3,4$. However, the optimum is not found yet, because $u_1\,,\,u_2 > 0$.
After a finite number of iterations the optimum is found by selecting columns 1, 2, 5, 6, 9 and 10 of the matrix A_0 for the matrix B. We find the optimal solution

$$\theta = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix} \qquad \lambda = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \mu = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 2 \end{bmatrix} \qquad u_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad u_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

and the Kuhn-Tuker conditions (10.3-10.2) are satisfied for these values.

Algorithms that use a modified version of the simplex method are Wolfe's algorithm [133], and the pivoting algorithm of Lemke [71].

Appendix B: Basic State Space Operations

Cascade:

$$G(q) \quad \longrightarrow \quad \hat{G}(q) = G_1(q) \cdot G_2(q)$$

$$G_1 \equiv \left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array}\right] \qquad G_2 \equiv \left[\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array}\right]$$

$$\begin{aligned}
\hat{G} &\equiv \left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array}\right] \cdot \left[\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array}\right] \\
&= \left[\begin{array}{cc|c} A_1 & B_1C_2 & B_1D_2 \\ 0 & A_2 & B_2 \\ \hline C_1 & D_1C_2 & D_1D_2 \end{array}\right] \\
&= \left[\begin{array}{cc|c} A_2 & 0 & B_2 \\ B_1C_2 & A_1 & B_1D_2 \\ \hline D_1C_2 & C_1 & D_1D_2 \end{array}\right]
\end{aligned}$$

Parallel:

$$G(q) \quad \longrightarrow \quad \hat{G}(q) = G_1(q) + G_2(q)$$

$$G_1 \equiv \left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array}\right] \qquad G_2 \equiv \left[\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array}\right]$$

$$\hat{G} \equiv \left[\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array}\right] + \left[\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array}\right]$$

$$= \left[\begin{array}{cc|c} A_1 & 0 & B_1 \\ 0 & A_2 & B_2 \\ \hline C_1 & C_2 & D_1 + D_2 \end{array}\right]$$

Change of variables:

$$\begin{array}{lll} x & \longrightarrow & \hat{x} = T\,x \qquad\qquad T \text{ is invertible} \\ u & \longrightarrow & \hat{u} = P\,u \qquad\qquad P \text{ is invertible} \\ y & \longrightarrow & \hat{y} = R\,y \end{array}$$

$$G \equiv \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

$$\hat{G} \equiv \left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array}\right] = \left[\begin{array}{c|c} TAT^{-1} & TBP^{-1} \\ \hline RCT^{-1} & RDP^{-1} \end{array}\right]$$

State feedback:

$$u \longrightarrow \hat{u} + F\,x$$

$$G \equiv \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

$$\hat{G} \equiv \left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array}\right] = \left[\begin{array}{c|c} A + BF & B \\ \hline C + DF & D \end{array}\right]$$

Output injection:

$$x(k+1) = Ax(k) + Bu(k) \quad \longrightarrow \quad \hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Hy(k)$$

$$G \;\equiv\; \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

$$\hat{G} \;\equiv\; \left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array}\right] = \left[\begin{array}{c|c} A+HC & B+HD \\ \hline C & D \end{array}\right]$$

Transpose:

$$G(q) \quad \longrightarrow \quad \hat{G}(q) = G^T(q)$$

$$G \;\equiv\; \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

$$\hat{G} \;\equiv\; \left[\begin{array}{c|c} A^T & C^T \\ \hline B^T & D^T \end{array}\right]$$

Left (right) Inversion:

$$G(q) \quad \longrightarrow \quad \hat{G}(q) = G^+(q)$$

G^+ is a left (right) inverse of G.

$$G \;\equiv\; \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

$$\hat{G} \;\equiv\; \left[\begin{array}{c|c} A - BD^+C & BD^+ \\ \hline -D^+C & D^+ \end{array}\right]$$

where D^+ is the left (right) inverse of D.
For $G(q)$ is square, we obtain $G^+(q) = G^{-1}(q)$ and $D^+ = D^{-1}$.

Appendix C: Some results from linear algebra

Left inverse and Left complement:

Consider a full-column rank matrix $M \in \mathrm{IR}^{m \times n}$ for $m \geq n$. Let the singular value decompostion of M be given by:

$$M = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T$$

The left inverse is define as

$$M^{\ell} = V\,\Sigma^{-1}\,U_1^T$$

and the left complement is defined as

$$M^{\ell\perp} = U_2^T$$

The left-inverse and left-complement have the following properties:

$$\begin{aligned} M^{\ell}\,M &= I \\ M^{\ell\perp}\,M &= 0 \end{aligned}$$

Right inverse and Right complement:

Consider a full-row rank matrix $M \in \mathrm{IR}^{m \times n}$ for $m \leq n$. Let the singular value decompostion of M be given by:

$$M = U \begin{bmatrix} \Sigma & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

The right inverse is define as

$$M^{r} = V_1\,\Sigma^{-1}\,U^T$$

and the right complement is defined as

$$M^{r\perp} = V_2$$

The right-inverse and right-complement have the following properties:

$$\begin{aligned} M\,M^{r} &= I \\ M\,M^{r\perp} &= 0 \end{aligned}$$

Singular value decomposition

If $A \in \mathbb{R}^{m\times n}$ then there exist orthogonal matrices

$$U = \left[\begin{array}{ccc} u_1 & \cdots & u_m \end{array}\right] \in \mathbb{R}^{m\times m}$$

and

$$V = \left[\begin{array}{ccc} v_1 & \cdots & v_n \end{array}\right] \in \mathbb{R}^{n\times n}$$

such that

$$U^T A V = \Sigma$$

where

$$\Sigma = \left[\begin{array}{ccc|ccc} \sigma_1 & & & 0 & \cdots & 0 \\ & \ddots & & \vdots & & \vdots \\ & & \sigma_m & 0 & \cdots & 0 \end{array}\right] \quad \text{for } \ m \leq n$$

$$\Sigma = \left[\begin{array}{ccc} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ \hline 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{array}\right] \quad \text{for } \ m \geq n$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$, $p = \min(m, n)$. This means that

$$A = U \Sigma V^T$$

This decomposition of A is known as *the singular value decomposition of* A. The σ_i are known as the *singular values* of A (and are usually arranged in decending order).

Property:
U contains the eigenvectors of AA^T. V contains the eigenvectors of A^TA. Σ is a diagonal matrix whose entries are the nonnegative square roots of the eigenvalues A^TA (for $m \geq n$) or AA^T (for $m \leq n$).

Property:
Let σ_{min} and σ_{max} be the smallest and largest singular value of a square matrix A. Further let λ_i denote the eigenvalues of A. Then there holds:

$$\sigma_{min} \leq |\lambda_i| \leq \sigma_{max} \quad , \quad \forall i$$

Appendix D: Gradients of performance criteria

In this section the derivatives $\dfrac{\partial \chi}{\partial \theta_i}$ are derived for the performance specifications of section 7.2.

The response $s_i(k)$ for $i = r, d$ can be calculated from the closed loop state space matrices:

$$s_i(k) = \sum_{j=0}^{k-1} C_{T2} A_T^{k-j-1} B_{T2} \tilde{w}_i(j) \qquad , \quad i = r, d$$

For all χ there holds:

$$\chi \;=\; \chi(A_T(\theta,\eta), B_{T1}(\theta,\eta), B_{T2}(\theta,\eta), C_{T1}(\theta,\eta), C_{T2}(\theta,\eta), D_{T1}(\theta,\eta), D_{T2}(\theta,\eta))$$

and so

$$\frac{\partial \chi}{\partial \theta_i} \;=\; f\left(\frac{\partial A_T(\theta,\eta)}{\partial \theta_i}, \frac{\partial B_{T1}(\theta,\eta)}{\partial \theta_i}, \frac{\partial B_{T2}(\theta,\eta)}{\partial \theta_i}, \frac{\partial C_{T1}(\theta,\eta)}{\partial \theta_i}, \frac{\partial C_{T2}(\theta,\eta)}{\partial \theta_i}, \frac{\partial D_{T1}(\theta,\eta)}{\partial \theta_i}, \frac{\partial D_{T2}(\theta,\eta)}{\partial \theta_i}\right)$$

1. Overshoot of output signal on step reference signal

Let $k_o = \arg\max_k(\ s_{yr}(k) - 1\ )$. Then

$$\chi_{os} = (\ s_{yr}(k_o) - 1\ ) \qquad \text{with} \quad s_{yr}(k_o) = \sum_{j=0}^{k_o-1} C_{T2} A_T^{k_o-j-1} B_{T2} \tilde{w}_r(j)$$

$$\frac{\partial \chi_{os}}{\partial \theta_i} = \frac{\partial s_{yr}(k_o) - 1}{\partial \theta_i} = \frac{\partial s_{yr}(k_o)}{\partial \theta_i}$$

where

$$\frac{\partial s_{yr}(k_o)}{\partial \theta_i} \;=\; \sum_{j=0}^{k_o-1} C_{T2} \frac{\partial A_T^{k_o-j-1}}{\partial \theta_i} B_{T2} \tilde{w}_r(j)$$

$$
\begin{aligned}
&+ \sum_{j=0}^{k_o-1} C_{T2} A_T^{k_o-j-1} \frac{\partial B_{T2}}{\partial \theta_i} \tilde{w}_r(j) + \sum_{j=0}^{k_o-1} \frac{\partial C_{T2}}{\partial \theta_i} A_T^{k_o-j-1} B_{T2} \tilde{w}_r(j) \\
&= \sum_{j=0}^{k_o-1} C_{T2} \sum_{i=0}^{k_o-j-1} A_T^i \frac{\partial A}{\partial \theta_i} A_T^{k_o-j-1-i} B_{T2} \tilde{w}_r(j) \\
&+ \sum_{j=0}^{k_o-1} C_{T2} A_T^{k_o-j-1} \frac{\partial B_{T2}}{\partial \theta_i} \tilde{w}_r(j) + \sum_{j=0}^{k_o-1} \frac{\partial C_{T2}}{\partial \theta_i} A_T^{k_o-j-1} B_{T2} \tilde{w}_r(j) \qquad (10.11)
\end{aligned}
$$

2. Rise time of output signal on step reference signal

Let $k_o = \chi_{rt}$

$$\hat{\chi}_{rt} = \frac{0.8 + (k_o - 1)s_{yr}(k_o) - k_o s_{yr}(k_o - 1)}{s_{yr}(k_o) - s_{yr}(k_o - 1)}$$

$$
\begin{aligned}
\frac{\partial \hat{\chi}_{rt}}{\partial \theta_i} &= \frac{\frac{\partial}{\partial \theta_i} 0.8 + (k_o - 1)s_{yr}(k_o) - k_o s_{yr}(k_o - 1)}{s_{yr}(k_o) - s_{yr}(k_o - 1)} \\
&- \left(\frac{\partial s_{yr}(k_o) - s_{yr}(k_o - 1)}{\partial \theta_i} \right) \left(\frac{0.8 + (k_o - 1)s_{yr}(k_o) - k_o s_{yr}(k_o - 1)}{\left(s_{yr}(k_o) - s_{yr}(k_o - 1) \right)^2} \right) \\
&= \frac{(k_o - 1)\frac{\partial s_{yr}(k_o)}{\partial \theta_i} - k_o \frac{\partial s_{yr}(k_o - 1)}{\partial \theta_i}}{s_{yr}(k_o) - s_{yr}(k_o - 1)} \\
&- \left(\frac{\partial s_{yr}(k_o)}{\partial \theta_i} - \frac{\partial s_{yr}(k_o - 1)}{\partial \theta_i} \right) \left(\frac{0.8 + (k_o - 1)s_{yr}(k_o) - k_o s_{yr}(k_o - 1)}{\left(s_{yr}(k_o) - s_{yr}(k_o - 1) \right)^2} \right)
\end{aligned}
$$

where $\frac{\partial s_{yr}(k_o)}{\partial \theta_i}$ and $\frac{\partial s_{yr}(k_o - 1)}{\partial \theta_i}$ are given by as (10.11) for k_o is equal to χ_{rt}.

3. Settling time of output signal on step reference signal

Let $k_o = \chi_{st}$

$$\hat{\chi}_{st} = \frac{0.05 + (k_o - 1)|s_{yr}(k_o)| - k_o |s_{yr}(k_o - 1)|}{|s_{yr}(k_o)| - |s_{yr}(k_o - 1)|}$$

$$
\begin{aligned}
\frac{\partial \hat{\chi}_{st}}{\partial \theta_i} &= \frac{(k_o - 1)\frac{\partial}{\partial \theta_i}|s_{yr}(k_o)| - k_o \frac{\partial}{\partial \theta_i}|s_{yr}(k_o - 1)|}{s_{yr}(k_o) - s_{yr}(k_o - 1)} \\
&- \left(\frac{\partial |s_{yr}(k_o)|}{\partial \theta_i} - \frac{\partial |s_{yr}(k_o - 1)|}{\partial \theta_i} \right) \left(\frac{0.8 + (k_o - 1)|s_{yr}(k_o)| - k_o |s_{yr}(k_o - 1)|}{\left(|s_{yr}(k_o)| - |s_{yr}(k_o - 1)| \right)^2} \right)
\end{aligned}
$$

where

$$\frac{\partial|s_{yr}(k_o)|}{\partial\theta_i} = \mathrm{sgn}\Big(s_{yr}(k_o)\Big)\frac{\partial s_{yr}(k_o)}{\partial\theta_i}$$

and $\dfrac{\partial s_{yr}(k_o)}{\partial\theta_i}$ and $\dfrac{\partial s_{yr}(k_o-1)}{\partial\theta_i}$ are given by as (10.11) for k_o is equal to χ_{st}.

4. Peak value of input signal on step reference signal

Let $k_o = \arg\max_k \mid s_{vr}(k) \mid$. Then

$$\chi_{pvr} = \mid s_{vr}(k_o) \mid = \sum_{j=0}^{k_o-1} C_{T1}A_T^{k_o-j-1}B_{T2}\tilde{w}_r(j)$$

$$\begin{aligned}
\frac{\partial\chi_{pvr}}{\partial\theta_i} &= \sum_{j=0}^{k_o-1} C_{T1}\frac{\partial A_T^{k_o-j-1}}{\partial\theta_i}B_{T2}\tilde{w}_r(j) \\
&+ \sum_{j=0}^{k_o-1} C_{T1}A_T^{k_o-j-1}\frac{\partial B_{T2}}{\partial\theta_i}\tilde{w}_r(j) + \sum_{j=0}^{k_o-1}\frac{\partial C_{T1}}{\partial\theta_i}A_T^{k_o-j-1}B_{T2}\tilde{w}_r(j) \\
&= \sum_{j=0}^{k_o-1} C_{T1}\sum_{i=0}^{k_o-j-1} A_T^i\frac{\partial A}{\partial\theta_i}A_T^{k_o-j-1-i}B_{T2}\tilde{w}_r(j) \\
&+ \sum_{j=0}^{k_o-1} C_{T1}A_T^{k_o-j-1}\frac{\partial B_{T2}}{\partial\theta_i}\tilde{w}_r(j) + \sum_{j=0}^{k_o-1}\frac{\partial C_{T1}}{\partial\theta_i}A_T^{k_o-j-1}B_{T2}\tilde{w}_r(j)
\end{aligned}$$

5. Peak value of output signal on step disturbance signal

Let $k_o = \arg\max_k \mid s_{yd}(k) \mid$. Then

$$\chi_{pyd} = \mid s_{yd}(k_o) \mid = \sum_{j=0}^{k_o-1} C_{T2}A_T^{k_o-j-1}B_{T2}\tilde{w}_d(j)$$

$$\begin{aligned}
\frac{\partial\chi_{pyd}}{\partial\theta_i} &= \sum_{j=0}^{k_o-1} C_{T2}\frac{\partial A_T^{k_o-j-1}}{\partial\theta_i}B_{T2}\tilde{w}_d(j) \\
&+ \sum_{j=0}^{k_o-1} C_{T2}A_T^{k_o-j-1}\frac{\partial B_{T2}}{\partial\theta_i}\tilde{w}_d(j) + \sum_{j=0}^{k_o-1}\frac{\partial C_{T2}}{\partial\theta_i}A_T^{k_o-j-1}B_{T2}\tilde{w}_d(j) \\
&= \sum_{j=0}^{k_o-1} C_{T2}\sum_{i=0}^{k_o-j-1} A_T^i\frac{\partial A}{\partial\theta_i}A_T^{k_o-j-1-i}B_{T2}\tilde{w}_d(j) \\
&+ \sum_{j=0}^{k_o-1} C_{T2}A_T^{k_o-j-1}\frac{\partial B_{T2}}{\partial\theta_i}\tilde{w}_d(j) + \sum_{j=0}^{k_o-1}\frac{\partial C_{T2}}{\partial\theta_i}A_T^{k_o-j-1}B_{T2}\tilde{w}_d(j)
\end{aligned}$$

6. RMS mistracking on zero-mean white noise signal

$$\chi_{rm} = \left(\int_{\pi}^{\pi} \mid M_{21}(e^{j\omega}) \mid d\omega \right)^{1/2}$$

$$\begin{aligned}
\frac{\partial \chi_{rm}}{\partial \theta_i} &= \frac{\partial}{\partial \theta_i} \left(\int_{\pi}^{\pi} \mid M_{21}(e^{j\omega}) \mid^2 d\omega \right)^{1/2} \\
&= \chi_{rm}^{-1} \frac{\partial}{\partial \theta_i} \int_{\pi}^{\pi} \mid M_{21}(e^{j\omega}) \mid^2 d\omega \\
&= \chi_{rm}^{-1} \int_{\pi}^{\pi} \frac{\partial \mid M_{21}(e^{j\omega}) \mid^2}{\partial \theta_i} d\omega \\
&= \chi_{rm}^{-1} \int_{\pi}^{\pi} \mathrm{Re} \left(\frac{\partial\, M_{21}(e^{j\omega})}{\partial \theta_i} \bar{M}_{21}(e^{j\omega}) \right) d\omega
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial\, M_{21}(e^{j\omega})}{\partial \theta_i} &= \frac{\partial C_{T2}(e^{j\omega} I - A_T)^{-1} B_{T1} + I}{\partial \theta_i} \\
&= \frac{\partial C_{T2}}{\partial \theta_i}(e^{j\omega} I - A_T)^{-1} B_{T1} + C_{T2} \frac{\partial (e^{j\omega} I - A_T)^{-1}}{\partial \theta_i} B_{T1} \\
&\quad + C_{T2}(e^{j\omega} I - A_T)^{-1} \frac{\partial B_{T1}}{\partial \theta_i} \\
&= \frac{\partial C_{T2}}{\partial \theta_i}(e^{j\omega} I - A_T)^{-1} B_{T1} - C_{T2}(e^{j\omega} I - A_T)^{-1} \frac{\partial A_T}{\partial \theta_i}(e^{j\omega} I - A_T)^{-1} B_{T1} \\
&\quad + C_{T2}(e^{j\omega} I - A_T)^{-1} \frac{\partial B_{T1}}{\partial \theta_i}
\end{aligned}$$

7. Bandwidth of closed loop system

$$\chi_{bw} = \min_{\omega} \left\{ \omega \,\middle|\, \mid M_{22}(e^{j\omega}) \mid^2 < 0.5 \right\}$$

Let $\chi_{bw} = \omega_o$, then

$$\begin{aligned}
\frac{\partial \chi_{bw}}{\partial \theta_i} &= \left(\frac{\partial \mid M_{22}(e^{j\omega_o}) \mid^2}{\partial \omega} \right)^{-1} \left(\frac{\partial \mid M_{22}(e^{j\omega_o}) \mid^2}{\partial \theta_i} \right) \\
&= \left(\frac{\partial \mid M_{22}(e^{j\omega_o}) \mid^2}{\partial \omega} \right)^{-1} \left(\mathrm{Re} \left(\frac{\partial\, M_{22}(e^{j\omega})}{\partial \theta_i} \bar{M}_{22}(e^{j\omega}) \right) d\omega \right)
\end{aligned}$$

where

$$\frac{\partial\, M_{22}(e^{j\omega})}{\partial \theta_i} = \frac{\partial C_{T2}}{\partial \theta_i}(e^{j\omega} I - A_T)^{-1} B_{T2}$$

$$
-C_{T2}(e^{j\omega}I - A_T)^{-1}\frac{\partial A}{\partial \theta_i}(e^{j\omega}I - A_T)^{-1}B_{T2}
$$
$$
+C_{T2}(e^{j\omega}I - A_T)^{-1}\frac{\partial B_{T2}}{\partial \theta_i}
$$

8. Stability radius of closed loop system

The stability radius is the maximum modulus of the closed loop poles, which is equal to the spectral radius ρ of the closed loop system matrix A_T.

$$
\chi_{sm} = \rho(A_t) = \max_i \mid \lambda_i(A_T) \mid
$$

where λ_i are the eigenvalues of closed loop system matrix A_T. Let i_o be the index of the maximum modulus eigenvalue, let the eigenvalue decomposition of A_T be given by $A_T = E\Lambda E^{-1}$, define $\bar{e}_i^T$ as the i-the row vector of E^{-1} and define e_i^T as the i-the column vector of E. Then $\lambda_i = \bar{e}_i^T A_T e_i$. We derive:

$$
\begin{aligned}
\frac{\partial \chi_{sm}}{\partial \theta_i} &= \mid \lambda_{i_o}(A_T) \mid = \\
&= \frac{1}{\mid \lambda_{i_o}(A_T) \mid}\mathrm{Re}\left(\bar{\lambda}_{i_o}(A_T)\frac{\partial \lambda_{i_o}(A_T)}{\partial \theta_i}\right)
\end{aligned}
$$

where

$$
\frac{\partial \lambda_{i_o}(A_T)}{\partial \theta_i} = \frac{\bar{e}_i^T\dfrac{\partial A_T}{\partial \theta_i}e_i}{\bar{e}_i^T e_i}
$$

9. Robustness margin of closed loop system

$$
\chi_{rob} = \|T(z)\|_\infty == \max_{|z|=1}\bar{\sigma}\Big(T(z)\Big) = u_1^* T(\theta,\eta,z_o)v_1
$$

where $z_o = \arg\max_{|z|=1}\bar{\sigma}\Big(T(z)\Big)$ and u_1 is the first column of U and v_1 is the first column of V for a singular value decomposition $T(\theta,\eta,z_o) = U\Sigma V^*$.

$$
T(\theta,\eta,z) = C_{\tilde{T}}(\theta,\eta)\Big(zI - A_{\tilde{T}}(\theta,\eta)\Big)^{-1}B_{\tilde{T}}(\theta,\eta) + D_{\tilde{T}}(\theta,\eta)
$$

Then

$$
\frac{\partial \|T\|_\infty}{\partial \theta_i} = \mathrm{Re}\left(u_1^*\frac{\partial T(\theta,\eta,z_o)}{\partial \theta_i}v_1\right)
$$

$$
\begin{aligned}
\frac{\partial T(\theta,\eta,z)}{\partial \theta_i} = \; & C_{\tilde{T}}(\theta,\eta)(zI - A_{\tilde{T}}(\theta,\eta))^{-1}\frac{\partial A_{\tilde{T}}(\theta,\eta)}{\partial \theta_i}(zI - A_{\tilde{T}}(\theta,\eta))^{-1}B_{\tilde{T}}(\theta,\eta) + \\
& C_{\tilde{T}}(\theta,\eta)(zI - A_{\tilde{T}}(\theta,\eta))^{-1}\frac{\partial B_{\tilde{T}}(\theta,\eta)}{\partial \theta_i} + \\
& \frac{\partial C_{\tilde{T}}(\theta,\eta)}{\partial \theta_i}(zI - A_{\tilde{T}}(\theta,\eta))^{-1}B_{\tilde{T}}(\theta,\eta) + + \frac{\partial D_{\tilde{T}}(\theta,\eta)}{\partial \theta_i}
\end{aligned}
$$

where

$$
\begin{aligned}
\frac{\partial A_{\tilde{T}}(\theta,\eta)}{\partial\theta_i} &= \begin{bmatrix} A + B_3\dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i}C_1 & -B_3\dfrac{\partial F(\theta,\eta)}{\partial\theta_i} - B_3\dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i}C_1 \\ B_3\dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i}C_1 & A - B_3\dfrac{\partial F(\theta,\eta)}{\partial\theta_i} - B_3\dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i}C_1 \end{bmatrix} \\
\frac{\partial B_{\tilde{T}}(\theta,\eta)}{\partial\theta_i} &= 0 \\
\frac{\partial C_{\tilde{T}}(\theta,\eta)}{\partial\theta_i} &= \begin{bmatrix} D_{12}\dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i}C_1 & -D_{12}\dfrac{\partial F(\theta,\eta)}{\partial\theta_i} - D_{12}\dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i}C_1 \end{bmatrix} \\
\frac{\partial D_{\tilde{T}}(\theta,\eta)}{\partial\theta_i} &= 0
\end{aligned}
$$

Derivatives of system matrices

$$
\begin{aligned}
\frac{\partial A_T(\theta,\eta)}{\partial\theta_i} &= \begin{bmatrix} A + B_3\dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i}C_{T1} & -B_3\dfrac{\partial F(\theta,\eta)}{\partial\theta_i} - B_3\dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i}C_{T1} \\ B_3\dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i}C_{T1} & A - B_3\dfrac{\partial F(\theta,\eta)}{\partial\theta_i} - B_3\dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i}C_{T1} \end{bmatrix} \\
\frac{\partial B_{T1}(\theta,\eta)}{\partial\theta_i} &= \begin{bmatrix} B_3\dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i} \\ 0 \end{bmatrix} \\
\frac{\partial B_{T2}(\theta,\eta)}{\partial\theta_i} &= \begin{bmatrix} B_3\dfrac{\partial D_w(\theta,\eta)}{\partial\theta_i} \\ 0 \end{bmatrix} \\
\frac{\partial C_{T1}(\theta,\eta)}{\partial\theta_i} &= \begin{bmatrix} -\dfrac{\partial F(\theta,\eta)}{\partial\theta_i} & -\dfrac{\partial F(\theta,\eta)}{\partial\theta_i} - \dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i}C_{T1} \end{bmatrix} \\
\frac{\partial C_{T2}(\theta,\eta)}{\partial\theta_i} &= \begin{bmatrix} 0 & 0 \end{bmatrix} \\
\frac{\partial D_{T1}(\theta,\eta)}{\partial\theta_i} &= \begin{bmatrix} \dfrac{\partial D_e(\theta,\eta)}{\partial\theta_i} \end{bmatrix} \\
\frac{\partial D_{T2}(\theta,\eta)}{\partial\theta_i} &= \begin{bmatrix} \dfrac{\partial D_w(\theta,\eta)}{\partial\theta_i} \end{bmatrix}
\end{aligned}
$$

With

$$
\begin{aligned}
M(\theta,\eta) &= E_v\,(\,\tilde{D}_3(\theta,\eta)^T\bar{\Gamma}(\eta)\tilde{D}_3\,) \\
F(\theta,\eta) &= M^{-1}\tilde{D}_3(\theta,\eta)^T\bar{\Gamma}(\eta)\tilde{C}_{T2} \\
D_e(\theta,\eta) &= -M^{-1}\tilde{D}_3(\theta,\eta)^T\bar{\Gamma}(\eta)\tilde{D}_1 \\
D_w(\theta,\eta) &= -M^{-1}\tilde{D}_3(\theta,\eta)^T\bar{\Gamma}(\eta)\tilde{D}_2
\end{aligned}
$$

we derive:

$$
\begin{aligned}
\frac{\partial M(\theta)}{\partial \theta_i} &= E_v\big(\,\frac{\partial \tilde{D}_3^T}{\partial \theta_i}\bar{\Gamma}\tilde{D}_3\,\big) + E_v\big(\,\tilde{D}_3^T\bar{\Gamma}\frac{\partial \tilde{D}_3}{\partial \theta_i}\,\big) \\
\frac{\partial F(\theta)}{\partial \theta_i} &= -M^{-1}\frac{\partial M}{\partial \theta_i}M^{-1}\tilde{D}_3^T\bar{\Gamma}\tilde{C}_{T2} + M\frac{\partial \tilde{D}_3^T}{\partial \theta_i}\bar{\Gamma}\tilde{C}_{T2} + M\tilde{D}_3^T\bar{\Gamma}\frac{\partial \tilde{C}_{T2}}{\partial \theta_i} \\
\frac{\partial D_e(\theta)}{\partial \theta_i} &= M^{-1}\frac{\partial M}{\partial \theta_i}M^{-1}\tilde{D}_3^T\bar{\Gamma}\tilde{D}_1 - M\frac{\partial \tilde{D}_3^T}{\partial \theta_i}\bar{\Gamma}\tilde{D}_1 - M\tilde{D}_3^T\bar{\Gamma}\frac{\partial \tilde{D}_1}{\partial \theta_i}
\end{aligned}
$$

The matrices $\tilde{C}_{T2}$, $\tilde{D}_1$ and $\tilde{D}_3$ are directly depending on the parameters θ for the GPC or MPC case (as shown in section 4.1).

Appendix E:

Standard Predictive Control Toolbox

T.J.J. van den Boom

Manual MATLAB Toolbox

Contents

Introduction

This toolbox is a collection of a few MATLAB functions for the design and analysis of predictive controllers. This toolbox has been developed by Ton van den Boom at the Delft Center for Systems and Control (DCSC), Delft University of Technology, the Netherlands. The Standard Predictive Control toolbox is an independent tool, which only requires the control system toolbox and the signal processing toolbox.

models in SYSTEM format

The notation in the SPC toolbox is kept as close as possible to the one in the lecture notes. To describe systems (IO model, IIO model, standard model, prediction model) we use the SYSTEM format, where all the system matrices are stacked in a matrix `G` and the dimensions are given in a vector `dim`.

IO model in SYSTEM format:

Consider the IO-system

$$\begin{aligned} x_o(k+1) &= A_o x_o(k) + K_o e_o(k) + L_o d_o(k) + B_o u(k) \\ y(k) &= C_o x_o(k) + D_H e_o(k) + D_F d_o(k) \end{aligned}$$

for $A_o \in \mathbb{R}^{n_a \times n_a}$, $K_o \in \mathbb{R}^{n_a \times n_k}$, $L_o \in \mathbb{R}^{n_a \times n_l}$, $B_o \in \mathbb{R}^{n_a \times n_b}$ and $C_o \in \mathbb{R}^{n_c \times n_a}$.
In SYSTEM format this will be given as:

$$\mathtt{G} = \begin{bmatrix} A_o & K_o & L_o & B_o \\ C_o & D_H & D_F & 0 \end{bmatrix} \qquad \mathtt{dim} = \begin{bmatrix} n_a & n_k & n_l & n_b & n_c & 0 & 0 & 0 \end{bmatrix}$$

IIO model in SYSTEM format:

Consider the IIO-system

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + K_i e_i(k) + L_i d_i(k) + B_i \Delta u(k) \\ y(k) &= C_i x_i(k) + D_H e_i(k) + D_F d_i(k) \end{aligned}$$

for $A_i \in \mathbb{R}^{n_a \times n_a}$, $K_i \in \mathbb{R}^{n_a \times n_k}$, $L_i \in \mathbb{R}^{n_a \times n_l}$, $B_i \in \mathbb{R}^{n_a \times n_b}$ and $C_i \in \mathbb{R}^{n_c \times n_a}$.
In SYSTEM format this will be given as:

$$\mathtt{G} = \begin{bmatrix} A_i & K_i & L_i & B_i \\ C_i & D_H & D_F & 0 \end{bmatrix} \qquad \mathtt{dim} = \begin{bmatrix} n_a & n_k & n_l & n_b & n_c & 0 & 0 & 0 \end{bmatrix}$$

Standard model in SYSTEM format:

Consider the standard model of the SPCP:

$$\begin{aligned}
x(k+1) &= Ax(k) + B_1e(k) + B_2w(k) + B_3v(k) \\
y(k) &= C_1x(k) + D_{11}e(k) + D_{12}w(k) \\
z(k) &= C_2x(k) + D_{21}e(k) + D_{22}w(k) + D_{23}v(k) \\
\phi(k) &= C_3x(k) + D_{31}e(k) + D_{32}w(k) + D_{33}v(k) \\
\psi(k) &= C_4x(k) + D_{41}e(k) + D_{42}w(k) + D_{43}v(k)
\end{aligned}$$

for $A \in \mathrm{IR}^{n_a\times n_a}$, $B_i \in \mathrm{IR}^{n_a\times n_{bi}}$, $i = 1,2,3$, $C_j \in \mathrm{IR}^{n_{cj}\times n_a}$, $j = 1,2,3,4$ and In SYSTEM format this will be given as:

$$\mathtt{G} = \begin{bmatrix} A & B_1 & B_2 & B_3 \\ C_1 & D_{11} & D_{12} & 0 \\ C_2 & D_{21} & D_{22} & D_{23} \\ C_3 & D_{31} & D_{32} & D_{33} \\ C_4 & D_{41} & D_{42} & D_{43} \end{bmatrix} \qquad \mathtt{dim} = \begin{bmatrix} n_a & n_{b1} & n_{b2} & n_{b3} & n_{c1} & n_{c2} & n_{c3} & n_{c4} \end{bmatrix}$$

Prediction model in SYSTEM format:

Consider the prediction model of the SPCP:

$$\begin{aligned}
x(k+1) &= A\,x(k) + \tilde{B}_1\,e(k) + \tilde{B}_2\tilde{w}(k) + \tilde{B}_3\tilde{v}(k) \\
\tilde{y}(k) &= \tilde{C}_1x(k) + \tilde{D}_{11}e(k) + \tilde{D}_{12}\tilde{w}(k) + \tilde{D}_{13}\tilde{v}(k) \\
\tilde{z}(k) &= \tilde{C}_2x(k) + \tilde{D}_{21}e(k) + \tilde{D}_{22}\tilde{w}(k) + \tilde{D}_{23}\tilde{v}(k) \\
\tilde{\phi}(k) &= \tilde{C}_3x(k) + \tilde{D}_{31}e(k) + \tilde{D}_{32}\tilde{w}(k) + \tilde{D}_{33}\tilde{v}(k) \\
\tilde{\psi}(k) &= \tilde{C}_4x(k) + \tilde{D}_{41}e(k) + \tilde{D}_{42}\tilde{w}(k) + \tilde{D}_{43}\tilde{v}(k)
\end{aligned}$$

where $A \in \mathrm{IR}^{n_a\times n_a}$, $\tilde{B}_i \in \mathrm{IR}^{n_a\times n_{bi}}$, $\tilde{C}_j \in \mathrm{IR}^{n_{cj}\times n_a}$, and $\tilde{D}_{ji} \in \mathrm{IR}^{n_{cj}\times n_{bi}}$, $i = 1,2,3$, $j = 2,3,4$. the SYSTEM format is given as:

$$\mathtt{G} = \begin{bmatrix} A & \tilde{B}_1 & \tilde{B}_2 & \tilde{B}_3 \\ \tilde{C}_1 & \tilde{D}_{11} & \tilde{D}_{12} & \tilde{D}_{13} \\ \tilde{C}_2 & \tilde{D}_{21} & \tilde{D}_{22} & \tilde{D}_{23} \\ \tilde{C}_3 & \tilde{D}_{31} & \tilde{D}_{32} & \tilde{D}_{33} \\ \tilde{C}_4 & \tilde{D}_{41} & \tilde{D}_{42} & \tilde{D}_{43} \end{bmatrix} \qquad \mathtt{dim} = \begin{bmatrix} n_a & n_{b1} & n_{b2} & n_{b3} & * & n_{c2} & n_{c3} & n_{c4} \end{bmatrix}$$

Reference

Common predictive control problems	
gpc	solves the generalized predictive control problem
gpc_ss	solves the state space generalized predictive control problem
lqpc	solves the linear quadratic predictive control problem
distcol	Solves the model predictive control problem for a distillation column

Construction of process model	
tf2syst	transforms a polynomial into a state space model in SYSTEM format
imp2syst	transforms a impulse response model into a state space model in SYSTEM format
ss2syst	transforms a state space model into the SYSTEM format
syst2ss	transforms SYSTEM format into a state space model
io2iio	transforms an IO system into an IIO system

Formulation of standard predictive control problem	
gpc2spc	transforms GPC problem into an SPC problem
lqpc2spc	transforms LQPC problem into an SPC problem
add_du	function to add increment input constraint to SPC problem
add_u	function to add input constraint to SPC problem
add_v	function to add input constraint to SPC problem
add_y	function to add output constraint to SPC problem
add_x	function to add state constraint to SPC problem
add_nc	function to add a control horizon to SPC problem
add_end	function to add a state end-point constraint to SPC problem

Standard predictive control problem	
`external`	function makes the external signal $\tilde{w}$ from d and r
`dgamma`	function makes the selection matrix $\bar{\Gamma}$
`pred`	function makes a prediction model
`contr`	function computes controller matrices for predictive controller
`contr`	function computes controller matrices for predictive controller
`contrinf`	function computes controller matrices for the infinite horizon predictive controller
`lticll`	function computes the state space matrices of the LTI closed loop
`lticon`	function computes the state space matrices of the LTI optimal predictive controller
`simul`	function makes a closed loop simulation with predictive controller
`rhc`	function makes a closed loop simulation in receding horizon mode

Demonstrations of tuning GPC and LQPC	
`demogpc`	Tuning GPC controller
`demolqpc`	Tuning LQPC controller

gpc

Purpose

Solves the Generalized Predictive Control (GPC) problem

Synopsis

```
[y,du]=gpc(ai,bi,ci,fi,P,lambda,Nm,N,Nc,lensim,...
                              r,di,ei,dumax,apr,rhc);
```

Description

The function solves the GPC method (Clarke *et al.*,[21],[22]) for a controlled autoregressive integrated moving average (CARIMA) model, given by

$$a_i(q)\,y(k) = b_i(q)\Delta u(k) + c_i(q)\,e_i(k) + f_i(q)\,d_i(k)$$

where the increment input signal is given by $\Delta u(k) = u(k) - u(k-1)$.
A performance index, based on control and output signals, is minimized:

$$\min_{\Delta u(k)} J(\Delta u, k) = \min_{\Delta u(k)} \sum_{j=N_m}^{N} |\hat{y}_p(k+j|k) - r(k+j)|^2 + \lambda^2 \sum_{j=1}^{N_c} |\Delta u(k+j-1)|^2$$

where

$y_p(k)$	$= P(q)y(k)$ is the weighted process output signal
$r(k)$	is the reference trajectory
$y(k)$	is the process output signal
$d_i(k)$	is the known disturbance increment signal
$e_i(k)$	is the zero-mean white noise
$\Delta u(k)$	is the process control increment signal
N_m	is the minimum cost- horizon
N	is the prediction horizon
N_c	is the control horizon
λ	is the weighting on the control signal
$P(q)$	$= 1+p_1 q^{-1}+\ldots+p_{n_p} q^{-n_p}$ is a polynomial with desired closed-loop poles

The signal $\hat{y}_p(k+j|k)$ is the prediction of $y_p(k+j)$, based on knowledge up to time k. The input signal $u(k+j)$ is forced to become constant for $j \geq N_c$, so

$$\Delta u(k+j) = 0 \qquad \text{for} \qquad j \geq N_c$$

Further the increment input signal is assumed to be bounded:

$$|\,\Delta u(k+j)\,| \leq \Delta u_{max} \qquad \text{for} \qquad 0 \leq j < N_c$$

The parameter λ determines the trade-off between tracking accuracy (first term) and control effort (second term). The polynomial $P(q)$ can be chosen by the designer and broaden the class of control objectives. n_p of the closed-loop poles will be placed at the location of the roots of polynomial $P(q)$.

The function `gpc` returns the output signal `y` and increment input signal `du`. The input and output variables are the following:

`y`	the output signal $y(k)$.
`du`	the increment input signal $\Delta u(k)$.
`ai,bi,ci,fi`	polynomials of CARIMA model.
`P`	is the polynomial $P(q)$.
`lambda`	is the trade-off parameter λ.
`Nm,N,Nc`	are the summation parameters.
`r`	is the column vector with reference signal $r(k)$.
`di`	is the column vector disturbance increment signal $d_i(k)$.
`ei`	is the column vector with ZMWN noise signal $e_i(k)$.
`dumax`	is the maximum bound Δu_{max} on the increment input signal.
`lensim`	is length of simulation interval.
`apr`	vector with flags for a priori knowledge on external signal.
`rhc`	flag for receding horizon mode.

Setting `dumax=0` means no increment input constraint will be added.

The vector `apr` (with length 2) determines if the external signals $d_i(k+j)$ and $r(k+j)$ are a priori known or not. The disturbance $d_i(k+j)$ is a priori known for `apr(1)=1` and unknown for `apr(1)=0`. The reference $r(k+j)$ is a priori known for `apr(2)=1` and unknown for `apr(2)=0`.

For `rhc=0` the simulation is done in closed loop mode. For `rhc=1` the simulation is done in receding horizon mode, which means that the simulation is paused every sample and can be continued by pressing the return-key. Past signals and prediction of future signals are given within the prediction range.

Example 40

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Example for gpc
```

```
ai = conv([1 -1.787 0.798],[1 -1]);              % polynomial ai(q)
bi = 0.029*[0 1 0.928 0];                        % polynomial bi(q)
ci = [1 -1.787 0.798 0];                         % polynomial ci(q)
fi = [0  1 -1.787 0.798];                        % polynomial fi(q)

P = 1;              % No pole-placement
lambda=0;           % increment input is not weighted
Nm = 8;             % Mimimum cost horizon
Nc = 2;             % Control horizon
N  = 25;            % Prediction horizon
lensim = 60;        % length simulation interval
r  = [zeros(1,10) ones(1,100)];      % step on k=11
di = [zeros(1,20) 0.5 zeros(1,89)];  % step on k=21
ei = [zeros(1,30) -0.1*randn(1,31)]; % noise starting from k=31
dumax = 0;          % dumax=0 means that no constraint is added !!
apr = [0 0];        % no apriori knowledge about w(k)

[y,du] = gpc(ai,bi,ci,fi,P,lambda,Nm,N,Nc,lensim,r,di,ei,dumax,apr);

u  = cumsum(du);       % compute   u(k)   from    du(k)
do = cumsum(di(1:61)); % compute   do(k)   from   di(k)
eo = cumsum(ei(1:61)); % compute   eo(k)   from   ei(k)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

t=0:60;
subplot(211);
stairs(t,[y' r(:,1:61)' do' eo']);
title('green: r , blue: y , red: do , cyan: eo');
grid;
subplot(212);
stairs(t,[du' u']);
title('blue: du , green: u');
grid;
disp('   ');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

See Also

tf2io, io2iio, gpc2spc, external, pred, add_nc, add_du, add_y, contr, simul

gpc_ss

Purpose

Solves the Generalized Predictive Control (GPC) problem for a state space IIO model

Synopsis

```
[y,du]=gpc_ss(Ai,Ki,Li,Bi,Ci,Ap,Bp,Cp,Dp,Wy,Wu,Nm,N,Nc,lensim,...
                              r,di,ei,dumax,apr,rhc);
```

Description

The function solves the GPC method (Clarke *et al.*, [21],[22]) for a state space IIO model, given by

$$\begin{aligned} x(k+1) &= A_i x(k) + K_i e_i(k) + L_i d_i(k) + B_i \Delta u(k) \\ y(k) &= C_i x(k) + e_i(k) \end{aligned}$$

where the increment input signal is given by $\Delta u(k) = u(k) - u(k-1)$.
A performance index, based on control and output signals, is minimized:

$$\min_{\Delta u(k)} J(\Delta u, k) = \min_{\Delta u(k)} \sum_{j=N_m}^{N} |W_y\, \hat{y}_p(k+j|k) - r(k+j)|^2 + \sum_{j=1}^{N_c} |W_u\, \Delta u(k+j-1)|^2$$

where $|\alpha(k)|$ stands for the euclidean norm of the vector $\alpha(k)$, and the variables are given by:

$y_p(k)$	$= P(q)y(k)$ is the weighted process output signal
$r(k)$	is the reference trajectory
$y(k)$	is the process output signal
$d_i(k)$	is the known disturbance increment signal
$e_i(k)$	is the zero-mean white noise
$\Delta u(k)$	is the process control increment signal
N_m	is the minimum cost- horizon
N	is the prediction horizon
N_c	is the control horizon
W_y	is the weighting matrix on the tracking error
W_u	is the weighting matrix on the control signal
$P(q)$	is the reference system

The reference system $P(q)$ is given by the state space realization:

$$\begin{aligned} x_p(k+1) &= A_p x(k) + B_p y(k) \\ \psi(k) &= C_p x(k) + D_p y(k) \end{aligned}$$

The signal $\hat{y}_p(k+j|k)$ is the prediction of $y_p(k+j)$, based on knowledge up to time k. The input signal $u(k+j)$ is forced to become constant for $j \geq N_c$, so

$$\Delta u(k+j) = 0 \qquad \text{for} \qquad j \geq N_c$$

Further the increment input signal is assumed to be bounded:

$$|\ \Delta u(k+j)\ | \leq \Delta u_{max} \qquad \text{for} \qquad 0 \leq j < N_c$$

For MIMO systems Δu_{max} can be choosen as a scalar value or a vector value (with the same dimension as vector $\Delta u(k)$). In the case of a scalar the bound Δu_{max} holds for each element of $\Delta u(k)$. In the case of a vector Δu_{max}, the bound is taken elementwise.

The weighting matrices W_y and W_u determines the trade-off between tracking accuracy (first term) and control effort (second term). The reference system $P(q)$ can be chosen by the designer and broaden the class of control objectives.

The function `gpc_ss` returns the output signal `y` and increment input signal `du`. The input and output variables are the following:

`y`	the output signal $y(k)$.
`du`	the increment input signal $\Delta u(k)$.
`Ai,Ki,Li,Bi,Ci`	system matrices of IIO model.
`Ap,Bp,Cp,Dp`	system matrices of reference system $P(q)$.
`Wy`	weighting matrix on tracking error.
`Wu`	weighting matrix on input increment signal.
`Nm,N,Nc`	are the summation parameters.
`r`	is the column vector with reference signal $r(k)$.
`di`	is the column vector disturbance increment signal $d_i(k)$.
`ei`	is the column vector with ZMWN noise signal $e_i(k)$.
`dumax`	is the vector with bound Δu_{max}.
`lensim`	is length of simulation interval.
`apr`	vector with flags for a priori knowledge on external signal.
`rhc`	flag for receding horizon mode.

Setting `dumax=0` means no increment input constraint will be added.

The vector apr (with length 2) determines if the external signals $d_i(k+j)$ and $r(k+j)$ are a priori known or not. The disturbance $d_i(k+j)$ is a priori known for apr(1)=1 and unknown for apr(1)=0. The reference $r(k+j)$ is a priori known for apr(2)=1 and unknown for apr(2)=0.

For rhc=0 the simulation is done in closed loop mode. For rhc=1 the simulation is done in receding horizon mode, which means that the simulation is paused every sample and can be continued by pressing the return-key. Past signals and prediction of future signals are given within the prediction range.

See Also

gpc, tf2io, io2iio, gpc2spc, external, pred, add_nc, add_du, add_y, contr, simul

lqpc

Purpose

Solves the Linear Quadratic Predictive Control (LQPC) problem

Synopsis

```
[y,u,x]=lqpc(Ao,Ko,Lo,Bo,Co,Q,R,Nm,N,Nc,lensim,xo,do,eo, ...
                                umax,apr,rhc);
```

Description

The function solves the Linear Quadratic Predictive Control (LQPC) method (García *et al.*, [50]) for a state space model, given by

$$\begin{aligned} x(k+1) &= A_o x(k) + K_o e_o(k) + L_o d_o(k) + B_o u(k) \\ y(k) &= C_o x(k) + e_o(k) \end{aligned}$$

A performance index, based on state and input signals, is minimized:

$$\min_{u(k)} J(u,k) = \min_{u(k)} \sum_{j=N_m}^{N} \hat{x}(k+j)^T Q \hat{x}(k+j) + \sum_{j=1}^{N} u(k+j-1)^T R u(k+j-1)$$

where

$x(k)$	is the state signal vector
$u(k)$	is the process control signal
$y(k)$	is the process output signal
$d_o(k)$	is the known disturbance signal
$e_o(k)$	is the zero-mean white noise
N_m	is the minimum cost- horizon
N	is the prediction horizon
N_c	is the control horizon
Q	is the state weighting matrix
R	is the control weighting matrix

The signal $\hat{x}(k+j|k)$ is the prediction of state $x(k+j)$, based on knowledge up to time k. The input signal $u(k+j)$ is forced to become constant for $j \geq N_c$. Further the input signal is assumed to be bounded:

$$|\, u(k+j) \,| \leq u_{max} \qquad \text{for} \qquad 0 \leq j < N_c$$

The matrices Q and R determines the trade-off between tracking accuracy (first term) and control effort (second term). For MIMO systems the matrices can scale the states and inputs.

The function `lqpc` returns the output signal `y`, input signal `u` and the state signal `x`. The input and output variables are the following:

`y`	the output signal $y(k)$.
`u`	the input signal $u(k)$.
`x`	the state signal $x(k)$.
`Ao,Ko,Lo,Bo,Co`	system matrices of state space IO model.
`Q,R`	the weighting matrices Q and R
`Nm,N,Nc`	the summation parameters.
`xo`	the column vector with initial state $xo = x(0)$.
`do`	the disturbance signal $d_o(k)$.
`eo`	the ZMWN noise signal $e_o(k)$.
`umax`	the maximum bound u_{max} on the input signal.
`lensim`	length of simulation interval.
`apr`	flag for a priori knowledge on disturbance signal.
`rhc`	flag for receding horizon mode.

The signal `eo` must have as many rows as there are noise inputs. The signal `do` must have as many rows as there are disturbance inputs. Each column of `eo` and `do` corresponds to a new time point.

The signal `y` will have as many rows as there are outputs. The signal `u` will have as many rows as there are inputs. The signal `x` must have as many rows as there are states. Each column of `y`, `u` and `x` corresponds to a new time point.

Setting `umax=0` means no input constraint will be added.

The (scalar) variable `apr` determines if the disturbance signal $d_i(k+j)$ is a priori known or not. The signal $d_i(k+j)$ is a priori known for `apr=1` and unknown for `apr=0`.

For `rhc=0` the simulation is done in closed loop mode. For `rhc=1` the simulation is done in receding horizon mode, which means that the simulation is paused every sample and can be continued by pressing the return-key. Past signals and prediction of future signals are given within the prediction range.

See Also

`lqpc2spc, pred, add_nc, add_u, add_y, contr, simul`

mpcdist

Purpose

Solves the model predictive control problem for a distillation column

Synopsis

```
[y,du,x]=mpcdist(Ai,Ki,Li,Bi,Ci,Wy,Wu,Nm,N,Nc,lensim,y0,dumax,sim);
```

Description

The function solves the Model Predictive Control problem for a distillation column, with A linearized model in working point, given by

$$\begin{aligned} x(k+1) &= A_i x(k) + K_i e_i(k) + L_i d_i(k) + B_i \Delta u(k) \\ y(k) &= C_i x(k) + e_i(k) \end{aligned}$$

A performance index, based on state and input signals, is minimized:

$$\min_{u(k)} J(u,k) = \min_{u(k)} \sum_{j=N_m}^{N} \hat{y}(k+j)^T W_y \hat{y}(k+j) + \sum_{j=1}^{N} \Delta u(k+j-1)^T W_u \Delta u(k+j-1)$$

where

$x(k)$	is the model state signal vector
$y(k)$	is the process output signal
$d_i(k)$	is the known disturbance increment signal
$e_i(k)$	is the zero-mean white noise
$\Delta u(k)$	is the process control increment signal
N_m	is the minimum cost- horizon
N	is the prediction horizon
N_c	is the control horizon
W_y	is the weighting matrix on the tracking error
W_u	is the weighting matrix on the control signal

The signal $\hat{y}(k+j|k)$ is the prediction of state $y(k+j)$, based on knowledge up to time k. The input increment signal $\Delta u(k+j)$ is forced to become constant for $j \geq N_c$. Further the input signal is assumed to be bounded:

$$|\, \Delta u(k+j) \,| \leq \Delta u_{max} \qquad \text{for} \qquad 0 \leq j < N_c$$

The matrices W_y and W_u determines the trade-off between tracking accuracy (first term) and control effort (second term). For MIMO systems the matrices can scale the states and inputs.

The matrices Ai, Ki, Li, Bi, Ci of the linearized model are obtained by running the m-file `f2ss`.

There are three modes for the simulation:
For `sim = 'lin'`, the linearized model will be used in the simulation. This means the simulation is based on the same model as the one used in the controller design.
For `sim = 'nl1'`, a nonlinear process model is used in the simulation. This nonlinear process model has been shifted to the first working point, for which the linearized model was derived.
For `sim = 'nl2'`, a nonlinear process model is used in the simulation. This nonlinear process model has been shifted to the second working point, which is different from the one for which the linearized model was derived.
At time $k \leq 1$ the process is assumed to be shifted from the working point $y(1) = y0$, but in steady-state for constant input $\Delta u(k) = 0$ for $k \leq 0$.

The function `lqpc` returns the output signal `y`, input signal `u` and the state signal `x`. The input and output variables are the following:

`y`	the output signal $y(k)$.
`u`	the input signal $u(k)$.
`x`	the state signal $x(k)$.
`Ao,Ko,Lo,Bo,Co`	system matrices of state space IO model.
`Wy`	weighting matrix on tracking error.
`Wu`	weighting matrix on input increment signal.
`Nm,N,Nc`	the summation parameters.
`yo`	the column vector with initial output $yo = y(0)$.
`dumax`	is the vector with bound Δu_{max}.
`lensim`	length of simulation interval.
`sim`	flag to determine simulation mode.

Setting `dumax=0` means no input constraint will be added.

tf2syst

Purpose

Transforms a polynomial into a state space model in SYSTEM format

Synopsis

```
[G,dim]=tf2syst(a,b,c,f);
```

Description

The function transforms a polynominal model into a state space model in SYSTEM format (For SISO system only). The input system is:

$$a(q)y(k) = b(q)v(k) + c(q)e(k) + f(q)d(k)$$

The state space equations become

$$\begin{aligned} x(k+1) &= Ax(k) + B_1 e(k) + B_2 d(k) + B_3 v(k) \\ y(k) &= C_1 x(k) + e(k) \end{aligned}$$

$$A \in \mathbb{R}^{na \times na} \quad , \quad B_1 \in \mathbb{R}^{na \times nb1} \quad , \quad B_2 \in \mathbb{R}^{na \times nb2}$$

$$B_3 \in \mathbb{R}^{na \times nb3} \quad , \quad C_1 \in \mathbb{R}^{nc1 \times na}$$

The state space representation is given in a `SYSTEM` format, so

$$\mathtt{G} = \left[\begin{array}{c|ccc} A & B_1 & B_2 & B_3 \\ \hline C_1 & I & 0 & 0 \end{array} \right]$$

with dimension vector

$$\mathtt{dim} = \left[\begin{array}{cccccccc} na & nb1 & nb2 & nb3 & nc1 & 0 & 0 & 0 \end{array} \right]$$

The function `tf2syst` returns the state space model in `Go` and its dimension `dimo`, given in `SYSTEM` format. The input and output variables are the following:

`G`	The state space model in `SYSTEM` format.
`dim`	the dimension of the system.
`a,b,c,f`	polynomials of CARIMA model.

See Also

`io2iio, imp2syst, ss2syst`

imp2syst

Purpose

Transforms an impulse reponse model into a state space model in SYSTEM format

Synopsis

```
[G,dim]=imp2syst(g,f,nimp);
[G,dim]=imp2syst(g,f);
```

Description

The function transforms an impulse response model into a state space model in SYSTEM format (For SISO system only) The input system is:

$$y(k) = \sum_{i=1}^{n_g} g(i)u(k-i) + \sum_{i=1}^{n_f} f(i)d(k-i) + e(k)$$

where the impulse response parameters $g(i)$ and $f(i)$ are given in the vectors

$$\begin{aligned} \texttt{g} &= \begin{bmatrix} 0 & g(1) & g(2) & \ldots & g(n_g) \end{bmatrix} \\ \texttt{f} &= \begin{bmatrix} 0 & f(1) & f(2) & \ldots & f(n_f) \end{bmatrix} \end{aligned}$$

The state space equations become

$$\begin{aligned} x(k+1) &= Ax(k) + B_1 e(k) + B_2 d(k) + B_3 v(k) \\ y(k) &= C_1 x(k) + e(k) \end{aligned}$$

$$A \in \mathbb{R}^{nimp \times nimp} \quad , \quad B_1 \in \mathbb{R}^{nimp \times nb1} \quad , \quad B_2 \in \mathbb{R}^{nimp \times nb2}$$

$$B_3 \in \mathbb{R}^{nimp \times nb3} \quad , \quad C_1 \in \mathbb{R}^{nc1 \times nimp}$$

The state space representation is given in a `SYSTEM` format, so

$$\texttt{G} = \left[\begin{array}{c|ccc} A & B_1 & B_2 & B_3 \\ \hline C_1 & I & 0 & 0 \end{array} \right]$$

with dimension vector

$$\texttt{dim} = \begin{bmatrix} nimp & nb1 & nb2 & nb3 & nc1 & 0 & 0 & 0 \end{bmatrix}$$

The function `imp2syst` returns the state space model in `Go` and its dimension dimo, given in SYSTEM format. The input and output variables are the following:

G	The state space model in SYSTEM format.
dim	the dimension of the system.
g	vectors with impulse response parameters of process model.
f	vectors with impulse response parameters of disturbance model.
nimp	length of impulse response in model G

If nimp is not given, it will be set to nimp = max(length(g),length(h))-1.

See Also

io2iio, tf2syst, ss2syst

ss2syst

Purpose

Transforms a state space model into the SYSTEM format

Synopsis

```
[G,dim]=ss2syst(A,B1,B2,B3,C1,D11,D12,D13);
[G,dim]=ss2syst(A,B1,B2,B3,C1,D11,D12,D13,C2,D21,D22,D23);
[G,dim]=ss2syst(A,B1,B2,B3,C1,D11,D12,D13,C2,D21,D22,D23,...
                                  C3,D31,D32,D33,C4,D41,D42,D43);
```

Description

The function transforms a state space model into the SYSTEM format The state space equations are

$$\begin{aligned} x(k+1) &= Ax(k) + B_1 e(k) + B_2 w(k) + B_3 v(k) \\ y(k) &= C_1 x(k) + D_{11} e(k) + D_{12} w(k) + D_{13} v(k) \\ z(k) &= C_2 x(k) + D_{21} e(k) + D_{22} w(k) + D_{23} v(k) \\ \phi(k) &= C_3 x(k) + D_{31} e(k) + D_{32} w(k) + D_{33} v(k) \\ \psi(k) &= C_4 x(k) + D_{41} e(k) + D_{42} w(k) + D_{43} v(k) \end{aligned}$$

$$A \in \mathbb{R}^{na \times na} \quad , \quad B_1 \in \mathbb{R}^{na \times nb1} \quad , \quad B_2 \in \mathbb{R}^{na \times nb2} \quad , \quad B_3 \in \mathbb{R}^{na \times nb3}$$

$$C_1 \in \mathbb{R}^{nc1 \times na} \quad , \quad C_2 \in \mathbb{R}^{nc2 \times na} \quad , \quad C_3 \in \mathbb{R}^{nc3 \times na}$$

The output system is given in the `SYSTEM` format, so

$$\mathtt{G} = \left[\begin{array}{c|ccc} A & B_1 & B_2 & B_3 \\ \hline C_1 & D_{11} & D_{12} & D_{13} \\ C_2 & D_{21} & D_{22} & D_{23} \\ C_3 & D_{31} & D_{32} & D_{33} \\ C_4 & D_{41} & D_{42} & D_{43} \end{array} \right]$$

with dimension vector

$$\mathtt{dim} = \left[\begin{array}{cccccccc} na & nb1 & nb2 & nb3 & nc1 & nc2 & nc3 & nc4 \end{array} \right]$$

The function `ss2syst` returns the state space model in `G` and its dimension `dim`, given in `SYSTEM` format. The input and output variables are the following:

`G`	The state space model in `SYSTEM` format.
`dim`	the dimension of the system.
`A,B1,...,D43`	system matrices of state space model.

See Also

`io2iio, imp2syst, tf2syst, syst2ss`

syst2ss

Purpose

Transforms SYSTEM format into a state space model

Synopsis

```
[A,B1,B2,B3,C1,D11,D12,D13]=syst2ss(sys,dim);
[A,B1,B2,B3,C1,D11,D12,D13,C2,D21,D22,D23]=syst2ss(sys,dim);
[A,B1,B2,B3,C1,D11,D12,D13,C2,D21,D22,D23,C3,D31,D32,D33,...
                              C4,D41,D42,D43]=syst2ss(sys,dim);
```

Description

The function transforms a model in SYSTEM format into its state space representation.

Consider a system in the SYSTEM format, so

$$\mathtt{sys} = \left[\begin{array}{c|ccc} A & B_1 & B_2 & B_3 \\ \hline C_1 & D_{11} & D_{12} & D_{13} \end{array}\right]$$

with dimension vector `dim`. Using

```
[A,B1,B2,B3,C1,D11,D12,D13]=syst2ss(sys,dim);
```

the state space equations become:

$$\begin{aligned} x(k+1) &= Ax(k) + B_1 e(k) + B_2 w(k) + B_3 v(k) \\ y(k) &= C_1 x(k) + D_{11} e(k) + D_{12} w(k) + D_{13} v(k) \end{aligned}$$

Consider a standard predictive control problem in the SYSTEM format, so

$$\mathtt{sys} = \left[\begin{array}{c|ccc} A & B_1 & B_2 & B_3 \\ \hline C_1 & D_{11} & D_{12} & D_{13} \\ C_2 & D_{21} & D_{22} & D_{23} \end{array}\right]$$

with dimension vector `dim`. Using

```
[A,B1,B2,B3,C1,D11,D12,D13,C2,D21,D22,D23]=syst2ss(sys,dim);
```

the state space equations become:

$$x(k+1) = Ax(k) + B_1 e(k) + B_2 w(k) + B_3 v(k)$$

$$
\begin{aligned}
y(k) &= C_1 x(k) + D_{11} e(k) + D_{12} w(k) + D_{13} v(k) \\
z(k) &= C_2 x(k) + D_{21} e(k) + D_{22} w(k) + D_{23} v(k)
\end{aligned}
$$

Consider a standard predictive control problem after prediction in `SYSTEM` format, so

$$
\texttt{sysp} = \left[\begin{array}{c|ccc}
A & \tilde{B}_1 & \tilde{B}_2 & \tilde{B}_3 \\
\hline
\tilde{C}_1 & \tilde{D}_{11} & \tilde{D}_{12} & \tilde{D}_{13} \\
\tilde{C}_2 & \tilde{D}_{21} & \tilde{D}_{22} & \tilde{D}_{23} \\
\tilde{C}_3 & \tilde{D}_{31} & \tilde{D}_{32} & \tilde{D}_{33} \\
\tilde{C}_4 & \tilde{D}_{41} & \tilde{D}_{42} & \tilde{D}_{43}
\end{array}\right]
$$

with dimension vector `dimp`. Using

```
[A,tB1,tB2,tB3,tC1,D11,D12,D13,tC2,tD21,tD22,tD23,...
 tC3,tD31,tD32,tD33,tC4,tD41,tD42,tD43]= syst2ss(sysp,dimp);
```

the state space equations become:

$$
\begin{aligned}
x(k+1) &= A x(k) + \tilde{B}_1 e(k) + \tilde{B}_2 \tilde{w}(k) + \tilde{B}_3 \tilde{v}(k) \\
\tilde{y}(k) &= \tilde{C}_1 x(k) + \tilde{D}_{11} e(k) + \tilde{D}_{12} \tilde{w}(k) + \tilde{D}_{13} \tilde{v}(k) \\
\tilde{z}(k) &= \tilde{C}_2 x(k) + \tilde{D}_{21} e(k) + \tilde{D}_{22} \tilde{w}(k) + \tilde{D}_{23} \tilde{v}(k) \\
\tilde{\phi}(k) &= \tilde{C}_3 x(k) + \tilde{D}_{31} e(k) + \tilde{D}_{32} \tilde{w}(k) + \tilde{D}_{33} \tilde{v}(k) \\
\tilde{\psi}(k) &= \tilde{C}_4 x(k) + \tilde{D}_{41} e(k) + \tilde{D}_{42} \tilde{w}(k) + \tilde{D}_{43} \tilde{v}(k)
\end{aligned}
$$

See Also

`io2iio, imp2syst, tf2syst, ss2syst`

io2iio

Purpose

Transformes a IO system (Go) into a IIO system (Gi)

Synopsis

```
[Gi,dimi]=io2iio(Go,dimo);
```

Description

The function transforms an IO system into an IIO system

The state space representation of the input system is:

$$\texttt{Go} = \left[\begin{array}{c|ccc} Ao & Ko & Lo & Bo \\ \hline Co & DH & DF & 0 \end{array}\right]$$

and the dimension vector

$$\texttt{dimo} = \left[\begin{array}{ccccc} na & nk & nl & nb & nc \end{array}\right]$$

The state space representation of the output system is:

$$\texttt{Gi} = \left[\begin{array}{c|ccc} Ai & Ki & Li & Bi \\ \hline Ci & DH & DF & 0 \end{array}\right]$$

and a dimension vector

$$\texttt{dimi} = \left[\begin{array}{ccccc} na+nc & nk & nl & nb & nc \end{array}\right]$$

The function `io2iio` returns the state space IIO-model `Gi` and its dimension `dimi`, The input and output variables are the following:

`Gi`	The increment input output(IIO) model in `SYSTEM` format.
`dimi`	the dimension of the IIO model.
`Go`	The Input output(IO) model in `SYSTEM` format.
`dimo`	the dimension of the IO model.

gpc2spc

Purpose

Transforms GPC problem into a SPC problem

Synopsis

```
[sys,dim]=gpc2spc(Gi,dimi,P,lambda);
```

Description

The function transforms the GPC problem into a standard predictive control problem. For an IIO model G_i:

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + B_i \Delta u(k) \\ y(k) &= C_i x_i(k) + e_i(k) \end{aligned}$$

The weighting $P(q)$ is given in state space (!!) representation:

$$\begin{aligned} x_p(k+1) &= A_p x_p(k) + B_p y(k) \\ y_p(k) &= C_p x_p(k) + D_p y(k) \end{aligned}$$

Where x_p is the state of the realization describing $y_p(k) = P(q)y(k)$.

After the transformation, the system becomes

$$\begin{aligned} x(k+1) &= Ax(k) + Bv(k) + Ke(k) \\ y(k) &= C_1 x(k) + e(k) \\ z(k) &= C_2 x(k) + D_1 e(k) + D_2 w(k) + D_3 v(k) \\ z(k) &= \begin{bmatrix} y_p(k+1) - r(k+1) \\ \lambda \Delta u(k) \end{bmatrix} \\ w(k) &= \begin{bmatrix} d(k) \\ r(k+1) \end{bmatrix} \end{aligned}$$

The IO model `Gi`, the weighting filter `P` and the transformed system `sys` are given in the `SYSTEM` format. The input and output variables are the following:

`sys`	Standard model of SPC problem in `SYSTEM` format.
`dim`	The dimension in SPC problem.
`Gi`	The IIO model in `SYSTEM` format.
`P`	The weighting polynominal.
`lambda`	Weighting factor.

See Also

`lqpc2spc`, `gpc`

lqpc2spc

Purpose

Transforms LQPC problem into a SPC problem

Synopsis

```
[sys,dim]=lqpc2spc(Go,dimo,Q,R);
```

Description

The function transforms the LQPC problem into a standard predictive control problem. Consider the IO model G_o:

$$\begin{aligned} x_o(k+1) &= A_o x_o(k) + B_o u(k) + k e_o(k) \\ y(k) &= C_o x_o(k) + e_o(k) \end{aligned}$$

After the transformation, the system becomes

$$\begin{aligned} x(k+1) &= Ax(k) + Bv(k) + Ke(k) \\ z(k) &= \begin{bmatrix} Q^{1/2} x(K+1) \\ R^{1/2} u(k) \end{bmatrix} \\ w(k) &= d(k) \end{aligned}$$

where

Q	is the state weighting matrix
R	is the control weighting matrix

The IO model `Gi` and the transformed system `sys` are given in the `SYSTEM` format. The input and output variables are the following:

`sys`	Standard model of SPC problem in `SYSTEM` format.
`dim`	The dimension in SPC problem.
`Go`	The IO model in `SYSTEM` format.
`Q`	The state weighting matrix .
`R`	The control weighting matrix.

See Also

`gpc2spc, lqpc`

add_du

Purpose

Function to add increment input constraint to standard problem

Synopsis

```
[sys,dim] = add_du(sys,dim,dumax,sgn,descr);
[sysp,dimp]= add_du(sysp,dimp,dumax,sgn,descr);
```

Description

The function adds an increment input constraint to the standard problem (`sys`) or the prediction model (`sysp`). For `sgn`=1 , the constraint is given by:

$$\Delta u(k+j) \leq |\Delta u_{max}| \quad \text{for} \quad j = 0, \cdots, N-1$$

For `sgn`=-1 , the constraint is given by:

$$\Delta u(k+j) \geq -|\Delta u_{max}| \quad \text{for} \quad j = 0, \cdots, N-1$$

For `sgn`=0 , the constraint is given by:

$$-|\Delta u_{max}| \leq \Delta u(k+j) \leq |\Delta u_{max}| \quad \text{for} \quad j = 0, \cdots, N-1$$

where

Δu_{max}	is the maximum bound on the increment input signal
$\Delta u(k+j)$	is the input increment signal

For MIMO systems Δu_{max} can be choosen as a scalar value or a vector value (with the same dimension as vector $\Delta u(k)$). In the case of a scalar the bound Δu_{max} holds for each element of $\Delta u(k)$. In the case of a vector Δu_{max}, the bound is taken elementwise.

The function add_du returns the system, subjected to the increment input constraint. The input and output variables are the following:

`sys`	Standard model
`dim`	The dimension of the standard model.
`sysp`	Prediction model
`dimp`	The dimension of the prediction model.
`dumax`	The maximum bound on the increment input.
`sgn`	Scalar with value of +1/0/-1.
`descr`	Flag for model type (descr='IO' for input output model. descr='IIO' for an increment input output model).

Setting `dumax=0` means no input constraint will be added.

See Also

`add_u, add_v, add_y, add_x, gpc`

add_u

Purpose

Function to add input constraint to standard problem

Synopsis

```
[sys,dim]=add_u(sys,dim,umax,sgn,descr);
[sysp,dimp]=add_u(sysp,dimp,umax,sgn,descr);
```

Description

The function adds input constraint to the standard problem (`sys`) or the prediction model (`sysp`). For `sgn`=1 , the constraint is given by:

$$u(k+j) \le |u_{max}| \quad \text{for } j = 0, \cdots, N-1$$

For `sgn`=-1 , the constraint is given by:

$$u(k+j) \ge -|u_{max}| \quad \text{for } j = 0, \cdots, N-1$$

For `sgn`=0 , the constraint is given by:

$$-|u_{max}| \le u(k+j) \le |u_{max}| \quad \text{for } j = 0, \cdots, N-1$$

Where

u_{max}	is the maximum bound on the input signal
$u(k+j)$	is the input signal

For MIMO systems u_{max} can be choosen as a scalar value or a vector value (with the same dimension as vector $u(k)$). In the case of a scalar the bound u_{max} holds for each element of $u(k)$. In the case of a vector u_{max}, the bound is taken elementwise.

The function add_u returns the system, subjected to the input constraint. The input and output variables are the following:

`sys`	Standard model
`dim`	The dimension of the standard model.
`sysp`	Prediction model
`dimp`	The dimension of the prediction model.
`umax`	The maximum bound on the input signal.
`sgn`	Scalar with value of +1/0/-1.
`descr`	Flag for model type (descr='IO' for input output model. descr='IIO' for an increment input output model).

Setting `umax=0` means no input constraint will be added.

See Also

`add_du, add_v, add_y, add_x, lqpc`

add_v

Purpose

Function to add input constraint to standard problem

Synopsis

```
[sys,dim] = add_v(sys,dim,vmax,sgn);
[sysp,dimp]= add_v(sysp,dimp,vmax,sgn);
```

Description

The function adds an increment input constraint to the standard problem (`sys`) or the prediction model (`sysp`). For `sgn`=1 , the constraint is given by:

$$v(k+j) \leq |v_{max}| \quad \text{for} \quad j = 0, \cdots, N-1$$

For `sgn`=-1 , the constraint is given by:

$$v(k+j) \geq -|v_{max}| \quad \text{for} \quad j = 0, \cdots, N-1$$

For `sgn`=0 , the constraint is given by:

$$-|v_{max}| \leq v(k+j) \leq |v_{max}| \quad \text{for} \quad j = 0, \cdots, N-1$$

where

v_{max}	is the maximum bound on the input signal
$v(k+j)$	is the input signal

For MIMO systems v_{max} can be choosen as a scalar value or a vector value (with the same dimension as vector $v(k)$). In the case of a scalar the bound v_{max} holds for each element of $v(k)$. In the case of a vector v_{max}, the bound is taken elementwise.

The function add_v returns the system, subjected to the increment input constraint. The input and output variables are the following:

`sys`	Standard model
`dim`	The dimension of the standard model.
`sysp`	Prediction model
`dimp`	The dimension of the prediction model.
`vmax`	The maximum bound on the input.
`sgn`	Scalar with value of +1/0/-1.

Setting `vmax=0` means no input constraint will be added.

See Also

`add_u, add_du, add_y, add_x, gpc`

add_y

Purpose

Function to add output constraint to to the standard problem (`sys`) or the prediction model (`sysp`).

Synopsis

```
[sys,dim] =add_y(sys,dim,ymax,sgn);
```

```
[sysp,dimp]=add_y(sysp,dimp,ymax,sgn);
```

Description

The function adds output constraint to the standard problem (`sys`) or the prediction model (`sysp`). For `sgn`=1 , the constraint is given by:

$$y(k+j) \leq |y_{max}| \quad \text{for } j = 0, \cdots, N-1$$

For `sgn`=-1 , the constraint is given by:

$$y(k+j) \geq -|y_{max}| \quad \text{for } j = 0, \cdots, N-1$$

For `sgn`=0 , the constraint is given by:

$$-|y_{max}| \leq y(k+j) \leq |u_{max}| \quad \text{for } j = 1, \cdots, N-1$$

Where

y_{max}	is the maximum bound on the output signal
$y(k+j)$	is the system output signal

For MIMO systems y_{max} can be choosen as a scalar value or a vector value (with the same dimension as vector $y(k)$). In the case of a scalar the bound y_{max} holds for each element of $y(k)$. In the case of a vector y_{max}, the bound is taken elementwise.

The function add_y returns the system subjected to the output constraint. The input and output variables are the following:

`sys`	Standard model
`dim`	The dimension of the standard model.
`sysp`	Prediction model
`dimp`	the dimension of the prediction model.
`ymax`	The maximum bound on the output signal.
`sgn`	Scalar with value of +1/0/-1.

Setting `ymax=0` means no output constraint will be added.

See Also

`add_u, add_du, add_v, add_x`

add_x

Purpose

Function to add state constraint to to the standard problem.

Synopsis

`[sys,dim]=add_x(sys,dim,E);`

`[sys,dim]=add_x(sys,dim,E,sgn);`

Description

The function adds state constraint to the standard problem.
NOTE: The function add_x can NOT be applied to the prediction model !!!! (i.e. add_x must be used before pred).

For `sgn`=1 , the constraint is given by:

$$Ex(k+j) \leq 1 \;\; \text{for } j = 0, \cdots, N-1$$

For `sgn`=-1 , the constraint is given by:

$$-Ex(k+j) \leq 1 \;\; \text{for } j = 0, \cdots, N-1$$

For `sgn`=0 , the constraint is given by:

$$-1 \leq Ex(k+j) \leq 1 \;\; \text{for } j = 1, \cdots, N-1$$

Where

E	is a row vector
$x(k+j)$	is the state of the system

The function add_x returns the standard problem, subjected to the state constraint. The input and output variables are the following:

`sys`	The standard model.
`dim`	The dimension of the standard model.
`E`	Row vector to define bound on the state.
`sgn`	Scalar with value of +1/0/-1.

Setting `E=0` means no output constraint will be added.

See Also

`add_u`, `add_du`, `add_v`, `add_y`

add_nc

Purpose

Function to add a control horizon to prediction problem

Synopsis

```
[sysp,dimp]=add_nc(sysp,dimp,dim,Nc);
[sysp,dimp]=add_nc(sysp,dimp,dim,Nc,descr);
```

Description

The function add_nc returns the system with a control horizon `sysp` and the dimension `dimp`. The input and output variables are the following:

`sysp`	The prediction model.
`dimp`	The dimension of the prediction model.
`dim`	The dimension of the original IO or IIO system
`Nc`	The control horizon.
`descr`	Flag for model type (descr='IO' for input output model, descr='IIO' for an increment input output model)

The default value of `descr` is `'IIO'`.

See Also

`gpc, lqpc`

add_end

Purpose

Function to add a state end-point constraint to prediction problem.

Synopsis

```
[sysp,dimp] = add_end(sys,dim,sysp,dimp,N);
```

Description

The function adds a state end-point constraint to the standard problem:

$$x(k+N) = x_{ss} = D_{ssx}w(k+N)$$

The function add_end returns the system with a state end-point constraint in `sysp` and the dimension `dimp`. The input and output variables are the following:

`sys`	The standard model.
`dim`	The dimension of the standard model.
`sysp`	The system of prediction problem.
`dimp`	The dimension of the prediction problem.
`N`	The prediction horizon.

external

Purpose

Function makes the external signal $\tilde{w}$ from d and r for simulation purpose

Synopsis

```
[tw]=external(apr,lensim,N,w1);
[tw]=external(apr,lensim,N,w1,w2);
[tw]=external(apr,lensim,N,w1,w2,w3);
[tw]=external(apr,lensim,N,w1,w2,w3,w4);
[tw]=external(apr,lensim,N,w1,w2,w3,w4,w5);
```

Description

The function computes the vector $\tilde{w}$ with with predictions of the external signal w:

$$\tilde{w} = \begin{bmatrix} w(k) \\ w(k+1) \\ \vdots \\ w(k+N-1) \end{bmatrix}$$

where

$$w(k) = \begin{bmatrix} w_1(k) \\ \vdots \\ w_5(k) \end{bmatrix}$$

For the lqpc controller, the external signal $w(k)$ is given by

$$w(k) = \begin{bmatrix} d(k) \end{bmatrix}$$

and so we have to choose $w_1(k) = d(k)$. For the gpc controller, the external signal $w(k)$ is given by

$$w(k) = \begin{bmatrix} d(k) \\ r(k+1) \end{bmatrix}$$

and so we have to choose $w_1(k) = d(k)$ and $w_2(k) = r(k+1)$.

tw	is the column vector $\tilde{w}$ with predictions of the external signal w.
apr	vector with flags for a priori knowledge on external signal.
lensim	is length of simulation interval.
N	is the prediction horizon.
w1	is the column vector with external signal $w_1(k)$.
$\vdots$	$\vdots$
w5	is the column vector with external signal $w_5(k)$.

The vector `apr` determine if the external signal $w(k+j)$ is a priori known. For $i = 1, \ldots, 5$ there holds:

$$\begin{cases} \text{signal } w_i(k) \text{ is a priori known for } apr(i) = 1 \\ \text{signal } w_i(k) \text{ is not a priori known for } apr(i) = 0 \end{cases}$$

See Also

`gpc`, `lqpc`

dgamma

Purpose

Function makes the selection matrix $\bar{\Gamma}$. For the infinite horizon case the funtion also creates the steady state matrix Γ_{ss}.

Synopsis

```
[dGamma] = dgamma(Nm,N,dim);
```

```
[dGamma,Gammass] = dgamma(Nm,N,dim);
```

Description

The selection matrix Γ is defined as

$$\bar{\Gamma} = \mathrm{diag}(\Gamma(0), \Gamma(1), \ldots, \Gamma(N-1))$$

where

$$\Gamma(j) = \begin{cases} \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} & \text{for} \quad 0 \leq j \leq N_m - 1 \\ \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} & \text{for} \quad N_m \leq j \leq N - 1 \end{cases}$$

Where

N_m	is the minimum cost- horizon
N	is the prediction horizon
`dim`	is the dimension of the system

The function dgamma returns the diagonal elements of the selection matrix $\bar{\Gamma}$, so `dGamma` $= \mathrm{diag}(\bar{\Gamma})$. The input and output variables are the following:

`dGamma`	vector with diagonal elements of $\bar{\Gamma}$.
`Gammass`	vector with diagonal elements of Γ_{ss}.
`Nm`	is the minimum-cost horizon.
`N`	is the prediction horizon.
`Nc`	is the control horizon.
`dim`	the dimension of the system.

See Also

`gpc`, `lqpc`, `blockmat`

pred

Purpose

Function computes the prediction model from the standard model

Synopsis

```
[sysp,dimp]=pred(sys,dim,N);
```

Description

The function makes a prediction model for the system

$$
\begin{aligned}
x(k+1) &= Ax(k) + B_1 e(k) + B_2 w(k) + B_3 v(k)\\
y(k) &= C_1 x(k) + D_{11} e(k) + D_{12} w(k)\\
z(k) &= C_2 x(k) + D_{21} e(k) + D_{22} w(k) + D_{23} v(k)\\
\phi(k) &= C_3 x(k) + D_{31} e(k) + D_{32} w(k) + D_{33} v(k)\\
\psi(k) &= C_4 x(k) + D_{41} e(k) + D_{42} w(k) + D_{43} v(k)
\end{aligned}
$$

for

$$
\texttt{sys} = \begin{bmatrix} A & B_1 & B_2 & B_3\\ C_1 & D_{11} & D_{12} & 0\\ C_2 & D_{21} & D_{22} & D_{23}\\ C_3 & D_{31} & D_{32} & D_{33}\\ C_4 & D_{41} & D_{42} & D_{43} \end{bmatrix}
$$

$$
\texttt{dim} = \begin{bmatrix} n_a & n_{b1} & n_{b2} & n_{b3} & n_{c1} & n_{c2} & n_{c3} & n_{c4} \end{bmatrix}
$$

Where

$x(k)$	is the state of the system
$e(k)$	is zero-mean white noise (ZMWN)
$w(k)$	is known external signal
$v(k)$	is the control signal
$z(k)$	is the performance signal
$\phi(k)$	is the equality constraint at time k
$\psi(k)$	is the inequality constraint at time k
$A, B_, C_j, D_m$	the system matrices

For the above model, the prediction model is given by

$$
\begin{array}{rcl}
x(k+1) & = & Ax(k) + \tilde{B}_1 e(k) + \tilde{B}_2 \tilde{w}(k) + \tilde{B}_3 \tilde{v}(k) \\
\tilde{y}(k) & = & \tilde{C}_1 x(k) + \tilde{D}_{11} e(k) + \tilde{D}_{12} \tilde{w}(k) + \tilde{D}_{13} \tilde{v}(k) \\
\tilde{z}(k) & = & \tilde{C}_2 x(k) + \tilde{D}_{21} e(k) + \tilde{D}_{22} \tilde{w}(k) + \tilde{D}_{23} \tilde{v}(k) \\
\tilde{\phi}(k) & = & \tilde{C}_3 x(k) + \tilde{D}_{31} e(k) + \tilde{D}_{32} \tilde{w}(k) + \tilde{D}_{33} \tilde{v}(k) \\
\tilde{\psi}(k) & = & \tilde{C}_4 x(k) + \tilde{D}_{41} e(k) + \tilde{D}_{42} \tilde{w}(k) + \tilde{D}_{43} \tilde{v}(k)
\end{array}
$$

where

$$
\tilde{p} = \begin{bmatrix} \hat{p}(k|k) \\ \hat{p}(k+1|k) \\ \vdots \\ \hat{p}(k+N|k) \end{bmatrix}
$$

stands for the predicted signal for $p = y, z, \phi, \psi$ and for future values for $p = v, w$.

The function `pred` returns the prediction model `sysp`:

$$
\texttt{sysp} = \begin{bmatrix} A & \tilde{B}_1 & \tilde{B}_2 & \tilde{B}_3 \\ \tilde{C}_1 & \tilde{D}_{11} & \tilde{D}_{12} & \tilde{D}_{13} \\ \tilde{C}_2 & \tilde{D}_{21} & \tilde{D}_{22} & \tilde{D}_{23} \\ \tilde{C}_3 & \tilde{D}_{31} & \tilde{D}_{32} & \tilde{D}_{33} \\ \tilde{C}_4 & \tilde{D}_{41} & \tilde{D}_{42} & \tilde{D}_{43} \end{bmatrix}
$$

$$
\texttt{dimp} = \begin{bmatrix} n_a & n_{b1} & N \cdot n_{b2} & N \cdot n_{b3} & N \cdot n_{c1} & N \cdot n_{c2} & N \cdot n_{c3} & N \cdot n_{c4} \end{bmatrix}
$$

The input and output variables are the following:

`sysp`	The prediction model.
`dimp`	The dimension of the prediction model.
`sys`	The system to be controlled.
`dim`	The dimension of the system.
`N`	The prediction horizon.

See Also

`gpc, lqpc, blockmat`

contr

Purpose

Function computes controller matrices for predictive controller

Synopsis

```
[vv,HH,AA,bb]=contr(sysp,dimp,dim,dGam);
```

Description

The function computes the matrices for predictive controller for use in the function `simul`.

`sysp` is the prediction model, `dimp` is the dimension-vector of the prediction model, `dim` is the dimension-vector of the standard model, and `dGam` is a row-vector with the diagonal elements of the matrix $\bar{\Gamma}$.

If inequality constraints are present, the control problem is transformed into a quadratic programming problem, where the objective function is chosen as

$$J(\mu, k) = \frac{1}{2}\tilde{\mu}^T(k)H\tilde{\mu}(k)$$

subject to

$$A\tilde{\mu} - b \leq 0$$

which can be solved at each time instant k in the function `simul`. There holds The variable `vv` is given by:

$$\mathtt{vv} = \begin{bmatrix} -F & D_e & D_w & D_\mu \end{bmatrix}$$

so that

$$\begin{aligned} v(k) &= \mathtt{vv}\begin{bmatrix} x_c(k) \\ \hat{e}(k) \\ \tilde{w}(k) \\ \tilde{\mu}(k) \end{bmatrix} \\ &= -F\,x_c(k) + D_e\,\hat{e}(k) + D_w\,\tilde{w}(k) + D_\mu\tilde{\mu}(k) \end{aligned}$$

Further `bb` is such that

$$b(k) = \texttt{bb}\begin{bmatrix} 1 \\ x_c(k) \\ \hat{e}(k) \\ \tilde{w}(k) \end{bmatrix}$$

If no inequality constraints are present, we find $\tilde{\mu} = 0$ and the control problem can be solved analytically at each time instant in the function `simul`, using

$$v(k) = -F\, x_c(k) + D_e\, \hat{e}(k) + D_w\, \tilde{w}(k)$$

The variable `vv` is now given by:

$$\texttt{vv} = \begin{bmatrix} -F & D_e & D_w \end{bmatrix}$$

so that

$$v(k) = \texttt{vv}\begin{bmatrix} x_c(k) \\ \hat{e}(k) \\ \tilde{w}(k) \end{bmatrix}$$

The input and output variables are the following:

`sysp`	The prediction model.
`dimp`	The dimension of the prediction model.
`dim`	the dimension of the system.
`dGam`	vector with diagonal elements of $\bar{\Gamma}$.
`vv,HH,AA,bb`	controller parameters.

See Also

`simul, gpc, lqpc`

contrinf

Purpose

Function computes controller matrices for predictive controller for the infinite prediction horizon case.

Synopsis

```
[vv]= contrinf(sys,sysp,dim,dimp,Nc,dGam,dGamss);
```

Description

The function computes the matrices for predictive controller for use in the function `simul` in the infinite prediction horizon case ($N = \infty$).

`sysp` is the prediction model, `dimp` is the dimension-vector of the prediction model, `dim` is the dimension-vector of the system, and `dGam` is a row-vector with the diagonal elements of the matrix $\bar{\Gamma}$.

Only the LTI case is implemented, so inequality constraints can not be included. The control problem can be solved analytically at each time instant in the function `simul`, using

$$v(k) = -F\, x_c(k) + D_e\, \hat{e}(k) + D_w\, \tilde{w}(k)$$

The variable `vv` is given by:

$$\mathtt{vv} = \begin{bmatrix} -F & D_e & D_w \end{bmatrix}$$

so that

$$v(k) = \mathtt{vv} \begin{bmatrix} x_c(k) \\ \hat{e}(k) \\ \tilde{w}(k) \end{bmatrix}$$

`sys`	The standard model.
`sysp`	The prediction model.
`dim`	the dimension of the system.
`dimp`	The dimension of the prediction model.
`dGam`	vector with diagonal elements of $\bar{\Gamma}$.
`dGamss`	vector with diagonal elements of $\bar{\Gamma}_{ss}$.
`vv`	controller parameters.

See Also

`simul`, `gpc`, `contr`

lticll

Purpose

Function computes the state space matrices of the closed loop in the LTI case (no inequality constraints).

Synopsis

```
[Acl,B1cl,B2cl,Ccl,D1cl,D2cl]=lticll(sys,dim,vv,N);
```

Description

Function `lticll` computes the state space matrices of the LTI closed loop, given by

$$\begin{aligned} x_{cl}(k+1) &= A_{cl}\, x_{cl}(k) + B_{1cl}\, e(k) + B_{2cl}\, \tilde{w}(k) \\ \begin{bmatrix} v(k) \\ y(k) \end{bmatrix} &= C_{cl}\, x_{cl}(k) + D_{1cl}\, e(k) + D_{2cl}\, \tilde{w}(k) \end{aligned}$$

The input and output variables are the following:

`sys`	the model.
`dim`	dimensions sys.
`N`	the prediction horizon.
`vv`	controller parameters.
`Acl,B1cl,B2cl,Ccl,D1cl,D2cl`	system matrices of closed loop

See Also

`gpc, lqpc, contr, simul, lticon`

lticon

Purpose

Function computes the state space matrices of the optimal predictive controller in the LTI case (no inequality constraints).

Synopsis

```
[Ac,B1c,B2c,Cc,D1c,D2c]=lticon(sys,dim,vv,N);
```

Description

Function `lticon` computes the state space matrices of the optimal predictive controller, given by

$$\begin{aligned} x_c(k+1) &= A_c\, x_c(k) + B_{1c}\, y(k) + B_{2c}\, \tilde{w}(k) \\ v(k+1) &= C_c\, x_c(k) + D_{1c}\, y(k) + D_{2c}\, \tilde{w}(k) \end{aligned}$$

The input and output variables are the following:

`sys`	the model.
`dim`	dimensions sys.
`N`	the prediction horizon.
`vv`	controller parameters.
`Ac,B1c,B2c,Cc,D1c,D2c`	system matrices of LTI controller

See Also

`gpc, lqpc, contr, simul, lticll`

simul

Purpose

Function makes a closed loop simulation with predictive controller

Synopsis

```
[x,xc,y,v]=simul(syst,sys,dim,N,x,xc,y,v,tw,e,...
                           k1,k2,vv,HH,AA,bb);
```

Description

The function makes a closed loop simulation with predictive controller, the input and output variables are the following:

`syst`	the true system.
`sys`	the model.
`dim`	dimensions sys and syst.
`N`	the horizon.
`x`	state of the true system.
`xc`	state of model/controller.
`y`	the true system output.
`v`	the true system input.
`tw`	vector with predictions of the external signal.
`k1`	begin simulation interval.
`k2`	end simulation interval.
`vv,HH,AA,bb`	controller parameters.

See Also

`contr, gpc, lqpc`

rhc

Purpose

Function makes a closed loop simulation in receding horizon mode

Synopsis

```
[x,xc,y,v]=rhc(syst,sys,sysp,dim,dimp,dGam,N,x,xc,y,v,tw,e,k1,k2);
```

Description

The function makes a closed loop simulation with predictive controller in the receding horizon mode. Past signals and prediction of future signals are given within the prediction range. The simulation is paused every sample and can be continued by pressing the return-key. The input and output variables are the following:

`syst`	the true system.
`sys`	the model.
`sysp`	the prediction model.
`dim`	dimensions sys and syst.
`dimp`	dimensions sysp.
`dGamma`	vector with diagonal elements of $\bar{\Gamma}$.
`N`	the horizon.
`x`	state of the true system.
`xc`	state of model/controller.
`y`	the true system output.
`v`	the true system input.
`tw`	vector with predictions of the external signal.
`k1`	begin simulation interval.
`k2`	end simulation interval.

See Also

`simul, contr, gpc, lqpc`

Index

Bibliography

[1] M. Alamir and G. Bornard. New sufficient conditions for global stability of receding horizon control for discrete-time nonlinear systems. In *Advances in MBPC*, Oxford University Press, 1994.

[2] F. Allgöwer, T.A. Badgwell, J.S. Qin, and J.B. Rawlings. Nonlinear predictive control and moving horizon estrimation – an introductory overview. In *Advances in Control, Highlights of ECC'99, Edited by F. Frank*, Springer-Verlag, London, UK, 1999.

[3] K.J. Åström and P. Eykhoff. System identification - a survey. *Automatica*, 7:123–162, 1971.

[4] K.J. Åström and B. Wittenmark. On self-tuning regulators. *Automatica*, 9:185–199, 1973.

[5] M.G. Ayalla Botto, T.J.J. van den Boom, A.J. Krijgsman, and J.S. da Costa. Constrained nonlinear predictive control based on input-output linearization using a neural network model. *International Journal of Control*, 72(17), 1999.

[6] A.C.P.M. Backx. Identification of an industrial process: A Markov parameter approach. PhD-thesis, Eindhoven University of Technology, The Netherlands, 1987.

[7] A.C.P.M. Backx and A.A.H. Damen. Identification for the control of MIMO processes. *IEEE AC*, 30:980–986, 1992.

[8] A.C.P.M. Backx and A.H. Damen. Identification of industrial MIMO processes for fixed controllers, part 2: Case studies. *Journal A*, 30(2):33–43, 1989.

[9] A.C.P.M. Backx and A.H. Damen. Identification of industrial MIMO processes for fixed controllers, part 1: General theory & practice. *Journal A*, 30(1):3–12, 1989.

[10] V. Balakrishnan, A. Zheng, and M. Morari. Constrained stabilization of discrete-time systems. In *Advances in MBPC*, Oxford University Press, 1994.

[11] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The expicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[12] B.W. Bequette. Nonlinear control of chemical processes: A review. *Ind. Eng. Chem. Res.*, 30(7):1391–1413, 1991.

[13] R.R. Bitmead, M. Gevers, and V. Wertz. *Adaptive Optimal Control. The Thinking Man's GPC*. Prentice Hall, Upper Saddle River, New Jersey, 1990.

[14] H. H. J. Bloemen, T. J. J. van den Boom, and H. B. Verbruggen. Optimizing the end point state weighting in Model-based Predictive Control. *Automatica*, 38(6):1061–1068, June 2002.

[15] S.P. Boyd and C.H. Barratt. *Linear Controller Design, Limits of performance*. Prentice Hall, Information and System Sciences Series, Englewood Cliffs, New Jersey, 1991.

[16] D.D. Brengel and W.D. Seider. Multistep nonlinear predictive controller. *Ind. Eng. Chem. Res.*, 28(12):1812–1822, 1989.

[17] M.L. Brisk. Process control: Theories and profits. In *IFAC World Congress*, volume 7, pages 241–250, 1993.

[18] E.F. Camacho and C. Bordons. *Model Predictive Control in the Process Industry, Advances in Industrial Control*. Springer, London, 1995.

[19] D.W. Clarke. Application of generalized predictive control to industrial processes. *IEEE Control Systems Magazine*, 8(2):49–55, 1988.

[20] D.W. Clarke and C. Mohtadi. Properties of generalized predictive control. *Automatica*, 25(6):859–875, 1989.

[21] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control - part 1. the basic algorithm. *Automatica*, 23(2):137–148, 1987.

[22] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control - part 2. extensions and interpretations. *Automatica*, 23(2):149–160, 1987.

[23] D.W. Clarke and R. Scattolini. Constrained receding horizon predictive control. *IEE Proc-D*, 138(4), 1991.

[24] C.R. Cutler. *Dynamic matrix control, An optimal multivariable control algorithm with constraints*. Dissertation, Faculty of Chemical Engineering, University of Houston, 1983.

[25] C.R. Cutler and R.B. Hawkins. Constrained multivariable control of a hydrocracker reactor. In *ACC*, pages 1014–1020, 1987.

[26] C.R. Cutler and B.L. Ramaker. Dynamic matrix control - a computer control algorithm. In *AIChE Nat. Mtg*, 1979.

[27] C.R. Cutler and B.L. Ramaker. Dynamic matrix control - a computer control algorithm. In *Proceeding Joint American Control Conference, San Francisco, CA, USA*, 1980.

[28] R.M.C. De Keyser and A.R. Van Cauwenberghe. Microcomputer-controlled servo system based on self-adaptive long-range prediction. In *Adv. Meas. & Contr. MECO'83*, 1983.

[29] R.M.C. De Keyser and A.R. van Cauwenberghe. Typical application possibilities for self-tuning predictive control. In *IFAC Symp.Ident.& Syst.Par.Est.*, 1982.

[30] D. De Vries. *Identification of Model Uncertainty for Control Design.* PhD Thesis, Delft University of Technology, Delft, The Netherlands, 1994.

[31] R.A.J. De Vries and H.B Verbruggen. Multivariable process and prediction models in predictive control – a unified approach. *Int. J. Adapt. Contr.& Sign. Proc.*, 8:261–278, 1994.

[32] F.J. Doyle, B.A. Ogunnaike, and R.K.Pearson. Nonlinear model-based control using second-order volterra models. *Automatica*, 31(5):697–714, 1995.

[33] J.C. Doyle. Analysis of feedback systems with structured uncertainties. In *IEE Proceedings*, volume 129-D, pages 242–250, 1982.

[34] J.C. Doyle. Lecture notes. ONR/Honeywell Workshop on Advances in Multivariable Control, 1984.

[35] J.C. Doyle, B.A. Francis, and A.R. Tannenbaum. *Feedback control systems.* MacMillan Publishing Company, New York, USA, 1992.

[36] J.C. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis. State-space solutions to standard H_2 and H_∞ control problems. *IEEE AC*, 34:pp831–847, 1989.

[37] J.C. Doyle, A. Packard, and K. Zhou. Review on LFTs, LMIs and μ. In *Proc. of the 30th Conference on Decision and Control, Brighton, UK*, pages 1227–1232, 1991.

[38] J.W. Eaton and J.B. Rawlings. Feedback control of chemical processes using on-line optimization techniques. *Comput.Chem.Eng.*, 14:469–479, 1990.

[39] J.W. Eaton, J.B. Rawlings, and T.F. Edgar. Model-predictive control and sensitivity analysis for constrained nonlinear processes. In *IFAC Worksh. on MBPC, ed. Arkun& Zafiriou*, pages 129–135, Pergamon Press, New York, 1989.

[40] J.W. Eaton, J.B. Rawlings, and L.H. Ungar. Stability of neural net based model predictive control. In *ACC*, 1995.

[41] C.G. Economou and M. Morari. Newton control laws for nonlinear controller design. In *CDC*, 1985.

[42] C.G. Economou, M. Morari, and B.O. Palsson. Internal model control 6. extension to nonlinear systems. *Ind.Eng.Chem.Proc.Des.Dev.*, 25:403–411, 1986.

[43] P. Eykhoff. *System Identification; Parameter and State Estimation.* John Wiley & Sons, 1974.

[44] H. Falkus. *Parametric Uncertainty in System Identification.* PhD Thesis,Eindhoven University of Technology, Eindhoven, The Netherlands, 1994.

[45] E-B. Feng, J-S. Yu, and W-S. Jiang. New method for predictive controller design for bilinear systems. *Int.J.Control*, 53(1):97–111, 1991.

[46] C.K. Finn, B. Wahlberg, and B.E. Ydstie. Constrained predictive control using orthogonal expansions. *AIChE Journal*, 39(11):1810–1826, 1993.

[47] C.E. Garcia. Quadratic/dynamic matric control of nonlinear processes: An application to a batch reaction process. In *AIChE Annual Meeting*, 1984.

[48] C.E. Garcia and M. Morari. Internal model control. 1. a unifying review and some new results. *Ind.Eng.Chem.Proc.Des.Dev.*, 21(2):308–323, 1982.

[49] C.E. Garcia and A.M. Morshedi. Quadratic programming solution of dynamic matrix control (QDMC). *Chem.Eng.Comm.*, 46(11):73–87, 1986.

[50] C.E. Garcia, D.M. Prett, and M. Morari. Model predictive control: Theory and practice - a survey. *Automatica*, 25(3):335–348, 1989.

[51] C.E. Garcia, D.M. Prett, and B.L. Ramaker. *Fundamental Process Control.* Butterworths, Stoneham, MA, 1988.

[52] H. Genceli and M. Nikolaou. Robust stability analysis of constrained ℓ_1-norm model predictive control. *AIChE Journ.*, 39(12):1954–1965, 1993.

[53] W Gerth. *Zur Minimalrealisierung von Mehrgrössenubertragungssytemen durch Markovparameter.* PhD Thesis, Fakultät für Maschinenwesen, Technische Universität, Hannover, 1972.

[54] P. Grossdidier, B. Froisy, and M. Hammann. The idcom-m controller. In *IFAC Workshop on Model Based Control*, 1988.

[55] R.G. Hakvoort. *System Identification for Robust Process Control, Nominal Models and Error Bounds.* PhD Thesis, Delft University of Technology, Delft, The Netherlands, 1994.

[56] M.A. Henson and D.E. Seborg. Theoretical analysis of unconstrained nonlinear model predictive control. *Int.J.Contr.*, 58(5):1053–1080, 1993.

[57] P.S.C. Heuberger. *On Approximate System Identification with System-based Othonormal Functions*. PhD Thesis, Delft University of Technology, Delft, The Netherlands, 1990.

[58] B.L. Ho and R.E. Kalman. Effective construction of linear state-variable models from input/output functions. *Regelungstechnik*, (14), 1966.

[59] Peter Hoekstra, Ton J.J. van den Boom, and Miguel Ayala Botto. Design of an analytic constrained predictive controller using neural networks. In *European Control Conference 2001, Porto, Portugal*, pages 3300–3305, 2001.

[60] T. Kailath. *Linear Systems*. Prentice Hall, New York, 1980.

[61] Kalman. 1960.

[62] M. Kinnaert. Adaptive generalized predictive controller for mimo systems. *Int.J. Contr.*, 50(1):161–172, 1989.

[63] M.V. Kothare, V. Balakrishnan, and M. Morari. Robust contrained predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.

[64] W.H. Kwon. Advances in predictive control: Theory and application. In *ASCC*, 1994.

[65] W.H. Kwon and A.E. Pearson. On feedback stabilization of time-varying discrete systems. *IEEE AC*, 23:479–481, 1979.

[66] F. Lebourgeois. Idcom, application and experiances on a pvc production plant. In *Joint Aut.Cont.Conf.*, 1980.

[67] Lee. 1996.

[68] J.H. Lee, S.M. Gelormino, and M. Morari. Model predictive control of multi-rate sampled-data systems: a state-space approach. *Int.J.Contr.*, 55(1), 1992.

[69] J.H. Lee, M. Morari, and C.E. Garcia. State-space interpretation of model predictive control. *Automatica*, 30(4):707–717, 1994.

[70] J.H. Lee and Z.H.Yu. Tuning of model predictive controllers for robust performance. *Comp.Chem.Eng.*, 18(1):15–37, 1994.

[71] C.E. Lemke. On complementary pivot theory. In *Mathematics of the Decision Sciences, G.B.Dantzig and A.F.Veinott (Eds.)*, 1968.

[72] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Englewood Cliffs, NJ, 1987.

[73] L. Ljung and T. Söderström. *Theory and practice of recursive identification.* The MIT press, Cambridge, MA, USA, 1983.

[74] J.M. Maciejowski. *Multivariable Feedback Control Design.* Addison-Wesley Publishers, Wokingham, UK, 1989.

[75] J.M. Maciejowski. *Predictive control with constraints.* Prentice Hall, Pearson Education Limited, Harlow, UK, 2002.

[76] T.N. Matsko. Internal model control for a chemical recovery. *Chem.Eng.Prog.*, 81(12):46–51, 1985.

[77] D.Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE AC*, 35(7):814–824, 1990.

[78] Meadows and Rawlings. 1993.

[79] R.K. Mehra, R. Rouhani, and J. Eterno. *Model Algorithmic Control (MAC): Review and Recent Developments.* In: T.F. Edgar and D.E. Seborg, ed., Chemical Process Control II, 1982.

[80] H. Michalska and D.Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE AC*, 38(11):1623–1633, 1993.

[81] B.C. Moore. Principal component analysis in linear systems: Controllability, observability and model reduction. *IEEE AC*, 26:17–32, 1981.

[82] M. Morari. Process control theory: Reflections on the past and goals for the next decade. In *The Second Shell Process Control Workshop, December 12-16, 1988, Butterworths, Stoneham, MA*, 1988.

[83] M. Morari and E. Zafiriou. *Robust Process Control.* Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[84] E. Mosca. *Optimal Predictive and Adaptive Control.* Prentice Hall, Englewood Cliffs, NJ, 1995.

[85] E. Mosca and J. Zhang. Stable redesign of predictive control. *Automatica*, 28(6):1229–1233,, 1992.

[86] M. M'Saad, L. Dugard, and Sh. Hammad. A suitable generalized predictive adaptive controller case study: Control of a flexible arm. *Automatica*, 29(3):585–608, 1993.

[87] A. Naganawa, G. Obinata, and H. Inooka. A design method of model predictive control system using coprime factorization approach. In *ASCC*, 1994.

[88] V. Nevistić and M. Morari. Constrained control of feedback-linearized systems. In *ECC*, 1995.

[89] V. Nevistić and M. Morari. Robustness of mpc-based schemes for constrained control of nonlinear systems. In *IFAC World Congress*, 1996.

[90] F. Özgülsen, S.J. Kendra, and A. Çinar. Nonlinear predictive comtrol of periodically forced chemical reactors. *AIChE Journ.*, 39(4):589–598, 1993.

[91] A.A. Patwardhan, J.B. Rawlings, and T.E. Edgar. Nonlinear model predictive control. *Chem.Eng.Comm.*, 87(1):123–141, 1990.

[92] L. Pernebo and L.M. Silverman. Model reduction via balanced state space representations. *IEEE AC*, 27:382–387, 1982.

[93] D.M. Prett and R.D. Gillette. Optimization and constrained multivariable control of a catalytic cracking unit. In *AIChE Nat. Mgr.*, 1979.

[94] S.J. Qin and T.A. Badgewell. An overview of industrial model predictive control technology. In *Chemical Process Control - V, AIChe Symposium Series - American Institute of Chemical Engineers*, volume 93, 1997.

[95] et al. Rawlings. 1994.

[96] J.B. Rawlings and K.R. Muske. The stability of constrained receding horizon control. *IEEE AC*, 38:1512–1516, 1993.

[97] J. Richalet. 1976.

[98] J. Richalet. Industrial applications of model based predictive control. *Automatica*, 29(5):1251–1274, 1993.

[99] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(1):413–428, 1978.

[100] N.L. Ricker, T. Subrahmanian, and T. Sim. Case studies of model-predictive control in pulp and paper production. In *Proc. 1988 IFAC Workshop on Model Based Process Control, T.J. McAvoy, Y. Arkun and E. Zafiriou eds., Pergamon Press, Oxford*, page 13, 1988.

[101] R. Rouhani and R.K. Mehra. Model algorithmic control (mac): Basic theoretical properties. *Automatica*, 18(4):401–414, 1982.

[102] J. Saint-Donat, N. Bhat, and T.J. McAvoy. Neural net based model predictive control. *Int.J.Control*, 54(6):1453–1468, 1991.

[103] R. Schrama. *Approximate Identification and Control Design*. PhD Thesis, Delft University of Technology, Delft, The Netherlands, 1992.

[104] et al. Scockaert. 1999.

[105] P.O.M. Scockaert and D.W. Clarke. Stability and feasibility in constrained predictive control. In *Advances in MBPC*, Oxford University Press, 1994.

[106] P.O.M. Scokaert. Infinite horizon genralized predictive control. *Int. J. Control*, 66(1):161–175, 1997.

[107] T. Söderström and P. Stoica. *System Identification*. Prentice Hall, UK, 1989.

[108] A.R.M. Soeterboek. *Predictive control - A unified approach*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[109] A.R.M. Soeterboek, A.F. Pels, H.B. Vergruggen, and G.C.A. van Langen. A predictive controller for the mach number in a transonic wind tunnel. *IEEE Control Systems Magazine*, 1991.

[110] S. Sommer. Model-based predictive control methods based on nonlinear and bilinear parametric system descriptions. In *Advances in MBPC*, Oxford University Press, 1994.

[111] E.D. Sontag. An algebraic approach to bounded controllability of linear systems. *Int.J.Contr.*, pages 181–188, 1984.

[112] V. Strejc. *State Space Theory of Discrete Linear Control*. Academia, Prague, 1981.

[113] H.A.C. Swaanenburg, W.M.M. Schinkel, G.A. van Zee, and O.H. Bosgra. Practical aspects of industrial multivariable process identification. In *Barker H.A. and P.C. Young (Eds.) Identification and System Parameter Estimation, IFAC Proc. Series 1985, York, U.K., pp. 201-206*, 1985.

[114] Y. Tan and R. De Keyser. Neural network based adaptive predictive control. In *Advances in MBPC*, Oxford University Press, 1994.

[115] M.T. Tham, F. Vagi, A.J. Morris, and R.K. Wood. Multivariable and multirate self-tuning control: a distillation column case study. *IEE Proc. Part D*, 138(1):9–24, 1991.

[116] T.J.J. van den Boom. MIMO-systems identification for H_∞ robust control: A frequency domain approach with minimum error bounds. PhD-thesis, Eindhoven University of Technology, The Netherlands, 1993.

[117] T.J.J. van den Boom. Model based predictive control, status and perspectives. In *Tutorial Paper on the CESA IMACS Conference, Lille, France. Symposium on Control, Optimization and Supervision*, volume 1, pages 1–12, Lille, France, 1996.

[118] T.J.J. van den Boom and R.A.J. de Vries. Robust predictive control using a time-varying Youla parameter. *Journal of applied mathematics and computer science*, 9(1):101–128, 1999.

[119] P.M.J. Van den Hof, P.S.C. Heuberger, and J. Bokor. Identification with generalized orthonormal basis functions - statistical analysis and error bounds. In *Proc. IFAC conf. System Identification and Parameter Estimation, Copenhagen*, 1994.

[120] E.T. van Donkelaar. *Improvement of efficiency in system identification and model predictive control of industrial processes*. PhD Thesis, Delft University of Technology, Delft, The Netherlands, 2000.

[121] E.T. van Donkelaar, O.H. Bosgra, and P.M.J. Van den Hof. Model predictive control with generalized input parametrization. In *ECC*, 1999.

[122] P. Van Overschee. *Subspace Identification, Theory-Implementation-Application*. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 1995.

[123] P. Van Overschee and B. de Moor. Subspace algorithms for the stochastic identification problem. *Automatica*, 29(3):649–660, 1993.

[124] M. Verhaegen and P. Dewilde. Subspace model identification. part i: The output-error state space model identification class of algorithms / part ii: Analysis of the elementary output-error state space model identification algorithm. *Int.J.Contr.*, 56(5):1187–1241, 1992.

[125] R.A.J. De Vries and T.J.J. van den Boom. Robust predictive control. In *3rd European Control Conference, Rome, Italy, September 5-8*, pages 1738–1743, 1995.

[126] R.A.J. De Vries and T.J.J. van den Boom. Robust stability constraints for predictive control. In *4th European Control Conference, Brussels, Belgium*, 1997.

[127] B. Wahlberg. System identification using high order models, revisited. In *Proc. 28th Conf. on Decision and Control, Tampa, Florida*, pages 634–639, 1989.

[128] B. Wahlberg and L. Ljung. On estimation of transfer function error bounds. In *Proc. of the European Control Conference, Grenoble, France*, 1991.

[129] et al. Wahlberg. 1993.

[130] J.C. Willems. From time series to linear systems; part i: Finite dimensional linear time invariant systems. *Automatica*, 22:561–580, 1986.

[131] J.C. Willems. From time series to linear systems; part ii: Exact modelling. *Automatica*, 22:675–694, 1986.

[132] J.C. Willems. From time series to linear systems; part iii: Approximate modelling. *Automatica*, 23:87–115, 1987.

[133] P. Wolfe. The simplex method for quadratic programming. *Econmetrica*, 27:382–398, 1959.

[134] T.H. Yang and E. Polak. Moving horizon control of nonlinear systems with input saturation, disturbances and plant uncertainty. *Int.J.Contr.*, 58(4):875–903, 1993.

[135] L.A. Zadeh. From circuit theory to system theory. In *Proc. IRE, Vol. 50*, pages 856–865, 1962.

[136] E. Zafiriou. Robust model predictive control of processes with hard constraints. *Comp.Chem.Engng.*, 14(4/5):359–371, 1990.

[137] J.G. Zeiger and A.J. McEwen. Approximate linear realisations of given dimension via ho's algorithm. *IEEE AC*, 19:585–601, 1974.

[138] A. Zheng. Reducing on-line computational demands in model predictive control by approximating qp constraints. *Journal of Process Control*, 9(4):279–290, 1999.

[139] A. Zheng and M. Morari. Global stabilization of linear discrete-time systems with bounded controls - a model predictive control approach. In *ACC*, 1994.

[140] A. Zheng and M. Morari. Stability of model predictive control with mixed constraints. *IEEE AC*, 40:1818, 1995.

[141] Z.Q. Zheng and M. Morari. Robust stability of contrained model predictive control. In *ACC*, 1993.

[142] Y.C. Zhu. *Identification and MIMO Control of Industrial Processes, An Integration Approach*. PhD Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 1991.

[143] Y.C. Zhu and T. Backx. *Identification of Multivariable Industrial Processes for Simulation, Diagnosis and Control*. Springer Verlag, 1993.