INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

Université d'Ottawa • University of Ottawa

ACCELERATION FEEDBACK IN MODEL PREDICTIVE CONTROL OF ELECTROMECHANICAL DRIVE SYSTEMS

By
Maricel Ceru

Submitted
as part of the requirements for a
M. A. Sc. Degree

Supervisor
Dr. D. Necsulescu

March 2001

ABSTRACT

The continuously increasing demands of industries such as machine tools, robotics and, recently, auto industry for fast and accurate electromechanical position control systems have stimulated the interest in new, non conventional methods and approaches for control. The advancement of computing industry has provided also the tools for more precise control systems and the opportunity for the development of many different methodologies.

This thesis discusses the use of acceleration measurements in predictive control schemes for electromechanical positioning systems. The proposed method is intended to improve the tracking error while maintaining the controllability and the robustness of the controllers. The influence of acceleration feedback on controllers and observers of predictive type is evaluated, and the results are compared with those obtained by classical control schemes, without this type of feedback. Multiple approaches are subsequently compared, from the classical PID, Linear Quadratic Gaussian (LQG) control, to Dynamic Matrix Control (DMC) with state feedback and DMC with acceleration feedback (DMC/AF) as well. The simulations and the experiments are done using the framework of MATLAB/Simulink and dSPACE computer packages, on an electromechanical positioning system (EMPS) with friction wheel used at the Cologne Laboratory of Mechatronics (CLM). A controller design suite using GUI developed to facilitate industrial implementation has been used for rapid testing of diverse schemes. The observers for non-measured states were of Kalman filters type. The improvements of a

MPC (Model Predictive Control) with acceleration feedback were in reducing the positioning error, and in the disturbance rejection of internal friction and external disturbing torque.

The present analysis has continued a series of different solutions that improved gradually the positioning error. A detailed formulation of DMC and DMC/AF is done starting from a predictive control system, and it is compared with the same alternative considered with the MATLAB/MPC toolbox. Some of the problems encountered in the simulations and the experiments were known previously from other publications and were considered in this thesis with the goal of obtaining better results. The problem of friction was treated in the framework of an augmented state space model of the nonlinear mechanical system.

A general formulation of the use of predictive control schemes is presented, with focus on DMC method, and on the use of acceleration measurements. The direct acceleration feedback for a system represented by a state space model, for a system with state observer and a system represented by an augmented state space model to include the acceleration are also discussed. Disturbances in the acceleration signal (noise, torque ripple), time delays caused by sampled data implementation of controller/observer, or the non collocation of the acceleration and velocity sensors were examined in the simulation study.

CONTENTS

NOTATIONS AND SYMBOLS

A, B, C, D	Matrices of state-space mode
AF	Acceleration feedback
ARIX	Auto-Regressive Integrated eXogenous
ARIMAX	Auto-Regressive Integrated Moving-Average eXogenous
b_d	Drive-side viscous damping coefficient
b_{dl}	Spring material damping coefficient
b_l	Load-side viscous friction coefficient
CLM	Cologne Laboratory of Mechatronics
c_{dl}	String stiffness coefficient
c, d, m, o, p	Subscripts, abbreviations of the words control, disturbance, measurement objective and plant
DMC	Dynamic Matrix Control
DMC/AF	Dynamic Matrix Control with Acceleration Feedback Control
d	Time delay of the process in samples
d^	Time delay of the model in samples
$\underline{d}_{poc}$	Feedthrough matrix
$\underline{d}_{pod}$	Column vector of the corresponding feedthrough matrix from the disturbance process
EHAC	Extended Horizon Adaptive Control
EMPS	Electromechanical Positioning System
EPSAC	Extended Prediction Self-Adaptive Control
e(k)	Discrete white noise with zero mean

FIR	Finite Impulse Response
F_l	External forces
FSR	Finite Step Response
GPC	Generalized Predictive Control
GUI	Graphical User Interface
H_c	Control horizon
H_m	Minimum-cost horizon
H_p, (P, M)	Prediction horizon
i	Gear ratio of ball screw drive
i_a	DC motor current
i_{amp}	Current measurement signal
incr	Load-side angular displacement
J	Criterion (cost) function
J_d	Drive-side moment of inertia
J_l	Load-side moment of inertia
K	Matrix gain (LQG)
$k_{filt\,r}{}^T$	Reference filter gain vector
$k_{ff\,r}{}^T$	Feedforward gain vector for the reference model states
$k_{fb\,p}{}^T$	Plant feedback gain vector
$k_{ff\,d}{}^T$	Disturbance feedforward gain
k_Ω, k_α, k_φ	Stationary gains
k_{servo}	Servo amplifier gain
k_Ω	Velocity measurement gain

k_φ	Position measurement gain
k_α	Acceleration measurement gain
L	LQG gain matrix for extended functions
LQ	Linear Quadratic
LQE	Linear Quadratic Estimator
LQG	Linear Quadratic Gaussian Control
LQR	Linear Quadratic Regulator
MAC	Model Algorithmic Control
M_d	Driving motor torque
Mext	External torque acting on inertia
Mfrl	Load-side friction torque
Mfrd	Momentum of friction at the drive side
Mfrl	Momentum of friction at the load side
MIMO	Multi-Input Multi-Output
M_l	Input torque
Ms (=Mk)	Load side static (and kinetic) friction torque
N	Prediction horizon (MIMO) in MPC Toolbox
N_u	Control horizon in MPC Toolbox
n^(t +k | t)	Disturbance model used by DMC
n_X	Degree of a polynomial X
P_e	Optimal steady-state covariance matrix of the estimation error ($x - x_e$)

PCA	Predictive Control Algorithm
PFC	Predictive Functional Control
PI	Proportional-Integrative Control
PID	Proportional-Integrative-Derivative Control
P, Q_n, Q_d	Polynomials in the unified criterion function
Q, R	Weighting matrices in LQR
q	Forward shift operator, $q\,x(k) = x(k+1)$
q^{-1}	Backward shift operator, $q^{-1}\,x(k) = x(k+1)$
R, S, T	Controller polynomials
r_j	Weights for the constant control signal over the prediction horizon
S	Matrix solution for the Ricatti equation
SISO	Single-Input Single-Output
Sp	Set point
Td	Time delay of the process in seconds
Ts (ts)	Sampling period in seconds
Tservo	Servo amplifier time constant
t	Time
tacho	DC tachometer voltage
ts	Sampling period
UPC	Unified Predictive Control
$\underline{u}$	Lower bound level constraint
$\bar{u}$	Upper bound level constraint
u	Future controller output sequence, $(= [u(k), \ldots, u(k+H_p-1)]^T)$

u_{acc}	Load-side accelerometer voltage
u_{amp}	Control signal
u_{pc}	Control input to the plant
$u_{pc,k}$	Constant control signal
u_{pd}	Disturbing load-side torque at the input
u_r	Reference model input
u_{servo}	Servo amplifier input voltage
$u(t+k \mid t)$	Future control signals, $k = 0,1 \ldots N-1$
u_{tacho}	Output voltage
V, W	Design parameters (similar meanings as weighting matrices Q, R in LQR)
x_i	ith element of a polynomial X
$(x - x_e)$	Estimation error
$\underline{x}_d$	Disturbance model state at the current time instant k
$\underline{x}_p$	State vector of the plant
$\underline{x}_r$	Reference model state
y(k)	Process output at t = k
$y (t+k \mid t)$	Predicted output
$\underline{y}_{po}$	Vector of the objective outputs of the plant
$y(t+k \mid t)$	Value at instant t+k, calculated at instant t
$y\hat{}(t+ k \mid t)$	Predicted value of the output at t+k, calculated at instant t
$y\hat{}$	Predicted output of the process, $[y\hat{}(k+1),\ldots,y\hat{}(k+ H_p)]^T$
z	Complex variable from the z-transform
w	Desired output (reference trajectory), $[w (k+1), \ldots ,w(k+ H_p)]^T$

$w(k)$	Reference trajectory at time $t = k$
$\alpha_l = \dot{\Omega}_l$	Load-side acceleration
α_{max}	Maximum angular acceleration
$\varepsilon(k)$	Prediction error at $t = k$
Δ	Differencing operator, $\Delta = 1 - q^{-1}$
Δ_Ω	Constant of exponential decay (37%)
φ_d	Drive-side (motor) angular position
φ_l	Load-side (ballscrew) angular position
ρ	Weighting factor
Ω_d	Drive-side (motor) velocity
Ω_l	Load-side (ballscrew) velocity
Ω_{max}	Maximum angular velocity
ω	Frequency in rad/sec
$(.)^T$	Transpose operator

LIST OF FIGURES

I – INTRODUCTION

1.1 – Objective

In the last decade a very active area of research has tried to satisfy the continuous increase in demand for precise positioning systems. Industries producing tools, medical devices, cars or robots require better systems in terms of robustness and accuracy positioning. In this regard, other than improved classical control schemes, new, non-conventional methods became more and more present bringing continuously better performance. One of the new methodologies, among many others contained in advanced control methods (optimal, adaptive, fuzzy), is that of model predictive control. Model predictive control was developed in various forms: Generalised Predictive Control, Dynamic Matrix Control, Extended Prediction Self-Adaptive Control, and Predictive Functional Control being some of the best known of them. The present study will use the Dynamic Matrix Control approach to build control schemes for an electromechanical positioning system with friction compensation, given its simplicity and suitability for precise positioning applications. In high precision positioning, tracking accuracy requirements make friction compensation indispensable. Typical mechanisms with DC motors and ball screws pose always problems of friction, backlash or errors due to

measurement errors and delays due to signal processing. The ideal sensors mounted in the location of the physical quantity to be measured, with minimal noise and high precision, have still to be produced even though tremendous progress has been already reported.

The positioning systems have attracted already significant attention and numerous studies and experiments have introduced new methods for reducing the position tracking errors, the influence of internal and external disturbances, and for increasing the precision and the smoothness of the movements. One of the approaches considered was the use of acceleration feedback, a method of interest in non-linear systems in particular when friction is present. The method has been proposed on many occasions for different types of mechanisms, but not in conjunction with model predictive control. Compared to the classical methods of state feedback, and the Linear Quadratic Gaussian (LQG), the method showed better results in tracking errors and maintained the system robustness. The acceleration measurement was used directly, in an augmented state feedback to include acceleration feedback, and indirectly, through an observer, for DMC predictive type controller to improve the estimates of position and velocity. Both methods showed important gains in precision of the position and have not significantly reduced the robustness of the system.

The use of a GUI for the Matlab/Simulation application facilitated rapid testing of different controllers and allowed the investigation of the problem in continuos and discrete approaches. The development has been a logical follow-up of a previous work using predictive control schemes with only position and velocity as model states and with only these two state measurements. The investigation of the use of augmented state systems with multiple state derivatives in various predictive control schemes for non-

linear systems remains an area for further research. The main aspects of the method are discussed in part IV of the thesis, and a comparison with results obtained with Matlab/MPC toolbox is also carried out. The following section presents a literature review of the methods proposed in various publications and identifies the similarities and the novelties of the approaches in this thesis with respect to the previously proposed schemes.

1.2 – Literature Review

For precise positioning of an electromechanical drive system, an MPC controller has been developed using the position and velocity as outputs. This system was implemented with integrated tools for design, and showed good active damping of elastic modes present in most mechanisms, which resulted in higher control bandwidth [1]. The improvements in positioning errors have also been significant and, compared to a LQG approach, showed further improvement in tracking error and robustness. Both MPC and LQG showed improved performance when compared to classical schemes [2]. The compensation of friction disturbance by augmenting appropriately the plant model was also given in [2]. An approach for the design and implementation of Linear Quadratic Gaussian (LQG) control was presented and evaluated. With regard to the improvements in vibration damping and friction compensation, the overall result was a higher control bandwidth, good disturbance rejection and precise positioning when rapid changes in velocity and acceleration may occur.

The LQG approach has shown important improvements compared to the classical PI, PID control methods even though it was somehow more complex, and required better

theoretical knowledge and lead to difficult computational tasks. The efforts to find improved compensation for friction by using the acceleration measurements in different ways lead to interesting results as reported in [3] where an acceleration gain was calculated and used in a position-velocity-acceleration feedback control and was implemented. The search for better models of the friction disturbance on servomechanisms [4] opened also the way of using other system measurements, in particular the acceleration. Results have been obtained for various systems such as hard disk drives [4], [5]. An overview regarding the use of acceleration feedback is presented in [6].

Acceleration feedback was effective at low velocity regimes in the presence of significant friction and torque disturbances. The improvements outlined in [6] referred to tracking errors of the order of mm of positioning with an XY-table (two degree-of-freedom model). Another similar development refers to the use of joint acceleration feedback in controlling the contact transition of robot arms [7]. Some limitations of the method are present however: disturbances in the acceleration signal (noise, torque ripple), time delays caused by sampled data implementation of controller/observer, non collocation of the acceleration and velocity sensors, phase lag created by signal processing equipment etc.

Basic methodologies for nonlinear and predictive control are presented in [9] and [11]. Predictive control methodologies used in the present study are presented in [12]. A predictive controller presented in [1] used a particular form of predictive control, Dynamic Matrix Control (DMC), [10]. A generic analysis of the acceleration feedback

methods is given in [13]. The control of DC servomotors for high precision positioning is analyzed in [15], [16].

Friction compensation has also stimulated the interest in the use of acceleration feedback. This review of the literature on the subject revealed that the use of acceleration feedback has an important potential to increase the performance of controllers for precise positioning, especially for those systems with significant friction and low velocities. An overview of the organization of the thesis follows, indicating the novelty of the propose approach.

1.3– Thesis Overview

The present thesis is organised in four parts. The first part starts with a short introduction outlining the main objectives of the thesis and a literature review of works related to the methodologies using predictive control and the acceleration feedback in different systems. The part II introduces the physical system with the models for the overall mechanism, for the friction phenomena and with some classical control approaches following the presentation in [2]. Some basic results obtained using classical and LQG methods are reviewed for further comparison with those obtained using the predictive control and acceleration feedback approach. The electromechanical positioning system (EMPS) in the form used for experiments at the Cologne Laboratory of Mechatronics and the GUI used for rapid testing are presented.

A basic Model Predictive Control is developed in part III to give the general framework that supported this study and which was based on the DMC approach. Robustness, stability and the continuos/discrete approaches are also described. It is in the

part IV that the predictive controller from [1] is augmented with acceleration feedback and acceleration based observer. The comparison with a Matlab/MPC toolbox development for the same problem and the results in their simulation/experiments aspects are also presented. The acceleration measurement is used directly as acceleration feedback (AF) and indirectly through an observer, to reduce the positioning errors and to reduce the effect of disturbances due to friction and disturbing torque. The main objective was the evaluation of the improvements introduced by a DMC with AF controller compared to that of a MPC without AF. The conclusions in the last part, V, show the significance of the main results of the thesis and the possibilities of further developing non-linear systems controllers with augmented number of states.

II – THE PHYSICAL MODEL

2.1 – Electromechanical Positioning System (EMPS)

The electromechanical positioning system used as a testbed in this study is shown in Figure 1. It consists of a current-controlled DC motor and a linear positioning unit. A backlash-free ballscrew drive converts the rotary motion of the screw into the linear carriage displacement. Measurements used for control are the output voltage u_{tacho} of a DC tacho generator at the drive side, u_{acc} of an accelerometer at the load side, and the counter value 'count' of an incremental encoder at the load side with a resolution equivalent to 1.25 μm carriage displacement. The control signal is the input voltage u_{servo} for the motor reference torque. A compliant coupling between motor and positioning unit, whose stiffness and damping are only approximately known, produces a mechanical resonance at about 100 Hz. Another phenomenon present in this system is friction, which is dominant in the ballscrew drive and results from its preloading to avoid backlash.

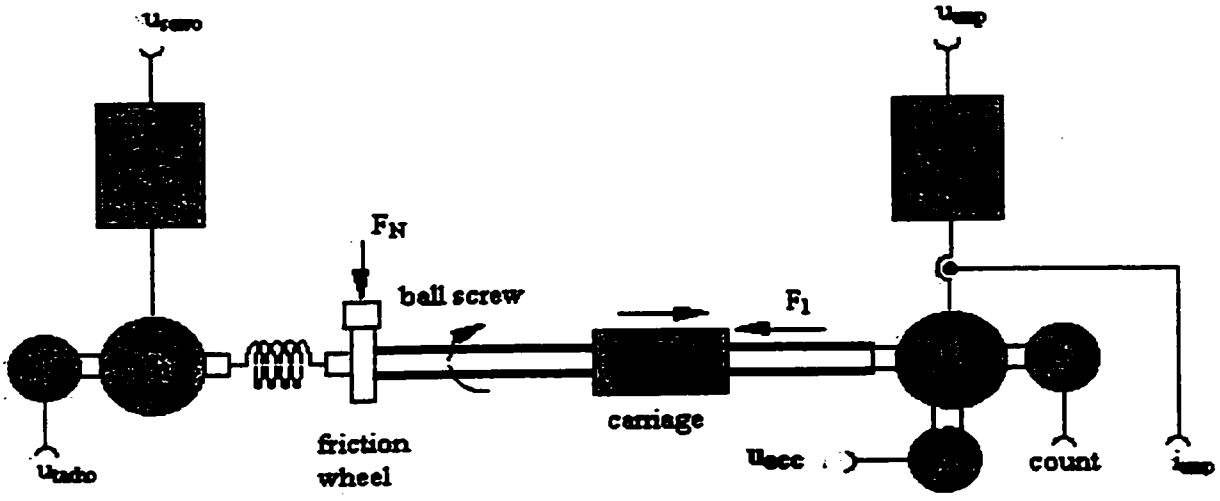


Figure 1: Electromechanical positioning system with friction wheel.

In order to investigate the performance of compensation schemes with varying friction conditions, a friction wheel can be used to increase this load-side friction. Extra shaft flexibility and the friction generated with the friction wheel permitted the evaluation of the relative performance of MPC versus LQG control. With the above structure, the EMPS incorporates all relevant properties of drive systems in machine tools, robotic and automation systems as well as of recent electomotor-driven actuators in automotive systems. Thus, the control approaches and results presented in this study can be considered relevant for industrial applications. The comprehensive nonlinear plant model has been developed to allow for realistic simulation of the controlled system i.e. for rapid prototyping [2] and it is presented in one of the subsequent sections. The model incorporates the dynamics of the current controlled servo amplifier, saturation of the reference current and supply voltage, amplifier noise, load-side friction, tachometer ripple and noise, and incremental encoder quantization. The whole model is also presented in the following sections to show all the characteristics of the system and of the model created to study it. The EMPS used in experiments is presented in figure 2a, and it includes an accelerometer mounted on the load side, on the carriage.

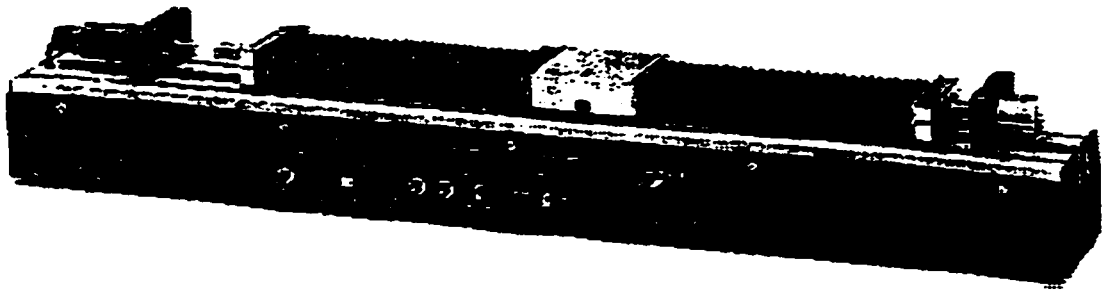


Figure 2a: Electromechanical positioning system (EMPS) used at CLM.

2.1.1 – Kinematics and Dynamics

The physical model, the mathematical equations of motion, the power and measurement electronics, a complete model in state-space form of EMPS, and the numerical parameters are all shown below:

- the mechanical model

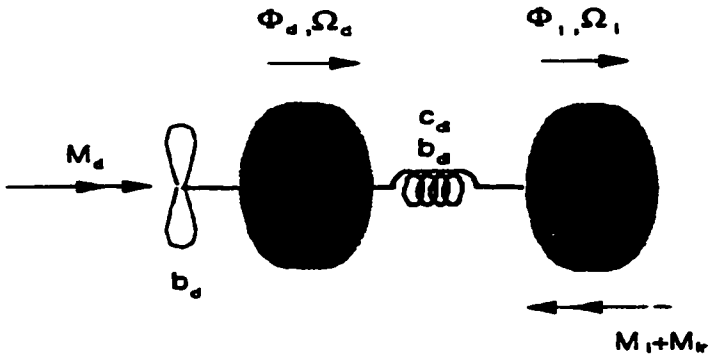


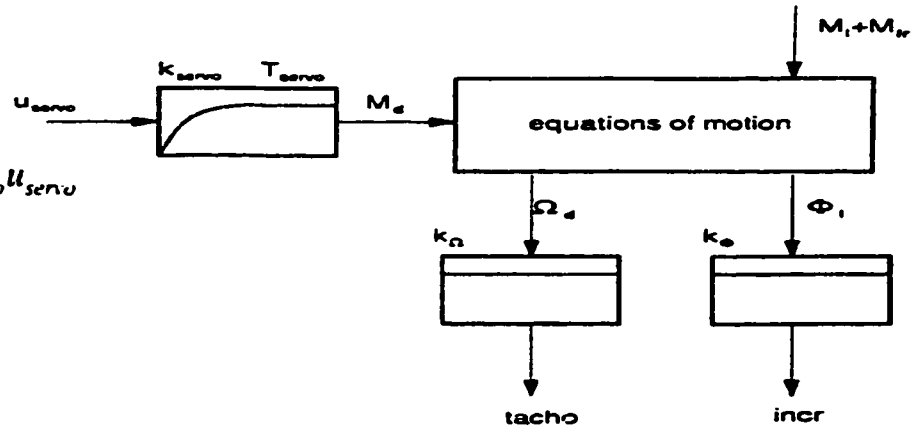
Figure 2b: Mechanical model of (EMPS).

- equations of motion [2] (2.1)

$$J_d\dot{\Omega}_d = c_{dl}(\Phi_l - \Phi_d) - b_d\Omega_d + b_{dl}(\Omega_l - \Omega_d) + M_d$$

$$J_l\dot{\Omega}_l = -c_{dl}(\Phi_l - \Phi_d) - b_{dl}(\Omega_l - \Omega_d) - (M_l + M_{fr})$$

- the power and measurement electronics [2]



$$T_{servo}\dot{M}_d + M_d = k_{servo}u_{servo}$$

$$tacho = k_{\Omega}\Omega_d$$

$$incr = k_{\Phi}\Phi_l$$

(2.2)

Figure 2c: The block diagram of the positioning system (EMPS) [2].

- the state-space model of the EMPS

$$\frac{d}{dt}\begin{bmatrix}\Phi_d\\ \Omega_d\\ \Phi_l\\ \Omega_l\\ M_d\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0 & 0\\ -\frac{c_{dl}}{J_d} & -\frac{(b_d+b_{dl})}{J_d} & \frac{c_{dl}}{J_d} & \frac{b_{dl}}{J_d} & \frac{1}{J_d}\\ 0 & 0 & 0 & 1 & 0\\ \frac{c_{dl}}{J_l} & \frac{b_{dl}}{J_l} & -\frac{c_{dl}}{J_l} & -\frac{b_{dl}}{J_l} & 0\\ 0 & 0 & 0 & 0 & -\frac{1}{T_{servo}}\end{bmatrix}\begin{bmatrix}\Phi_d\\ \Omega_d\\ \Phi_l\\ \Omega_l\\ M_d\end{bmatrix}$$

$$+\begin{bmatrix}0\\0\\0\\0\\ \frac{k_{servo}}{T_{servo}}\end{bmatrix}u_{servo}+\begin{bmatrix}0\\0\\0\\ \frac{-1}{J_l}\\0\end{bmatrix}(M_l+M_{fr})$$

$$\begin{bmatrix}tacho\\ incr\end{bmatrix}=\begin{bmatrix}0 & k_\Omega & 0 & 0 & 0\\ 0 & 0 & k_\Phi & 0 & 0\end{bmatrix}\begin{bmatrix}\Phi_d\\ \Omega_d\\ \Phi_l\\ \Omega_l\\ M_d\end{bmatrix}$$

- the matrix form of the state-space model [2]

$$\begin{aligned}\dot{x}_p &= A_p x_p + b_{pc}u_{pc} + b_{pd}u_{pd}\\ &= A_p x_p + [b_{pc} \quad b_{pd}]\begin{bmatrix}u_{pc}\\ u_{pd}\end{bmatrix}\\ &= A_p x_p + B_p u_p\\ y_{pm} &= C_{pm}x_p\end{aligned} \tag{2.3}$$

- the numerical parameters of the system

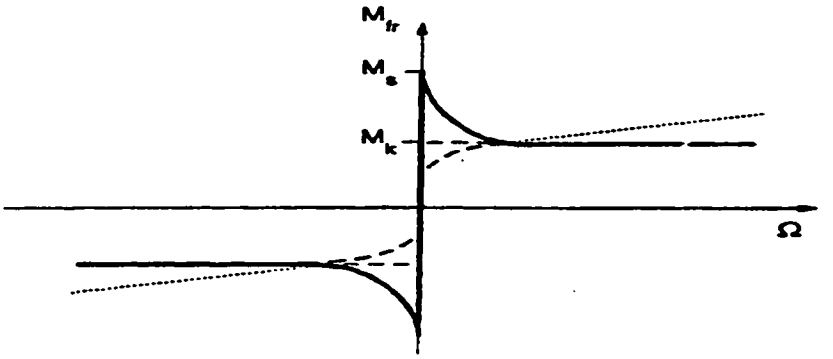
for linear model

J_d	drive-side moment of inertia	7.482E-6 kg m^2
b_d	drive-side viscous damping	4.716E-6 Nm/(rad/s)
c_{dl}	spring constant	0.885 Nm/rad
b_{dl}	dspring material damping	7.9E-5 Nm/(rad/s)
J_l	load-side moment of inertia	2.963E-6 kg m^2
k_{servo}	servo gain	0.1575E-1 Nm/V

k_Ω	velocity measurement gain	0.4966E-2 V/(rad/s)
k_Φ	position measurement gain tia	0.2429E-2 1/rad
for nonlinear model parts		
$M_s = M_k$	load-side static and kinetic friction torque	5.0E-4 sec
others		
i_s	gear ratio of ball screw drive	2π rad/2.5E-3 m
Ω_{max}	maximum angular velocity	604.0 rad/sec
α_{max}	maximum angular acceleration	15000 rad/sec^2
control and measurement signals		
u_{servo}	servo amplifier input voltage for motor reference current	+/- 10V
i_A	measurement output voltage for motor current, range +/-3A	+/- 3V
tacho	DC tachometer output voltage for motor--side angular velocity, rang +/- 604 rad/s	+/- 3V
incr	incremental encoder as a measure for the carriage displacement, range +/-0.16384 m (2.5 μm resolution) scaled to fractional number range	17 bits

2.1.2 – External Friction

The presence of friction is a source of steady-state errors and it requires compensation. The theoretical evaluation of friction was done and then experimental measurements have been used to indicate the influence of the different forms of friction. From the analytical expressions below, a nonlinear model for the friction was developed.



$$M_{fr} = \begin{cases} M_{fr}(\Omega) \cdot sgn(\Omega) & for \quad \Omega \neq 0 \\ M_{ext} & " \quad \Omega = 0 \wedge |M_{ext}| \leq M_s \\ M_s \cdot sgn(M_{ext}) & " \quad \Omega = 0 \wedge |M_{ext}| > M_s \end{cases}$$

$$M_{fr}(\Omega) = M_k + (M_s - M_k) \cdot e^{-\frac{|\Omega|}{\Delta_\Omega}}$$

where

M_{ext}	external torque acting on inertia
M_s	static friction torque
M_k	kinetic (Coulombic) friction torque
Δ_Ω	constant of exponential decay (to 37%)

Figure 3 shows the measurement of friction torque versus carriage velocity with dominant Coulombic and viscous friction. This characteristic represents the high friction level generated with the friction wheel used in [2] to show the relative performance of MPC versus LQG with respect to friction. For this high friction level the maximum absolute friction torque equals about 30% of the available absolute motor torque of 0.1575 Nm. A low friction level, without application of extra torque by the friction wheel, was used in [1] for the same purpose. For the low friction level case the maximum friction torque has been only about 10% of the available motor torque. The low friction level has been more suitable to demonstrate the active vibration damping properties of MPC versus LQG. Without compensation in the position-controlled system, the high friction causes a considerable error for the carriage position. Another reason for errors is

the external disturbance force F_l acting at the carriage. For testing purposes this disturbance force can be physically simulated by applying an equivalent torque $M_l = F_l / i_s$ to the load shaft, where i is the gear ratio of the ballscrew drive. The load torque can be generated by current control of the second load motor from Figure 1 with the control

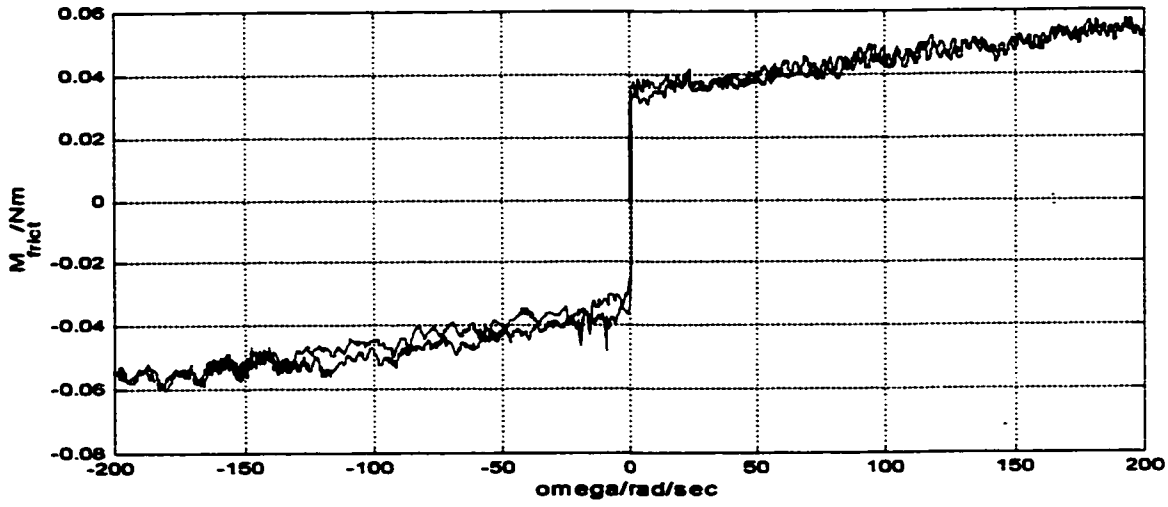


Figure 3: Friction torque versus carriage velocity (friction level increased by friction wheel).

signal u_{amp} and the current measurement signal i_{amp}. A comprehensive nonlinear plant model has been set up in [2] to allow for realistic simulation studies for the controlled system before a control is implemented by rapid prototyping for real-time testing. This model has been modified and implemented with predictive control schemes as presented in part IV of the thesis.

2.1.3 – Classical Control Approaches

The classical analysis was done comprehensively in [2] for the linear and non-linear positioning systems, with different types of controllers; the main results are presented here as the starting point for the further comparison with the predictive

control model results. The linear system was relatively easy to control using classical methods. To be close to the real plant, the non-linear model has been simulated in a complete closed-loop environment with controller, converters, signal conditioning and reference profile generator as shown in Figure 4.

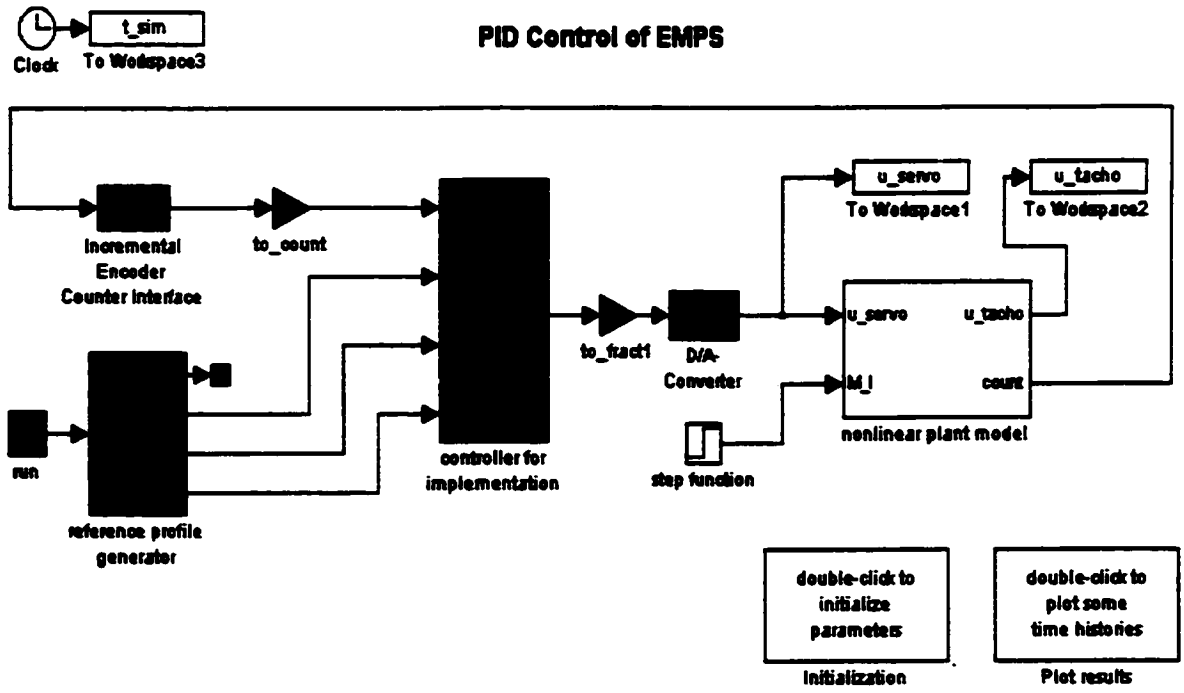


Figure 4: Classical control scheme for non-linear plant with controller for implementation, signal conditioning and reference profile generator.

The controller design is presented in Figure 5 and contains, for the non-linear model, feedback of position (incr), velocity (tacho), and feedforward of reference signal. A known PID controller is presented as it was developed in [2], and the performances obtained with these classical schemes in positioning are introduced through graphs for error vs. time, u_{servo}, u_{tacho}, vs. time and Bode plots.

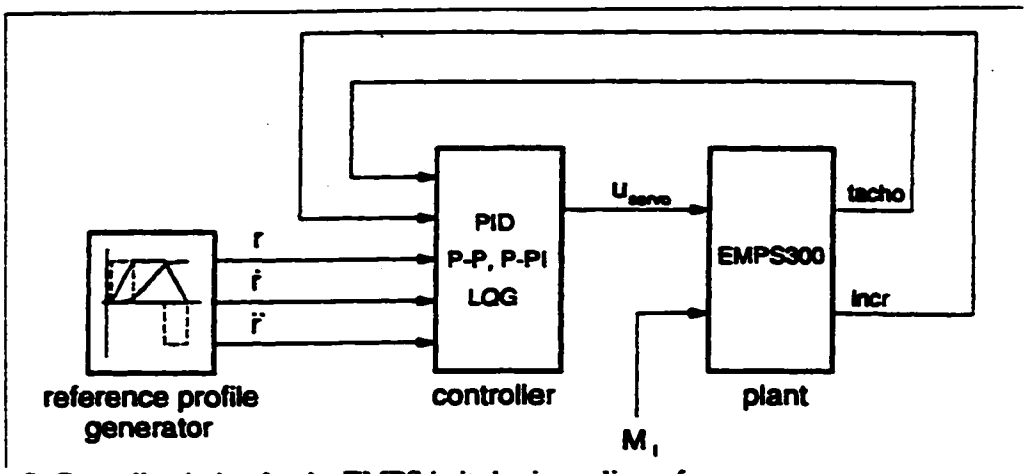


Figure 5: Controller design for the EMPS in its basic nonlinear form.

An example of controller structure, PID type, is presented in figure 6 with its equivalent Simulink structure and characteristic closed-loop Bode plots. The Matlab file created w ith the control toolbox is given in the Appendix.

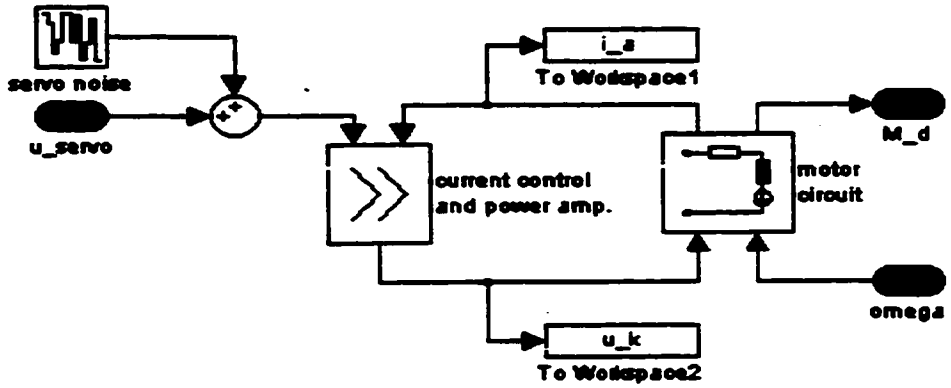


Figure 6a: PID controller structure and its expanded form.

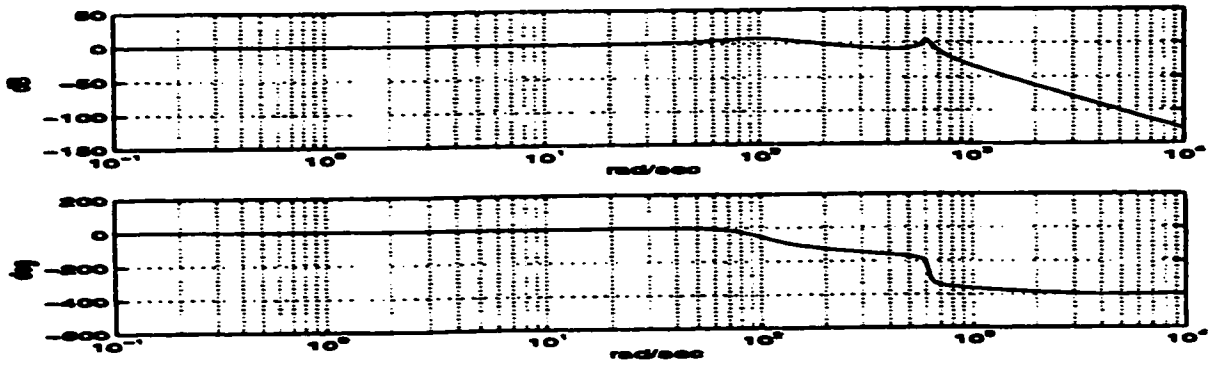


Figure 6b: Bode plots for the closed loop PID controller

Important improvements in the performances of the system were obtained with a LQG type control scheme that will be discussed in the next section. It is important to remark finally that the best positioning performance obtained with this classical method PID, shown in Figure 7 is compared to those obtained with the LQG method and finally with those from the predictive control methods. From the extensive analysis of the classical control methods in [2], it is important also to observe that all the schemes studied have been investigated also with the regard to the robustness criteria and the best results obtained have been further compared with those from the LQG method. In Figure 7 the transient position error reaches 70 μm.

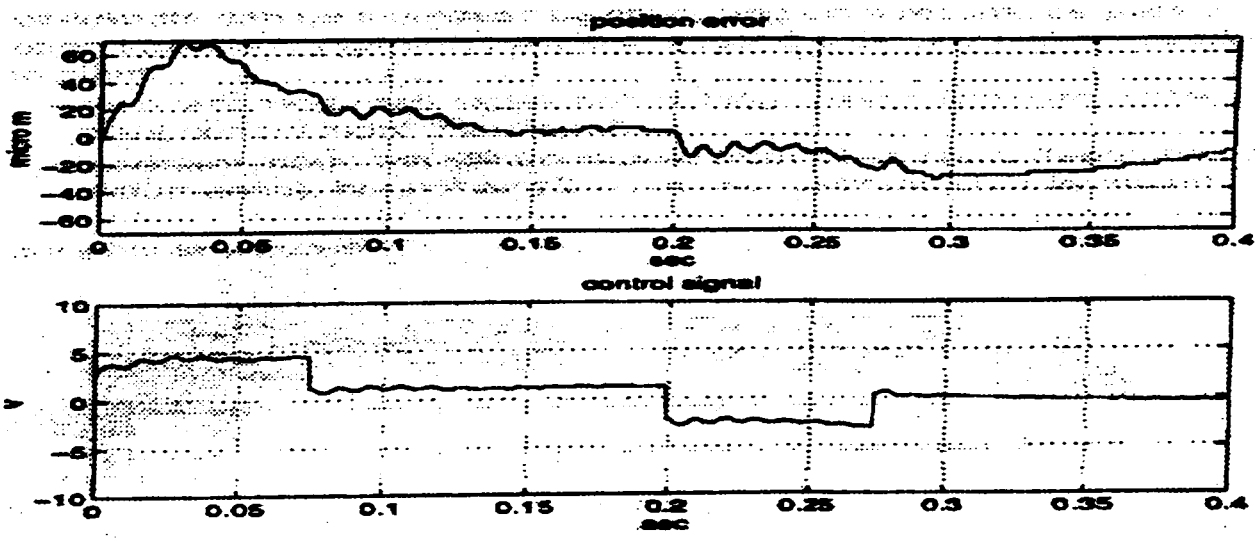


Figure 7: Simulated response of position error, and control signal u_{servo} for a closed loop system with classical PID control

2.2 – Linear Quadratic Gaussian Control (LQG)

This section is devoted to LQG control method because it constitutes an improvement over the classical PID control methods and proved to be reliable and easy to adapt despite having a more complex theoretical background and leading to more demanding computational tasks. For this reason it is important to compare MPC and MPC/AF not only with PID, but also with LQG. In [2] was introduced not only the mathematical and simulation developments but also the real time experiments with this type of controllers and later those with MPC against LQG in search for better positioning performance. The main characteristics of the method are described below.

The control schemes may include LQG compensators, regulators and estimators. The compensator could be a linear compensator for full state vector feedback (state vector of the plant augmented with exogenous variables to model reference/disturbance excitation and design objectives). Quadratic cost functions for compensator subsystems should be optimized, and the general conditions to apply the method include:

- design of constant state vector feedback/feedforward regulator (LQR) with all states measurable
- use of estimators (LQE) for unknown state variables and filtered noise measurements

In the case of EMPS the LQG analysis used exogenous variables introduced by excitation models (x,y) r,d,w for reference signals, disturbing torques and design objectives. A LQR model looked like that in figure 8.

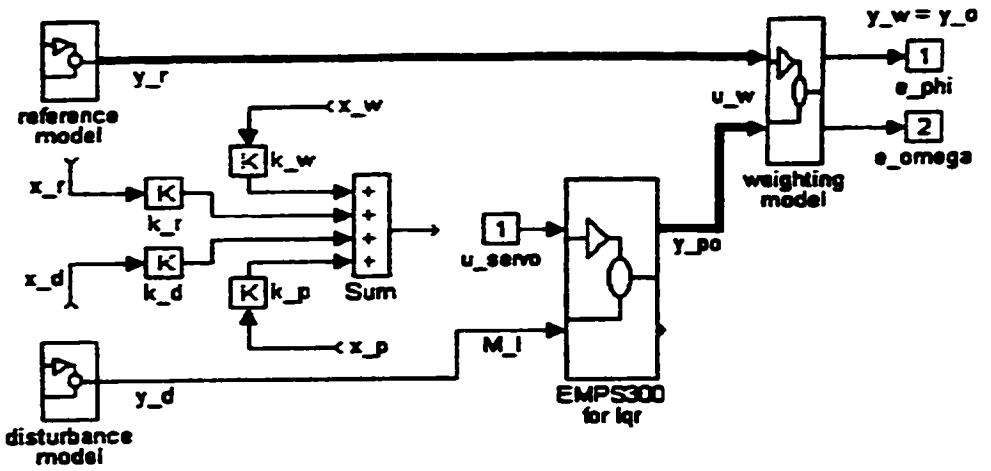


Figure 8: LQR (linear quadratic regulator) design.

For the LQR design problem the gains are calculated using a cost function J dependent on an output weighing matrix and a control weighing matrix R. The gain matrix has been calculated then as

$$K = -R^{-1} B^T S, \tag{2.4}$$

with S a matrix solution for the Ricatti equation:

$$SA + A^T S - S B R^{-1} B^T S + (C_0)^T Q_y C_0 = 0, \tag{2.5}$$

and A,B the deterministic and stochastic coefficients matrices representing the environment, and C, Q_y the control coefficients and weighting matrices.

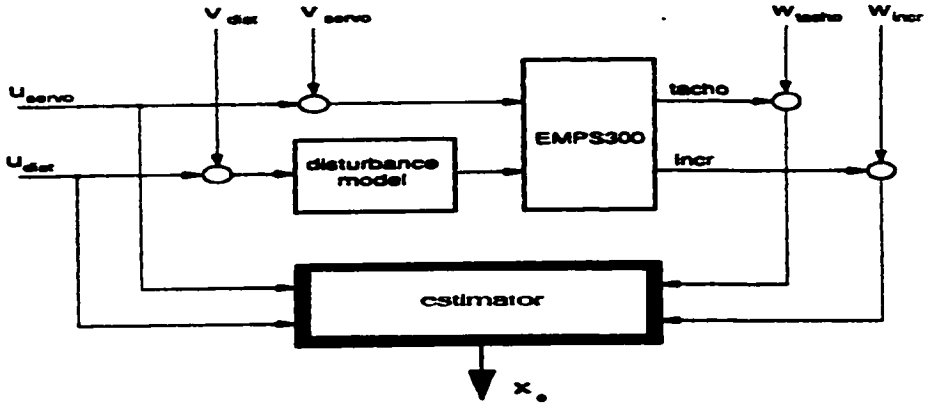


Figure 9: LQE (linear quadratic estimator) design.

The methodology with all its parameters has been implemented into a GUI application with Matlab/Simulink for rapid testing and implementation of different types of controllers including predictive types from a MPC suite [1]. It was used later for the simulations and testing of MPC/AF that are the main objective of the thesis. For the estimator (observer) the structure for the design is shown in Figure 9.

For the estimator design the following equation contains the state estimator, and the gain matrix:

$$dx_e/dt = (A - LC)\, x_e + Bu + W^{-1} P_e C^T y \qquad (2.6)$$

with P_e, the optimal steady-state covariance matrix of the estimation error ($x - x_e$) as the solution of the algebraic matrix Ricatti equation:

$$AP_e + P_e A^T - P_e C^T W^{-1} C P_e + BVB^T = 0. \qquad (2.7)$$

The matrices V and W are the design parameters and have similar meanings as weighting matrices Q, R in LQR. The basic blocks of LQG control, the estimator, the regulator and the compensator are assembled finally in a structure presented in Figure 10.

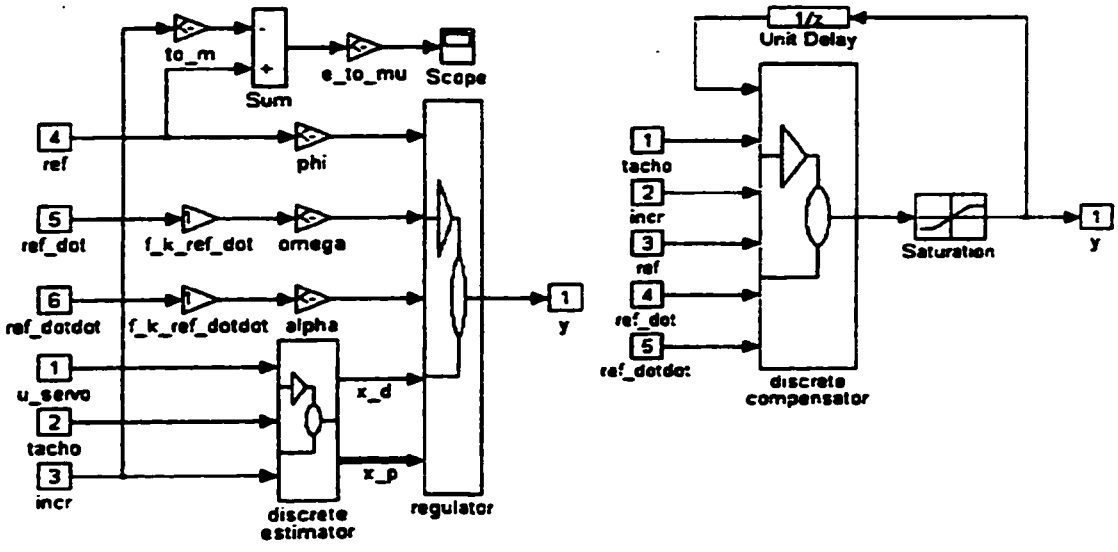


Figure 10: LQG compensator block diagram for implementation.

The stability analysis has shown the influence of the noise and the dependence of the estimator on the plant control input, and also evaluated the robustness of the system using open-loop Bode plots. The best results obtained with the LQG control in error tracking of positioning and the control signal are shown in Figure 11. The maximum transient position error is approximately 12 μm in this case. The LQG control has proven important advantages in rapport with the classical methods:

- active damping of resonant modes in the system to be controlled
- rejection of noise, quantization and ripple in the measurement signals up to the bandwidth
- increased robustness to parameter and structure variations

Limitations of the method are mainly the need for more knowledge of the theory and the environment, and more computational tasks involved in design, even though design tools support is very good.

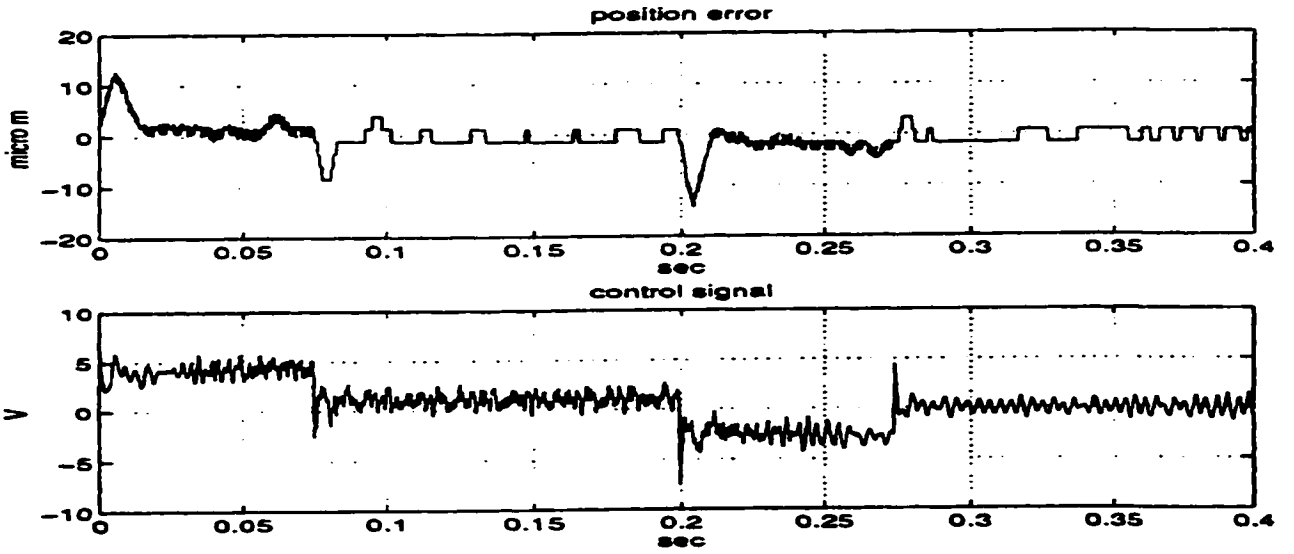


Figure 11: Experimental results for position error and control signal using a LQG compensator.

III - MODEL PREDICTIVE CONTROL

3.1 – General Formulation

The Model Predictive Control (MPC), or Model (Based) Predictive Control (MBPC) as it is known, has seen a tremendous development in the last decade after its introduction by [20] and [Cutler] in late seventies and the important contribution by [19] in late eighties. It is a very versatile methodology and constitutes one of the most general ways to solve problems of process control in the time domain. The MPC formulation overlaps with optimal control, stochastic control, control of processes with dead time, and multivariable control which are all characterized by the use of model-based development. One of the main aspects that facilitate the use of the method due to the finite control horizon used is the ability to handle constraints and non-linear processes. The same characteristic, i.e. the use of a finite control horizon, created also a difficulty in analyzing the stability and robustness for the systems examined. Most of the predictive control methods used in design for control of different processes have the following common characteristics:

- a horizon to predict the process output at future instants, based on the use of a model
- optimization (usually by minimizing) of an objective function through a calculated control sequence
- a receding development in the sense that, at any instant, the horizon is updated by moving forward in time and the first control signal of the sequence, calculated at each step, is applied

The MPC methodology contains many different forms or algorithms applicable to different types of problems and requirements, some of the most popular of them being the Dynamic Matrix Control (DMC), Model Algorithmic Control, Extended Prediction Self Adaptive Control (EPSAC), Extended Horizon Adaptive Control, Generalized Predictive Control (GPC). In this thesis the model used was DMC with acceleration feedback (DMC/AF) and it is presented in more detail in the next section. The most important advantages of the MPC method in general can be mentioned primarily as the ability to deal with multivariable cases, the use of feed forward to compensate for measurable disturbances, the simplicity in implementing a controller which is practically a linear control law. Also remarkable are its usefulness when future references are known (as in robotics), the compensation for dead times, applicability in a large variety of processes, and the openness to future extensions in the basic methodology. From the limitations, it should be mentioned that determination of the control law is more difficult than in the case of classical PID controllers and the stability and robustness characteristics of the systems cannot be easily evaluated due to the limited horizon used. The biggest drawback remains however the discrepancy between the real processes and the models on which the MPC is based. All MPC controllers have the characteristic strategy depicted in Figure 12, which could be explained in the following terms:

- for a determined horizon N (the prediction horizon), the future outputs are predicted at each instant t using the model; the predicted outputs y(t+k |t) depend on the known values up to moment t (past inputs and outputs) and on the future control signals u(t+k |t) for k = 0,1...N-1, calculated at the time t; the values y(t+k |t) and u(t+k |t) are the values at instant t+k calculated at instant t

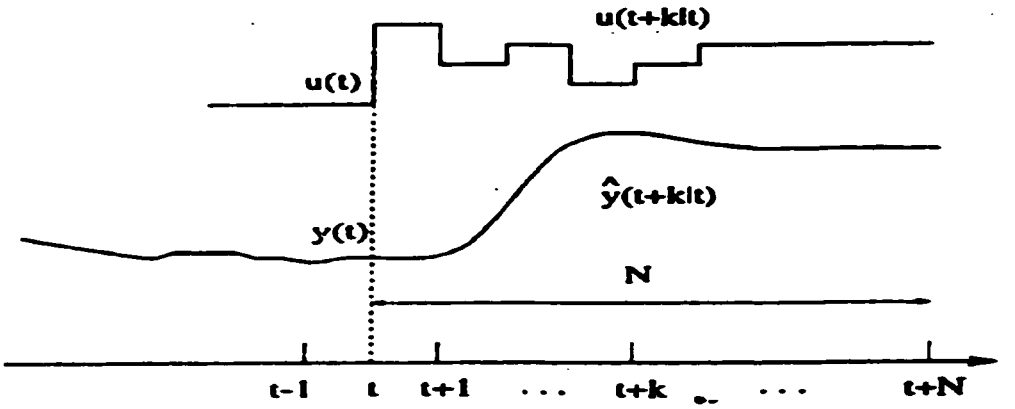


Figure 12: MPC Concept.

- the vector of future control signals is calculated by optimization of an established criterion for keeping the process the closest possible to the reference trajectory w(t+k); usually the criterion is a quadratic function of the errors between the predicted output signal and the predicted reference trajectory; an easy solution is obtained for a quadratic criterion with linear models, and without constraints; nonlinear models require iterative optimization; the control signal can be constant or variable in time
- the control signal u(t+k|t) is sent to the plant and the next control signals calculated are ignored because, for the next time step, y(t+1) is already known and the next step is repeated with the new value while all the sequences are updated; this way u(t+1| t+1) will be calculated, instead of u (t+1|t), because new information is collected for the receding horizon

The procedure is shown in Figure 13 and includes the structure of the signal interactions. For the receding horizon, most common is the use of predictive controllers in discrete time but there is some used also in continuous time. u(k), y(k) and w(k) are the controller output, the process output and the desired process output at sample k respectively.

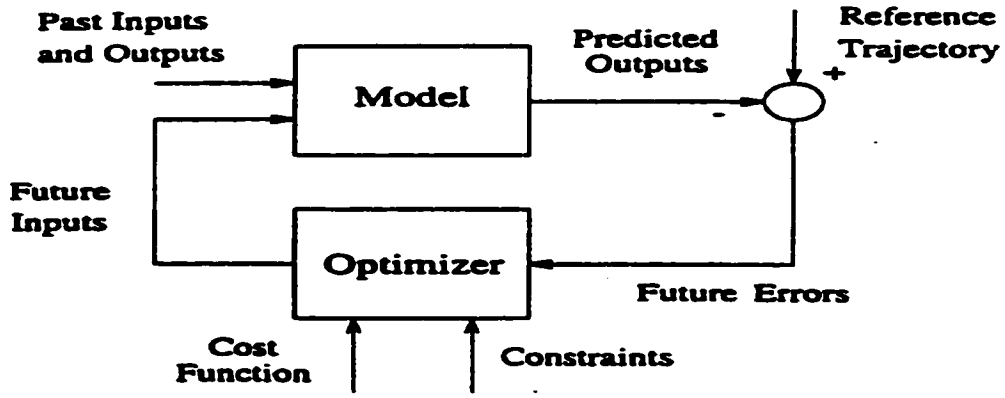


Figure 13: MPC basic structure [2].

The vectors

$$u = [u(k), \ldots, u(k+H_p-1)]^T$$

$$y\hat{} = [y\hat{}(k+1), \ldots, y\hat{}(k+H_p)]^T$$

$$w = [w(k+1), \ldots, w(k+H_p)]^T \qquad (3.1)$$

contain the future controller output sequence (u), the predicted output of the process (y^)and the desired process output (reference trajectory), while H_p is the prediction horizon (Figure 27). From the controller output sequence only the first element will be used (u(k)) to control the process. At the next instant (t+1), the whole procedure is repeated using the latest measured values. This is the principle of the receding horizon, as presented in Figure 14. This way, the predicted process output is continuously corrected for disturbances and modeling errors. The three characteristic elements of the MPC algorithms, prediction model, objective function, and the control law, are defined in various ways. In this study, only some characteristics connected to the DMC method will be presented in detail since the method has been employed for the study and the number of models used is too large to be discussed here in detail.

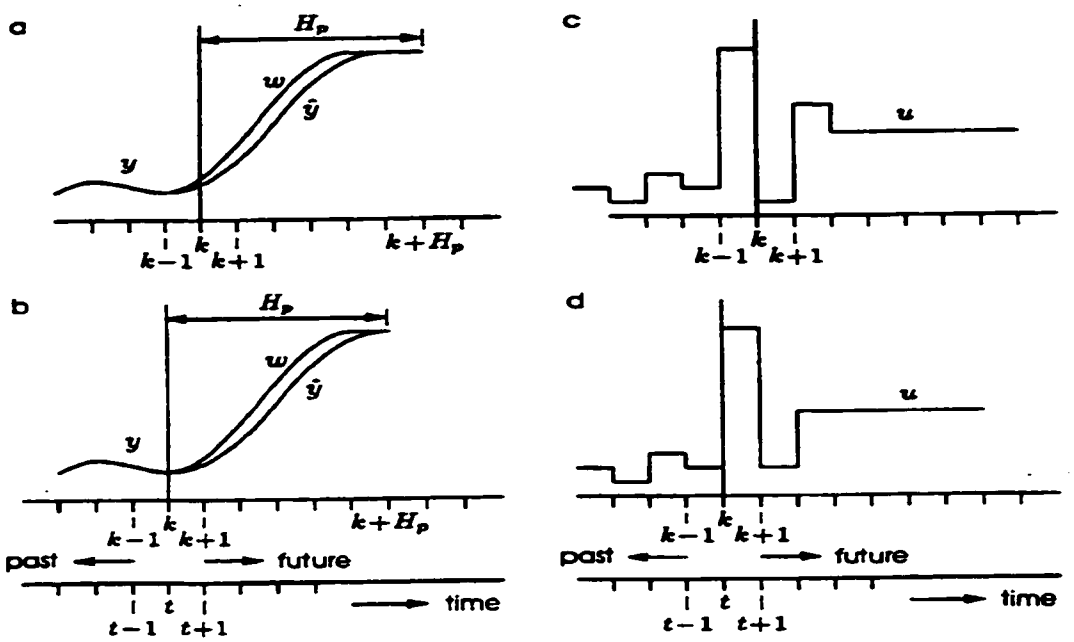


Figure 14: Receding horizon predictive control: in b and d the structure is at time t, in a and c it is at time t+1.

As part of the model-based class of controllers, predictive, LQ (Linear Quadratic), and pole-placement are some of the most popular and present many common features. The predictive and LQ controllers are both based on the minimization of a criterion function, usually called a cost function. Due to some very similar characteristics, it is worth presenting in parallel considerations regarding the predictive and LQ controllers. Then, the presentation continues with a short review of the way they relate to the control law derived with the polynomial approach and the unified predictive control methods. In many applications, LQ methods are based on linear state feedback or output feedback. As most representative, a state-space approach for a SISO discrete LQ control is presented in its basic form. The process is formulated as:

$$x(k+1) = A\,x(k) + B\,u(k)$$

$$y(k) = C^T x(k) \qquad (3.1.1)$$

where x(k) is the state of the process and A,B,C are matrices with process parameters. The criterion function to be minimized to determine the control law is used in the form:

$$J = \Sigma_{k=0,N} [x^T(k)\, Q\, x(k) + R\, u^2(k)] \qquad (3.1.2)$$

where N is the last sampling instant, Q is a weighting matrix and R is a weighting factor. Similar to the predictive control method, the criterion function in LQ control is minimized over a finite horizon. The fundamental difference is that the LQ method does not use a receding horizon. In LQ control the criterion function is minimized only once, at k = 0, and the result is that the controller is an optimal one with an output sequence used to control the process taken at every sample. Because the receding horizon is not used to recalculate at each step the new output values for the control signal, future disturbances and modeling errors are not taken into account. The controller output sequence determined for LQ control could be also generated with a linear state feedback in the form $u(k) = -K^T(k)\, x(k)$. Here the feedback vector K is variable in time even though the process is time invariant. In practice the solution for $N \rightarrow \infty$ is used because it was established that with the increase of k, K becomes constant. The final result is a time invariant feedback controller, which attenuates the effect of modeling errors and disturbances. Some optimization methods can be used further to take into account explicitly the disturbances by introducing them in the states and the output of the system. A similar problem is analyzed in chapter 4.2 for the predictive approach with acceleration feedback and extended state-space models including drive and load accelerations. In this regard, the optimization to follow usually uses a state estimator to obtain the best estimates from the observed outputs. The optimal state feedback law (in this case a feedback from the estimated states) will be determined by the minimization of the

criterion function (3.1.2) with the estimates replacing the real states. The use of $N \rightarrow \infty$ takes also the advantage of a closed-loop system guaranteed to be stable (assuming that the model and the process are identical). The feedback vector K in this condition is obtained by solving the algebraic Ricatti equation, as in chapter 2.2 where the LQG approach is also discussed. For a N-finite horizon, a controller output sequence can be calculated using dynamic programming to determine first the output u(N), then the value u(N-1), and following up to u(0). Once that the controller output sequence u(0),...,u(N) that minimizes (3.1.2) has been calculated, the minimization of the shorter form $J = \Sigma_{k=j,N} [x^T(k)\, Q\, x(k) + R\, u^2(k)]$, assuming that u(0),...,u(j-1) are known and there are not disturbances or errors in modeling, creates the controller output sequence u(j),...,u(N) equal to the one obtained by minimizing 3.1b. Using this property, a finite horizon LQ control can be established using a receding horizon framework almost as in predictive controllers, but here the prediction horizon is decreased by one with each sample instant as the time moves toward N. For the criterion function (3.1.2) if $Q = C^T C$ is taken as an example, the function J can be rewritten as:

$$J = \Sigma_{i=1,Hp} [y(k+i)^2 + Ru(k+i-1)^2 \qquad (3.1.3)$$

where H_p is used as the prediction horizon. In this particular example, very similar to the predictive controllers, an optimal controller output sequence is calculated over H_p but only the first controller output in sequence u(k) is used to control the process. At time t+1, an optimal controller output sequence is calculated again, over an interval with the length $H_p - 1$ keeping the end of the interval the same at time $t + H_p$. For the situation with no modeling errors and no disturbances, the controller output sequences calculated at

times t and t+1 are identical. If the prediction horizon goes to infinity the LQ and predictive approaches are practically identical.

For finite prediction horizons, the fundamental difference between the predictive control and finite horizon LQ control is the receding horizon aspect used with a fixed length prediction horizon by predictive controllers as opposed to a decreasing length optimization window for LQ controllers. The most common use of LQ controllers is, however, for an infinite prediction horizon, which results in a linear time-invariant state feedback control law. In this case the feedback vector is calculated by solving the algebraic Ricatti equation. Despite many similarities described previously, there are fundamental differences between the two methodologies which can be summarized as following:

- in the LQ approach, due to a preliminary established structure of the controllers and the calculations using a Ricatti equation, constraints on the controller output cannot be included, while in predictive controllers this is possible
- the LQ design is used for linear processes only, while predictive controllers yield notable results for nonlinear models predicting the process output
- the main advantage of the LQ methodology using an infinite horizon is that the closed-loop system is guaranteed to be stable, which for a finite horizon predictive controller cannot be guaranteed.

The other model-based controller mentioned earlier as similar to the predictive approach is the pole-placement design of polynomial controllers. It is used also mostly for discrete controllers and is characterized by the control law of the form

$$R\,u(k) = -S\,y(k) + T\,S_p \tag{3.1.4}$$

where R, S, and T are the controller polynomials in (q^{-1}) and S_p is the set point. The controller polynomials R and S are selected in such a way to allow the closed-loop poles to appear at the desired locations and T is calculated to insure that the closed-loop system behaves as designed to set point changes. The difficulty of the method is in determining the placement of the poles for the closed-loop system, i.e. in translating the design specifications in location of the poles and zeros for the closed-loop system. The problem is simplified greatly in the case of low-order processes with no zeros, but in general, with the controller outputs generated by (3.1.4), arbitrary reference trajectory and constraints on the controller output cannot be considered for the design. More than this, similar to the LQ approach, the pole-placement method can be applied only to linear or linearized processes. It is used in many occasions for issues common to the predictive control design and is also a model-based approach. The system model for polynomial control is usually given as:

$$y(k) = [q^{-d-1} B(q^{-1})/A\,(q^{-1})]\,u(k) + \xi(k) \tag{3.1.5}$$

where y(k) is the output of the process, u(k) is the input of the process, $\xi(k)$ is a disturbance acting on the process output and d is the time delay of the process in samples. The control law (3.1.4) and the process equation (3.1.5) form the closed-loop system in Figure 15

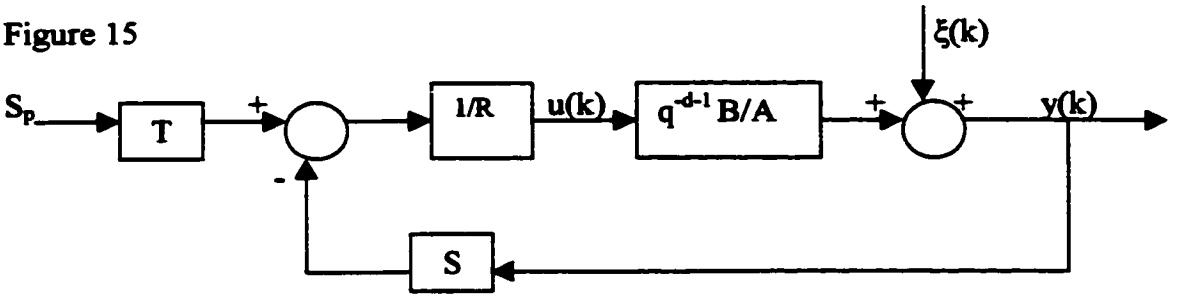


Figure 15: The closed-loop system used by pole-placement design method.

From the block-diagram, the output and control values at sample instant k are calculated with the formulas:

$$y(k) = [q^{-d-1} BT/(AR + q^{-d-1} BS)]\, S_p + [AR/(AR + q^{-d-1} BS)]\, \xi(k) \qquad (3.1.6)$$

$$u(k) = [AT/(AR + q^{-d-1} BS)]\, S_p + [AS/(AR + q^{-d-1} BS)]\, \xi(k) \qquad (3.1.7)$$

R and S are calculated from the Diophantine equation

$$AR + q^{-d-1} BS = P_d\, C \qquad (3.1.8)$$

where C is the observer polynomial (with roots inside the unit circle). T is calculated with

$$T = C(q^{-1})\, P_d(1) / B(1) \qquad (3.1.9)$$

For some well-known controllers, such as minimum-variance, moving-average, dead-beat or mean-level types, the values for P_d and T are given in tables.

The polynomial approach used to calculate the control law in pole-placement method controllers and in some predictive controllers is included in available software such as PFC CAD Software from Adersa or Scientific Systems Company, Inc. It provides fast solutions for different situations through relatively simple programs written in C or Matlab. A short description of the method as it is used in predictive control is presented below. The FIR and FSR models (FSR is used in DMC, the current model of the EMPS) are subsets of the models with the following transfer function type:

$$y(k) = [q^{-d} B(q^{-1}) / A(q^{-1})]\, u(k-1) \qquad (3.1.10)$$

where d is the time delay of the process in samples ($d \geq 0$) and A, B are polynomials of the following form

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_{nA} q^{-nA}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \ldots + b_{nB} q^{-nB} \qquad (3.1.11)$$

where n_A and n_B are the degrees of the polynomials A, B. The (3.1.10) model has the advantage of being used also for unstable processes. The most important advantages and disadvantages of the models using this type of transfer functions are

- the linear process is described by a minimal number of parameters
- processes stable and unstable can be described with these transfer functions
- the order of the process has to be known at least approximately
- the prediction of the output for the models is in general more difficult than that for FIR or FSR models

A predictive control law can be established as a unified control law that takes advantage of a series of properties described previously [11]. The process models differ slightly from (3.1.10) taking into consideration also disturbance model polynomials:

$$y(k) = [q^{-d} B(q^{-1}) / A(q^{-1})]\, u(k-1) + [C(q^{-1}) / D(q^{-1})]\, e(k) \qquad (3.1.12)$$

but here the degrees of the polynomials are chosen arbitrarily and the polynomials C, D representing the disturbance models are chosen by designer. For the models used by the well-known predictive controllers the polynomials A, B, C, D are given in tables and often can be chosen also arbitrarily. The unified predictive control law has the form

$$R\, u(k) = -S\, y(k) + T\, w(k + H_p) \qquad (3.1.13)$$

very similar to the pole-placement method (3.1.4). In fact the Figure 15 stays almost the same, the only difference being the replacement of S_p with $k + H_p$.

$$y(k) = [q^{-d-1} BT / (AR + q^{-d-1} BS)]\, w(k + H_p) + [AR / (AR + q^{-d-1} BS)]\, \xi(k) \qquad (3.1.14)$$

$$u(k) = [AT / (AR + q^{-d-1} BS)]\, w(k + H_p) + [AS / (AR + q^{-d-1} BS)]\, \xi(k) \qquad (3.1.15)$$

The polynomial approach in the unified predictive control law allows fast calculation of the polynomials described previously. With simple C or Matlab programs and fast

simulations of the processes in discussion, the design of predictive controllers with delays and constraints on inputs/outputs can be done properly. A block diagram for a predictive controller used by a linear system without constraints is shown in Figure 16

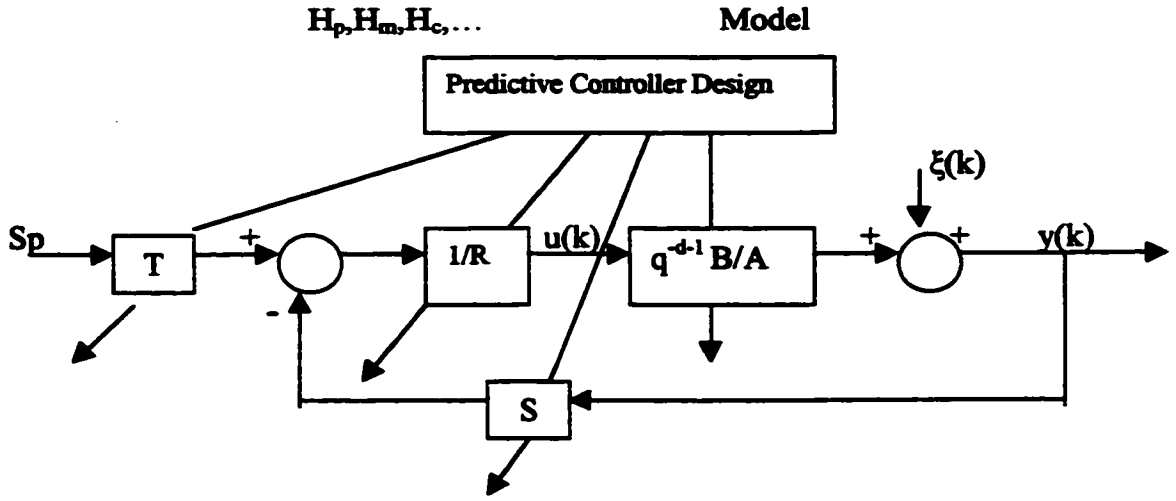


Figure 16: Predictive controller design for a linear system with no constraints.

As presented above, the main characteristics of predictive controllers consist of the model of the process, the criterion function to be minimized, the reference trajectory for the process output, and the minimization procedure itself. The i-step-ahead predictors are derived for different process and disturbance models including some of the most common of them FIR, FSR, transfer-function models and stochastic disturbances. The model used by DMC extensively applied in the EMPS application will be described in more detail in the next chapter and will be compared with some of the models the most used in practice. It was remarked also that a unified predictive controller design could be a fast solution for processes with transfer-function models and arbitrary degrees for the polynomials in the numerator and denominator expressions combined with stochastic disturbances. Another important remark is that the predictors have the structure of observers with predictions based on models corrected by the use of deviation between the process and the model output. Even though the disturbance models are difficult to

estimate and it is preferable that the designer of the control system imposes the parameters, some stochastic models recommended in [11] are good choices in increasing the robustness of the closed-loop systems. The second key parameter of the predictive control, the criterion function, has also a determining importance for the correct calculation of the controller output sequence. Using in-advance knowledge of the relationship between the desired process output and the controller output facilitates the minimization of the criterion function. A very useful result presented in the chapter was the possibility of calculating a control law in the form (3.1.13) with the polynomials involved being functions of the criterion parameters and of the model as it is presented in Figure 16. Practically, for the processes that are linear and without constraints, the pole-placement and the predictive control methods are identical. The difference consists in determining easier the design parameters for the predictive controller. Again, the advantage of using the predictive methods appears for nonlinear processes including input/output constraints that may be important for the systems in discussion. Even though a UPC was not used in the thesis, the properties and the way the calculations are done with this method are important and similar to the DMC method used on EMPS. Some other methods, well-known and frequently used in practice, are described in chapter 3.2.

3.2 – DMC Model Vs. Other MPC Approaches

The DMC controller presented earlier with its main characteristics and basic formulation is one of the most frequently used MPC approaches. Other approaches are, however, also used and main similarities and differences that make them appropriate for different design purposes will be presented. One of them, already discussed in the

previous chapter, is the UPC controller, which unifies many features from different types of predictive controllers. Figure 17 shows a table with choices for the design parameters that facilitate the minimization of the criterion function.

Controller	H_p	H_m	H_c	P	Φ	ρ	Q_n	Q_d	Controller	Model	A	B	C	D
DMC	√[1]	d+1	√	1	Δ	√	Δ	1	DMC	FSR	1	√	1	Δ
PCA	√	d+1	Hp-d	1	Δ	√	Δ	1	PCA	FIR	1	√	1	Δ
MAC	√	d+1	Hp-d	1	Δ	√	Δ	1	MAC	FIR	1	√	1	Δ
GPC	√	√	√	1	Δ	√	Δ	1	GPC	ARIMAX	√	√	√	AΔ
EPSAC	√	d+1	1	√	Δ	0	-	-	EPSAC	ARIMAX	√	√	√	AΔ
EHAC	√	H_p	1	1	Δ	0	-	-	EHAC	ARIX	√	√	1	AΔ

Figure 17: The criterion function parameters and process models used by some common predictive controllers.

In this table, the controller parameters represent: A, B, C, D polynomials selected by design function of the predictive method, H_p, H_m, H_c the prediction, minimum cost and control horizon, respectively, P a filter on tracking error, Φ a polynomial representing disturbances, ρ a weighting factor, Q_n and Q_d polynomials characteristic to the filters on controller output, and Δ the differencing operator ($\Delta = 1 - q^{-1}$). The types of controllers used are as follows: DMC - Dynamic Matrix Control, PCA - Predictive Control Algorithm, MAC - Model Algorithmic Control, GPC - Generalized Predictive Control, EPSAC - Extended Prediction Self-Adaptive Control, EHAC - Extended Horizon Adaptive Control. The predictive model mentioned represent: FSR - Finite Step Response, FIR - Finite Impulse Response, ARIMAX - Auto-Regressive Integrated Moving-Average eXogenous, ARIX - Auto-Regressive Integrated eXogenous.

All the predictive controllers listed in the table can handle processes with time delay, this being just one of the many reasons of their success in real-time applications.

[1] √ indicates that the parameter is a design one for UPC and the controller listed on the row

Each of these controllers will be characterized and will be compared with the DMC method, already introduced and used in the EMPS application. For the DMC methodology it is important to point out that:

- the use of FSR (Finite Step Response) model limits the method only to stable processes without integrators
- since $D\hat{} = \Delta$, $\phi = \Delta$, and $Q_n = \Delta$, for constant set points/ disturbances there are no steady-state errors; for non constant set points/ disturbances the steady-state errors can be determined [11]
- the disturbance model is considered $1/\Delta$, and the controller is optimal only for this type of noise
- because Q_n and Q_d are fixed, some frequencies in u(k) cannot be eliminated and sometimes are insufficiently damped such that unstable closed-loop systems may result
- the robustness of the DMC controller is influenced by the parameters H_p, H_c, and ρ allowing different settings which can bring a compromise between robustness and regulator behavior; this made successful many DMC applications
- it is necessary to define appropriately a reference trajectory

All these fundamental characteristics of the DMC method will be compared with those of the other methods. For the PCA controller the main characteristics are somehow different:

- it uses the FIR model for the process, limiting the applications to stable processes without integrators only (as in DMC case)

- steady-state errors occur in the same conditions as for the DMC method for the set points/ disturbances as in
- the same condition applies for the disturbance model fixed at the values in the table (Figure 17)
- an important difference from DMC is that PCA controllers do not use a control horizon and as a consequence it is not possible to select pole-placement, dead-beat or mean-level controllers; also ρ must be taken as a non zero value which makes difficult a selection for values known in advance
- the same condition on Q_n and Q_d determines the same consequence, i. e. the impossibility of eliminating certain frequencies in u(k)
- the reference trajectory is determined in the same way as in GPC method, using a first-order filter initiated at y(k); in same cases there may be difficulties with the generation of the reference trajectory

The MAC controller was introduced in late 70s by [20] and is been characterized mainly by the following:

- it is based on FIR models of the processes with predictions corrected by adding the term $y(k) - y\hat{}(k)$
- a major difference between a MAC controller and a PCA controller is the way the minimization of the criterion function is done: in PCA analytically while in MAC with an iterative optimization method

The GPC controller is related to a method that gained much popularity in the recent years and for this reason some fundamental characteristics are detailed next. The GPC (Generalized Predictive Control) method has been introduced by [19] in late 80s and

proved to be an excellent tool for applications in continuous and discrete time. It is based on models of the following type:

$$A(q^{-1})\, y(t) = B(q^{-1})\, u(t-1) + C(q^{-1})\, v(t) / \Delta \qquad (3.2.1)$$

where y(t), u(t), and v(t) are the output, the input, and the disturbance. A, B, and C are polynomials in q^{-1}, $\Delta = 1 - q^{-1}$ and C is a design polynomial (the observer polynomial). The control law is obtained by minimizing a criterion function of the following form:

$$J_{GPC} = \Sigma_{j=N1,\,N2} [y\hat{}(t+j) - w(t+j)]^2 + \lambda \Sigma_{j=1,\,Nu} [\Delta u(t+j-1)]^2 \qquad (3.2.2)$$

where $y\hat{}(t+j)$ is the predicted output at time t+j based on calculations at time t, w(t) is the set point, and N_1, N_2, N_u are the minimum prediction horizon, the maximum prediction horizon, and the control horizon. The parameters N_1, N_2, N_u, and λ are the design parameters for the GPC controller. The control law determined is:

$$\Delta u(t) = [1\ 0 \ldots 0]\, (G^T G + \lambda I)^{-1} G^T (w - f) \qquad (3.2.3)$$

where the corresponding matrices are given by:

$$G = \begin{bmatrix} g_{N1-1} \ldots g_0 \ldots\ldots\ldots 0 \\ g_{N1} \ldots\ldots\ldots g_0 \ldots\ldots 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ g_{N2-1} \ldots\ldots\ldots\ldots g_{N2-Nu} \end{bmatrix},$$

$w = [w(t+N1)\ w(t+N1+1) \ldots w(t+N2)]^T$

$f = [f(t+N1)\ f(t+N1+1) \ldots f(t+N2)]^T$

I, the unit matrix (3.2.4)

The elements of the matrix G are the step response values and f is the response of the open-loop system $B(q^{-1})/ A(q^{-1})$ to an input set at u(t-1). GPC, used in adaptive or non-adaptive forms, is characterized by the recursive least squares (RLS) estimator, the most popular estimator in this method. Models in the form

$$A(q^{-1})\, y(t) = B(q^{-1})\, u(t-1) + e(t) \tag{3.2.5}$$

with A, B polynomials in q^{-1} :

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots\ldots + a_n q^{-n}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \ldots\ldots + b_m q^{-m} \tag{3.2.6}$$

and e(t) is the noise or disturbance. The input/output data are filtered before reaching the estimator. The system model then can be written as:

$$A(q^{-1})\, y_f(t) = B(q^{-1})\, u_f(t-1) + e_f(t) \tag{3.2.7}$$

where $y_f(t)$, $u_f(t)$ and $e_f(t)$ are the filtered output, input and disturbance respectively. A filter of type $1/T(q^{-1})$, with $T(q^{-1})$ called the estimator filter polynomial, is selected and it determines in many regards the identification process. The equation (3.2.6) can be used also in the form of the linear in the parameters model form:

$$y_f(t) = \varphi^T(t)\, \theta + e_f(t) \tag{3.2.8}$$

where $\varphi(t)$ and θ are the data and parameter vectors in the following forms:

$$\varphi^T(t) = [-y_f(t-1) \ldots -y_f(t-n)\ u_f(t-1) \ldots u_f(t-m-1)]$$

$$\theta = [a_1 \ldots a_n\ b_0 \ldots b_m]^T \tag{3.2.9}$$

This way, then RLS uses the form (3.2.7), the unknown parameter vector θ being estimated in a recursive way. The estimates are expressed as:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\, \varphi(t)[\, y_f(t) - \varphi^T(t)\, \hat{\theta}(t-1)] \tag{3.2.10}$$

P(t) is the covariance matrix and has the expression:

$$P(t) = [1/\beta]\ \{\ P(t-1) - P(t-1)\, \varphi(t)\, \varphi^T(t)\, P(t-1)\, (\beta + \varphi^T(t)\, P(t-1)\, \varphi(t))^{-1}\ \} \tag{3.2.11}$$

The value for β, called the forgetting factor, is chosen usually higher than 0.95 but less than 1, and the initial value P(0) is taken $P(0) = \alpha I$, with α a big, positive scalar and I the unit matrix. When no advance knowledge of the system is available, the estimates $\hat{\theta}(0)$

could be taken equal to zero. The characteristics of the method, as described here with their main features, are important since the GPC has become a common tool for the recent applications in diverse scientific domains. The controller works with an ARIMAX model as presented in the table from Figure 17 and that implies some of the following defining properties for the method:

- both stable, and unstable processes can be modeled to be controlled
- the reference trajectory is usually obtained with a first-order filter initiated at y(k)
- the most important forms of noise to deal with, such as measurement noise, noise filtered by system dynamics or load changes, can be controlled
- both polynomials P and T from the criterion function and the noise model affect the behavior and the robustness of the closed-loop system
- as in the use of other controller types, the fixed values given to Q_n and Q_d make impossible to eliminate certain frequencies in u(k)
- the polynomial P is not explicitly taken into account in the criterion function but only indirectly; the polynomial T is used in both GPC and UPC to influence the robustness of the closed-loop system and useful choices for it in GPC may be $T = (1-\mu q^{-1})$ with default value $\mu=0.8$ or $T = A\hat{}(1-\mu q^{-1})$ with $D\hat{} = A\hat{} \Delta$ for well-damped processes

The other predictive controllers, EPSAC and EHAC, mentioned in the table, have also properties that make them useful in some particular applications. For the EPSAC controller the most important of these characteristics are:

- it uses a supplementary parameter λ, as in GPC criterion function, to weight future tracking errors, parameter not used in UPC approach; the influence of λ is similar to that of the minimum-cost horizon H_m ,which is not present in this method
- a major difference compared to UPC approach refers to the capability of dealing with the various noise models
- non constant reference trajectories and disturbances can produce steady-state errors
- because ρ is not considered in this approach, it is not possible to eliminate certain frequencies in the controller output
- compared to UPC, EPSAC controller is better for controlling unstable processes and is well-suited for well-damped and stable processes

The EHAC controller, introduced in middle 80s by Ydstie, did not contain a disturbance model originally and was lately modified and improved to eliminate steady-state errors in presence of constant disturbances. The most important characteristics of the model are described in short as:

- the minimum-cost horizon is equal to the prediction horizon, and no controller output weighting type ρ is used
- the criterion function to be minimized has the expression:

  $J = [y\hat{}(k+ H_p) - w(k+H_p)]^2$ that requires some conditions to insure unique controls, for example taking $H_c = 1$

- for well-damped minimum phase processes, H_p can be made equal to the desired response time of the closed-loop system, where H_p is the only parameter that can modify the robustness of the behavior of the controller

From the description of these predictive control approaches, some remarks could be made as to the particularities, differences and similarities encountered in presenting each method. It is observable that GPC and UPC approaches are very similar even though UPC controllers have some special characteristics:

- for most reference trajectories and disturbances, the steady-state errors can be eliminated
- selecting Q_n and Q_d, which are not fixed; in relation with $\rho > 0$, some undesired frequencies in the controller output can be reduced
- a pole-placement controller can be obtained directly using the polynomial P; the same P can be used to design specific behavior and robustness for the controller

The GPC and UPC controllers can be applied to unstable and improperly damped processes, and both can be considered to have as a subset the pole-placement approach. The other controllers are not well suited for this type of processes. The following properties characterize all the controllers mentioned above:

- all can control processes with time delays
- for constant reference trajectories and constant disturbances, the steady-state errors are not present
- attenuation of some frequencies in the output of the controllers is always difficult.

For the process model, the DMC approach has used the step response formulation:

$$y(t) = y_0 + \Sigma\Sigma_{i=1,N}[g_i \Delta u(t-i)] = y_0 + G(z^{-1})(1- z^{-1})u(t) \quad (3.2)$$

where gi are the output values for the step input and $\Delta u(ti) = u(t) - u(t-1)$. The value y_0 can be 0 without reducing the generality of the method. In this case the predictor is:

$$y\hat{}(t+k|\ t) = \Sigma\Sigma_{i=1,N}[g_i \Delta u(t+k-i|\ t)] \quad (3.3)$$

Some advantages of the method are: no prior information about the process is necessary, and complex dynamics of delays is easy to describe. Two important disadvantages are the truncated response with only N values and the large number of parameters used (normally N is in the order of 40-50).

The disturbance model used by DMC considers the values to be the same as at an instant t along the entire horizon:

$$n\hat{}(t+k \mid t) = n\hat{}(t \mid t) = y_m(t) - y\hat{}(t \mid t) \quad (3.4)$$

The predicted value of the output is:

$$y\hat{}(t+ k \mid t) = \Sigma\Sigma_{i=1,k}[g_i \Delta u(t + k -i)] + \Sigma\Sigma_{i=k+1,N}[g_i \Delta u(t + k -i) + n\hat{}(t +k \mid t)] \quad (3.5)$$

where the first term contains the future control to be calculated, the second contains the past values of the control actions and the last one represents the disturbance. The popularity of the method comes from the possibility of adding constraints, and that will introduce the following expression to the minimization process:

$$\Sigma\Sigma_{i=1,N}[Cy_i^j\ y\hat{}(t + k \mid t) + Cu_i^j\ u(t + k - i)] + c^j \text{ is negative for } j = 1\ldots N_c \quad (3.6)$$

The control horizon N_u and the prediction horizon N are values such that $N_u \leq N$. The main inconveniences of the method are the long computation effort and the inability to

deal with unstable processes. The typical criterion function used for the DMC approaches is a cost function:

$$J = \Sigma\Sigma_{i=1,Hp} (y\hat{}(k+i) - w(k+i))^2 \tag{3.7}$$

All the characteristics of the DMC methodology will be analyzed in the next section where a predictive DMC structure is extended from the design used in [1] to contain the acceleration feedback and the gains for the extended plant. The model is discussed with regard to stability and robustness and the performance is compared with the LQG control previously mentioned.

IV – EMPS/MPC APPROACHES

4.1 – Introduction

The search for improvements of the accuracy of an electromechanical positioning system resulted in the creation of an LQG suite and an MPC suite as Matlab applications with capabilities for rapid testing and implementing new types of controllers. In the first stage an MPC suite based on the DMC algorithm was introduced [1], which was developed for an SISO model and was implemented as a Matlab/Simulink application. Only position and velocity measurements were used. The promising results were better than those obtained with the classical methods and LQG control. The suite has subsequently been developed to include also acceleration on the load side. The software application has been built as a user-friendly interface with possibilities for rapid prototyping and testing and with real time experiments being possible through interfaces with dSPACE control package. The way the suite appeared has facilitated a large variety of tests and experiments for different models and approaches based on non-conventional

control methods. All the computations involved in the DMC algorithm have been implemented in the applications with different default values for the main parameters and multiple options concerning the characteristics of the algorithm. Later, the suite has been completed with a GUI (graphic user interface) for Matlab, predating the introduction of the specialized toolbox that came with the latest version of Matlab.

The simulations have been carried out by taking into account the friction disturbance through specific S-files and by using constant and variable control signals. More details about the main features of the application are given in the next sections where the simulation results are examined.

4.1.1 – Overview of Acceleration Assisted Control

All mechanisms, and especially those built for precise positioning, are affected in some degree by the nonlinearity of the processes involved, the presence of friction or other forms of disturbance or by the differences between the real processes and the models created to evaluate them for controlling purposes. In this regard, the presence of friction at low velocities, when friction models are not well defined and measurements may not be of high precision, creates important concerns over the performance of many servomechanisms including the type in discussion, EMPS. The search for reducing the effect of friction in these situations and for increasing the performances of different systems has opened the possibility for using acceleration feedback. Different types of controllers using the acceleration feedback have been proposed to improve the tracking performance and the robustness of positioning systems. The controller designed for EMPS in the present application is a predictive one and uses the acceleration measurements for feedback. As observed by different studies [6], the use of acceleration

signals can be done in different ways, the simplest being the direct use in a feedback loop to improve the tracking error, especially at low velocities.

An indirect use through an observer to obtain better estimates for position and velocity, or combined in different ways to include schemes using both direct and indirect use, are also valid options that have been tried on occasions with classical control schemes. The use of acceleration assisted control has been implemented on different mechanical systems from robotic arms to micro controllers and servomechanisms for precision positioning. In the case of EMPS, the advantages of the predictive control methods have been combined with the use of acceleration signal to obtain higher precision in positioning and a better rejection of vibrations. The acceleration signal has been used directly in a feedback loop and in an observer to improve the estimates of the position and velocity. In both cases the performance of the controller has increased significantly showing smaller errors in tracking the position profile imposed. In the case of using both approaches, combined in a configuration to include each aspect of the use of acceleration, the improvement is less significant compared to the contribution of each method used separately, even though simulations results have shown same range of accuracy improvement. In general, the most important factors that affect the performance for the controlled system can be presented as follows:

increasing performance

- the use of predictive control methods and the acceleration signal
- the use of acceleration through an observer or directly in the controller
- the presence of noise created by digital filtering the measurements

reducing performance

- disturbances in acceleration signal (motor torque ripple)
- phase lag introduced by the signal processing equipment
- time delay produced by the discrete models for the controller and observer
- the non collocation of position and acceleration sensors
- friction at low velocity
- measurement noise

In many applications, the acceleration feedback has been used to improve the tracking error, to reduce the level of vibrations, or to diminish the influence of friction. Some of the control schemes proposed by different authors are presented here with their main functions and characteristics. As presented in [9], a system to be controlled is evaluated with a m degrees-of-freedom model in the form of nonlinear equations of motion:

$$M(q,\theta)\,[d^2q / dt^2] + C(q, dq/dt, \theta)\, dq/dt + g(q, dq/dt, \theta) = f \qquad (4.1.1)$$

where $M(q,\theta)$ is the m x m positive definite inertia matrix with model parameters θ. The vector $C(q, dq/dt, \theta)\, dq/dt$ is the m vector of Coriolis and centripetal forces, $g(q, dq/dt, \theta)$ is the m vector of gravitational forces, Coulomb and viscous forces, and finally f is the m vector of generalized control forces. Each degree-of-freedom is associated with a motor for producing movement. The dynamics of some components (motors, sensors etc.) are ignored. Since the model refers to a manipulator, the controller consists of an approximate feedforward component (based on manipulator dynamics and a virtual reference trajectory), and a PD component with the control law:

$$f = M^{\wedge}(q)\,[d^2q_r / dt^2] + C^{\wedge}(q, dq/dt)\,[dq_r/dt] + g^{\wedge}(q, dq/dt) + K_v s \qquad (4.1.2)$$

where $M^{\wedge} = M(q, \theta^{\wedge})$, $C^{\wedge} = C(q, dq/dt, \theta^{\wedge})$, and $g^{\wedge} = g(q, dq/dt, \theta^{\wedge})$ are the same parameters from (4.1.1), with $\theta^{\wedge}$ an estimate of the model parameters, $dq_r/dt = dq_d/dt +$

$\Lambda q\sim$ a virtual reference trajectory, $s = dq\sim/dt + \Lambda q\hat{}$ a measure of the tracking accuracy, $q\hat{} = q_d - q$ the trackin error, and $q_d(t)$ the desired trajectory. The control parameters are K_v and Λ. The component $K_v s$ is a classical PD control described by:

$$K_v(dq\hat{} / dt + \Lambda q\hat{}) = K_v(dq\hat{} / dt) + K_p(dq\hat{} / dt) \text{ with } K_p = K_v \Lambda$$

The parameter s measures the tracking accuracy and should be used in the adaptive portion of the controller. In this presentation, the model is simplified to make explicit the acceleration signal influence. The control input signal (4.1.1) is modified by adding a term $(d\alpha/dt)$ s. With these parameters defined and the mentioned modification, some qualitative improvements can be already signaled in the following form:

- the parametric uncertainty influence on performance is reduced by a factor estimated at $1+\alpha / \beta$, where β is the gain margin
- a large value for α is desirable to improve the tracking performance, but possible gain through α may be just cancelled out by the level of noise in acceleration signal expressed as $n_{(d2q/dt2)}$
- the relative error in acceleration measurement Δ_r has the same influence on tracking performance as the disturbance signal at the input of level $\alpha\Delta_r/(\alpha\beta+1)$; for a small α the influence is not important but for large values of α it is proportional $t\Delta_r / \beta$; Δ_r should be kept small by insuring good conditioning of the acceleration signal (high quality sensors, filters);

The control signal (torque in this case) from (4.1.2):

$$f = M\hat{}(q)(v + d^2q_d / dt^2) + C\hat{}(q, dq/dt)(dq/dt) + g\hat{}(q, dq/dt) \qquad (4.1.3)$$

with v the output of a linear controller (in this case a PD controller) is modified further to become:

$$f = M\hat{}(q)(\beta I (v + d^2q_d / dt^2) - \alpha I (d^2q / dt^2) + C\hat{}(q, dq/dt) (dq/dt) + g\hat{}(q, dq/dt) \quad (4.1.4)$$

If the relative uncertainty $\gamma = \max_{q,\theta} \| M^{-1}M\hat{} - I \|$ in the mass matrix satisfies the condition $\gamma < 1$, and $M\hat{}$ is selected appropriately, while α is satisfying stability requirements, then the following form for β becomes optimal (largest reduction of errors influencing $M\hat{}$ on the closed-loop system):

$$B = 2 [(1+ \gamma)/(1+ \alpha(1+ \gamma)) + (1- \gamma)/(1+ \alpha(1- \gamma))]^{-1} \quad (4.1.5)$$

An even easier choice may be $\beta = 1+ \alpha$ which relates to (4.1.4) when $\gamma \rightarrow 0$; it is visible if the series expansion of β referring to γ is developed:

$$\beta = 1+ \alpha + \alpha \gamma^2/(1+\alpha) + 0(\gamma^4)$$

The introduction of the acceleration allows evaluating the equation error directly by using the measurements into the model equation. The resulting residue can be reduced using the acceleration feedback in different ways, as for example in the form presented in [9]. Equation error can be defined as:

$$e = M\hat{} (q_m) [d^2q_m / dt^2] + C\hat{}(q_m, dq_m/dt) dq_m/dt + g\hat{} (q_m, dq_m/dt) - f_m \quad (4.1.6)$$

where q_m, dq_m/dt, d^2q_m/dt^2, and f_m are measurements replacing q, dq/dt, d^2q/dt^2, and f.

To reduce the equation error, the acceleration is used as an additional input to the controller. If the output of the new controller can be expressed as a linear combination of the output of the first controller and the acceleration, then the equation error can be reduced. The control signal $f = f (q, dq/dt, t)$ can be then extended to

$f^* = f^*(q, dq/dt, d^2q/dt^2, t)$ when measurements for acceleration are available. With acceleration included in feedback law as following:

$$f^*(q, dq/dt, d^2q/dt^2, t) = (1+\alpha) f(q, dq/dt, t) - \alpha (M\hat{}(q) [d^2q_r/dt^2] + C\hat{}(q, dq/dt) [dq_r/dt] + g\hat{}(q, dq/dt)) \quad (4.1.7)$$

the equation error can be reduced to [9]

$$e/1+\alpha \quad (4.1.8)$$

This shows that a large value for α could reduce significantly the equation error. Some inconvenience with this development is that the acceleration is often affected by noise and is used in the feedback with some delay, and these conditions introduce some limitations with regard to the possible choices for α. Using some of the relations determined above in designing the parameters of the controller, feedback laws and control laws may be determined for predictive methods. However, methods for calculating acceleration feedback gains or for designing acceleration feedback schemes in more complex control systems are not yet available. This approach for dealing with the acceleration assisted control presents some very attractive features, especially for non-linear processes and for systems greatly affected by friction. The choice of α and the limitations in selecting values for it remain problems to be solved. In general, increasing the value of α will increase the differences between the results with the theoretical model and the real process. The influence of the noise and the disturbances of other nature, present in the acceleration signal, as well as some other influences mentioned at the beginning of the chapter, contribute to the difficulties encountered for these systems, even if in general the use of the acceleration feedback can improve the performance of the control system. It is also important to remark that the increase of the sampling rate can also improve the tracking performance, and has to be compared to the effect of the acceleration signal.

Most of these observations have taken into account the use of acceleration directly, in feedback schemes as related by the first approach, i.e. direct acceleration feedback. When the acceleration signal is used in an observer to improve the estimations, particularly for the velocities, some remarks also should be made accordingly. The high accuracy optical encoders used presently provide very precise position estimates and, consequently, most of the errors remain with the velocities estimation and in the predictions made in MPC. It can be stated that in principle, the use of the acceleration signals seems to be more profitable in the second approach, i.e. with an observer.

Another aspect of the importance and the possible improvements with acceleration feedback method is presented in [7], where it proves to increase sensibly the performance of a simple single-link manipulator. This refers to the introduction of the joint acceleration feedback into an integral force controller with velocity feedback. Similar results can be found also in [17] where acceleration feedback gains are calculated to apply the acceleration measurements and to increase the precision of movements for servomechanisms with friction. For the single-link manipulator a contact transition control was design to include velocity and acceleration feedback. The dynamics of the manipulator is given by:

$$J\, d^2\theta/dt^2 + \tau_e + \tau_n = \tau_a \qquad (4.1.9)$$

where J, $d^2\theta/dt^2$, τ_n , τ_a , τ_e are the inertia for the link and the joint acceleration, disturbances, actuated torque, and reactive torque determined by the environment. The reactive torque τ_e depends on the reactive force over the link f_e in the form:

$$f_e = \begin{cases} 0, & \text{for } x < 0 \\ B_e\, dx/dt + k_e\, x, & \text{for } x \geq 0 \end{cases} \qquad (4.1.10)$$

where B_e, k_e are the damping factor and the stiffness of the contacted environment, and x is the displacement of the environment surface. The reactive dynamics of the manipulator is discontinuous around x = 0 when the velocity dx/dt is not zero, and the kinetic energy has to be dissipated to establish a stable contact. Because the control law given by the active force may not absorb the energy efficiently, during the contact transition period oscillations or even instability may occur [21]. If oscillations are important, they depend on the resonance of the open-loop frequency response of the force control:

$$f_f(S) / \tau_a(S) = k_s (B_e S + k_e) / l(JS^2 + B_e S + k_e) \qquad (4.1.11)$$

where f_f (= k_s f_f) is the sensed force and k_s is the gain of the force sensor. A force control has to be capable to adapt to various environmental contacts. The presence of oscillations is not desirable and should be damped appropriately. In this regard a velocity feedback portion can supplement to the force control law, as for example:

$$\tau_a = k_f \, l \, (f_d - f_f) - k_v^* \, d\theta/dt \qquad (4.1.12)$$

where f_d represents the desired force, k_v and k_f are the velocity feedback and the force control law. If the manipulator interferes with the environment at time t = 0, and has a non zero velocity $d\theta_0/dt$, then the kinetic energy of the system at any time after (t > 0) is:

$$E_i(0) + E_a(t) = E_c(t) + E_r(t) \qquad (4.1.13)$$

with E_i the initial kinetic energy, E_a, E_c, and E_r being the actuating work, the work done by the reaction force, and the kinetic energy at time t. Their expressions are:

$$E_i = J \, (d\theta_0/dt)^2 / 2$$

$$E_a = \int_{0,t} \tau_a(t)(d\theta/dt)dt$$

$$E_c = \int_{0,t} l \, f_e(t)(d\theta/dt)dt$$

$$E_r = J \, (d\theta/dt)^2 (t) / 2 \qquad (4.1.14)$$

If f_d is constant and the initial velocity becomes 0 at $x = x_e$, at time $t = t_e$ (4.1.9)-(4.1.11) give:

$$J\,(d\theta/dt)^2(t)/2 = J\,(d\theta_0/dt)^2/2 + k_f\,f_d\,x(t) - + k_s\,k_f\,k_e\,x^2(t)/2 - E\hat{}_{ke} - E\hat{}_{be} - E\hat{}_v$$

$$E\hat{}_{ke} = k_e\,x^2(t)/2, \quad E\hat{}_{be} = (k_e\,k_f + 1)\,l^2\,B_e\,x\hat{}(t), \quad E\hat{}_v = k_v\,x\hat{}(t),$$

$$x\hat{} = \int_{0,t} (d\theta/dt)^2\,dt > 0, \qquad \text{with } 0 < x \le x_e,\ 0 < t \le t_e \qquad (4.1.15)$$

where $E\hat{}_{ke}$ is the potential energy possessed by the environment, $E\hat{}_{be}$ is the energy dissipated by the environment, and $E\hat{}_v$ is the energy consumed by the velocity feedback. The velocity feedback factor operates in the same way the damping factor B_e does. In principle any energy that may provoke the oscillations should be dissipated in a time period $[0, t_e]$ through the increase of k_v. This gain k_v is frequency weighted, and under some restrictions, affects high frequencies and sensor noise and can maintain stability of the closed-loop system. That implies that the velocity feedback operates with some efficiency within a limited bandwidth of ω_{cv} which limits the capability of $E\hat{}_v$ to respond to the oscillations of high frequencies.

In the case that the frequency of oscillations has been determined using (4.1.9), $E\hat{}_v$

$$\text{becomes } E\hat{}_v = \begin{cases} |k_v(j\,\omega_r)|\,x\hat{} \approx |k_v(j0)|\,x\hat{}, & \text{for } \omega_r \le \omega_{cv} \\ 0, & \text{for } \omega_r > \omega_{cv} \end{cases} \qquad (4.1.16)$$

where $|k_v(j\,\omega_r)|$ represents the gain of the velocity feedback at frequency ω_r, and ω_{cv} is the bandwidth of $k_v(j\omega)$. This shows that the velocity feedback cannot affect the oscillations at high frequencies but is effective at frequencies lower than ω_{cv}. The inclusion of acceleration feedback has the objective to increase the control performance.

A linear accelerometer, placed on the manipulator, change the force control law as follows:

$$\tau_a = k_f\, l\, (f_d - f_f) - k_a \int_{0,t} (d^2\theta/dt^2 + k_v\, d\theta/dt)\, dt \qquad (4.1.17)$$

where k_a is the gain for the acceleration feedback control. Using this control law in (4.1.11),(4.1.12) a term that represents the energy consuming factor can be calculated as:

$$E\hat{}_{av} = k_a\, x\hat{} + k_v\, k_a\, x^2 / (2\, l^2) \qquad (4.1.18)$$

with $x\hat{} = \int_{0,t} (d\theta/dt)^2 dt$ (4.1.19)

At a frequency ω_r the component representing the consumption of energy to reduce the oscillations can be written in the form:

$$E\hat{}_{av}(\omega_r) = |k_a(j\omega_r)|\, x\hat{} + |k_a(j\omega_r)|\, |k_v(j\omega_r)|\, x^2 / (2\, l^2) \qquad (4.1.20)$$

The acceleration feedback gain k_a has an effect similar to the velocity feedback gain k_v, but the bandwidth of the acceleration feedback can be much higher than that of the velocity feedback, that is $\omega_{cv} << \omega_{ca}$ where ω_{ca} is the bandwidth of $k_a(j\omega)$. Finally, the term $E\hat{}_{av}(\omega_r)$ becomes:

$$E\hat{}_{av}(\omega_r \approx \begin{cases} |k_a(j0)|\, x\hat{} + |k_a(j0)|\, |k_v(j0)|\, x^2 / (2\, l^2), & \text{for } \omega_r \le \omega_{cv} \\ |k_a(j0)|\, x\hat{}, & \text{for } \omega_{cv} < \omega_r \le \omega_{ca} \end{cases} \qquad (4.1.21)$$

In this particular example, the introduction of the acceleration feedback have increased the capability of the robot to eliminate the contact oscillations within a higher bandwidth $\omega_{cv} < \omega_r \le \omega_{ca}$. The classical integral control scheme, used for the force to control the manipulator, is able to handle contact transitions by taking into account joint velocity and acceleration feedback [22]. The loops in the feedback scheme behave practically as robust dampers and contribute to the elimination of the oscillations at contact with the

environmental elements. While in modeling the processes and designing the controller configuration it was proven mathematically that the use of acceleration feedback can improve the performance of the systems, during the experiment part different factors reduce the level of the expected improvements. Often the noise, improper sensors or not well tuned instruments can diminish the full amount of improvement calculated and expected from analytical work and simulations. Improvements have been observed in experiments and confirm the modeling and the analytical results.

4.1.2- Example of Controller Design with MATLAB MPC Toolbox

This example introduces through a simple application the design of predictive controller with/without disturbances and with constraints. The possibility of introducing constraints in outputs and inputs is one of the capabilities of the predictive control not found in other approaches. The system in discussion is a positional servo system in Problem A-4-3 [18]. The open loop stable transfer function of the DC servomotor has been calculated and used in the design of a predictive controller with constraints. The m-files and the plots obtained step by step in building the controller are presented entirely in the Appendix 5, in this chapter only the main steps and results are presented as an introduction of the method.

The MPC toolbox is an application specific to the predictive control which has been introduced with Matlab software. It uses less extensively the Simulink toolbox and facilitates design and analysis of systems with predictive approaches more analytical, rather old programming style. The entire design process could be done through one file

only, all the simulations and the plots being obtained successively with calls from within the window command of Matlab.

The system is a simple SISO (Single Input – Single Output) model [23], with the input the voltage for the actuator (control signal), and the output the position of the load (angular position). The setpoint for the output is the desired value for the angular position. The model, without disturbances and without constraints is represented with:

$$G(s) = 42.3/s^2+7.69s+42.3 \qquad (4.1.2.1)$$

The model is defined in mod format (specific to MPC Toolbox), with sampling period of 0.1 sec:

```
T=0.1;
ny =1;
G=poly2tfd(42.3,[1 7.69 42.3],0,0);
pmod=tfd2mod(T,1,G);
```

The model has been created in mod form after being transformed first in a discrete system with the same sampling period T=0.1 sec. Then, a MPC controller without constraints is build with the commands:

```
imod=pmod; % assume perfect modeling
ywt=[];    % default (unity) weight on output
uwt=[];    % default (zero) weight input
P=5;       % prediction horizon
M=P;       % control horizon
Ks=smpccon(imod,ywt,uwt,M,P);
```

The design parameters have been chosen such that the model is identical to the system, it is a perfect controller. To check the design, a simulation is run with a step increase in the setpoint of output y:

```
tend=10;    % time period for simulation.
r=[1];      % setpoints for the two outputs.
[y,u]=smpcsim(pmod,imod,Ks,tend,r);
plotall(y,u,T)
```

```
[clmod,cmod]=smpccl(pmod,imod,Ks);
smpcpole(cmod)
```

The result is shown in Figure 18: there is no error since the plant pmod and the

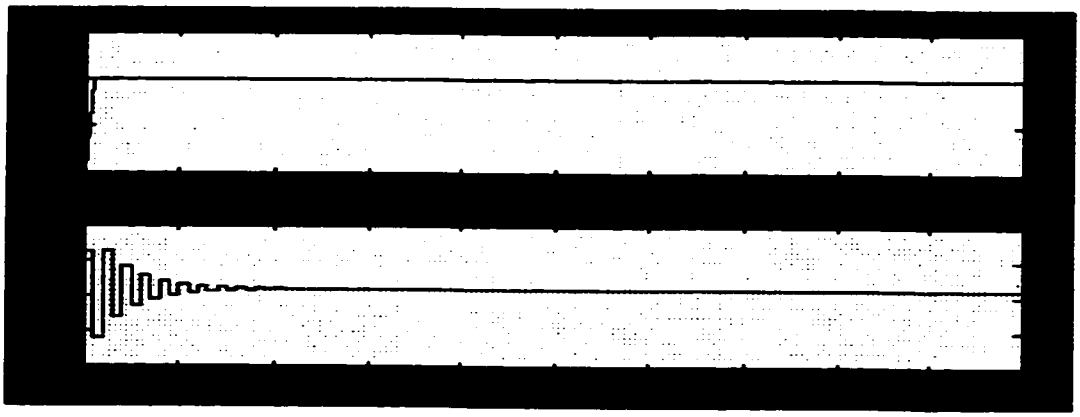


Figure 18: Response of the system for Unit step input, P=5,M=5.

controller imod are identical. The tracking is perfect, the specified setpoint change has been done. The control signal (voltage applied to the actuator) is however oscillatory for an extended period at the start and should be made smoother. This is the case given the poles of the controller that are closed to the imaginary axis:

```
» [clmod,cmod]=smpccl(pmod,imod,Ks);
» smpcpole(cmod)
ans =
      0
  -1.0000
  -0.0132
  -0.0000
```

To smoothen the control signal the prediction horizon P is increased from 5 to 10, and the control horizon M is reduced to 3:

```
P=10;
M=3;

Ks=smpccon(imod,ywt,uwt,M,P);

[y,u]=smpcsim(pmod,imod,Ks,tend,r);
```

```
plotall(y,u,T)
```

The improved results can be seen on Figure 19. Also blocking could be used as well to indicate the number of steps over which $\Delta u = 0$. The number of steps satisfying

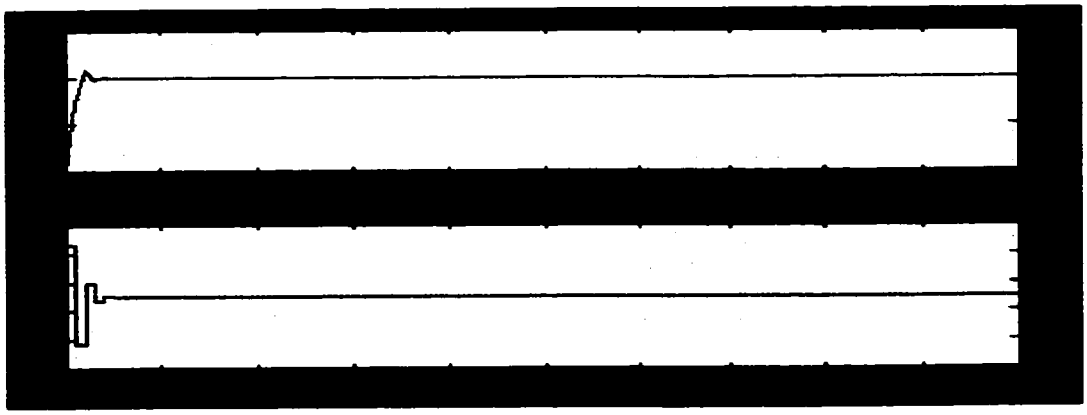


Figure 19: Response of the system for Unit step input, P=10,M=3.

the condition is indicated by the number of elements in the row for M, in this case equal to 3. The improvements can be seen again on Figure 20.

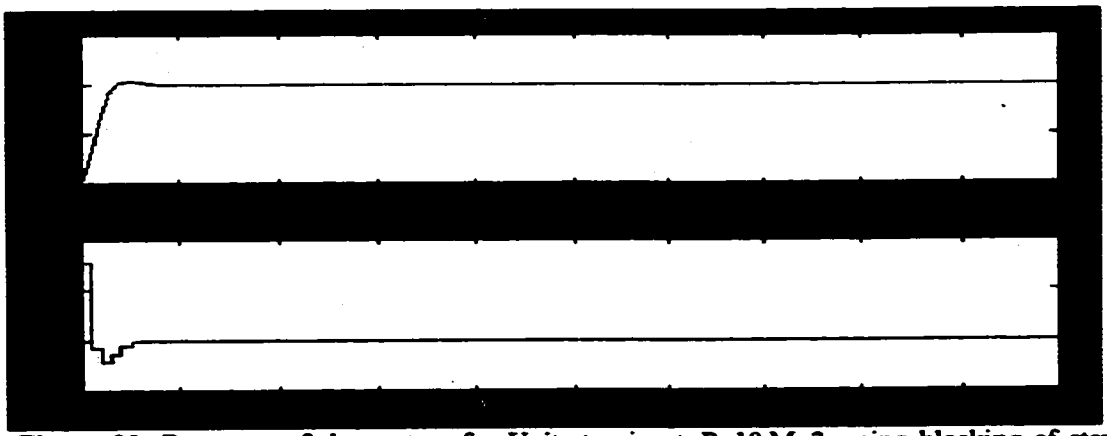


Figure 20: Response of the system for Unit step input, P=10,M=3, using blocking of steps of control horizon.

It is also possible to improve the performance by manipulating the weights and the horizon duration for prediction and control. The next important aspect of the design is to introduce disturbances. To see their influence, the output is set at 0 and measured or

unmeasured disturbances are introduced in the design to increase the robustness of the controller. In this case, the test was done with an unmeasured disturbance:

```
T=0.1;
ny =1;
G=poly2tfd(42.3,[1 7.69 42.3],0,0);
pmod1=tfd2mod(T,ny,G);

%Disturbance model: hd=1.2e^(-0.2s)/0.14s+1
Gd=poly2tfd(1.2,[0.14 1],0,0.2);
dmod=tfd2mod(T,ny,Gd);
pmod=addumd(pmod1,dmod);      %Combines the model with the disturbance

ulim=[];    % default (no) constraints on u variable.
Kest=[];    % default (DMC) state estimator.
r=[0];      % Output setpoint at zero.
z=[];       % default (zero) measurement noise.
v=[];       % default (zero) measured disturbances.
w=[1];      % unit-step in unmeasured disturbance.
[y,u]=smpcsim(pmod,imod,Ks,tend,r,ulim,Kest,z,v,w);
plotall(y,u,T)
```

As seen in the Figure 21, the influence of the disturbance is important. The models for the disturbance factors should be appropriately selected for the system to be designed.

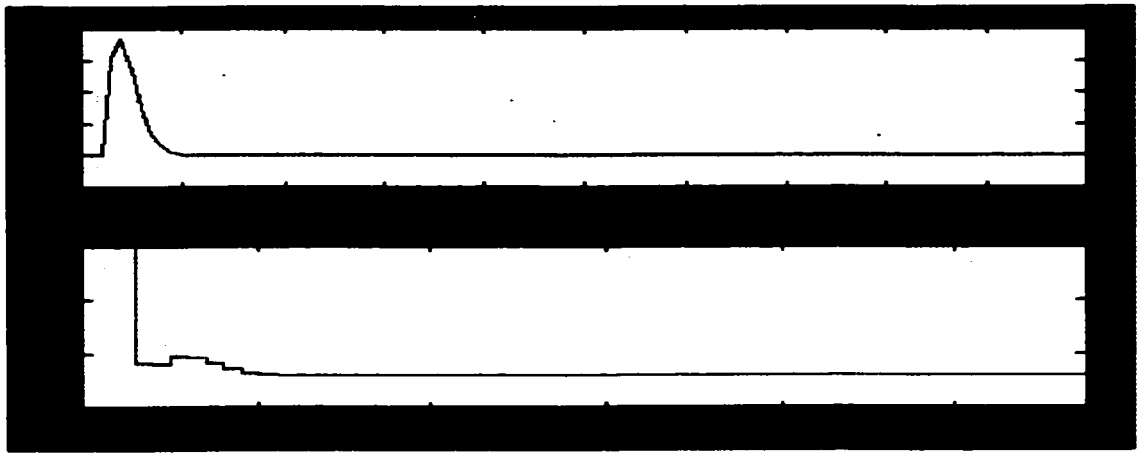


Figure 21: Response of the system for Unit step input in unmeasured disturbance, P=10,M=3, setpoint output 0.

The transient regime has to be acceptable for the case of reasonable disturbances. During the study, the disturbance could be increased to reach the limits of the stability of the system. As it can be observed, the disturbance also introduces a delay in the control

signal. To improve even further, a default estimator, a Kalman estimator, can be introduced to replace the default one used previously (DMC type):

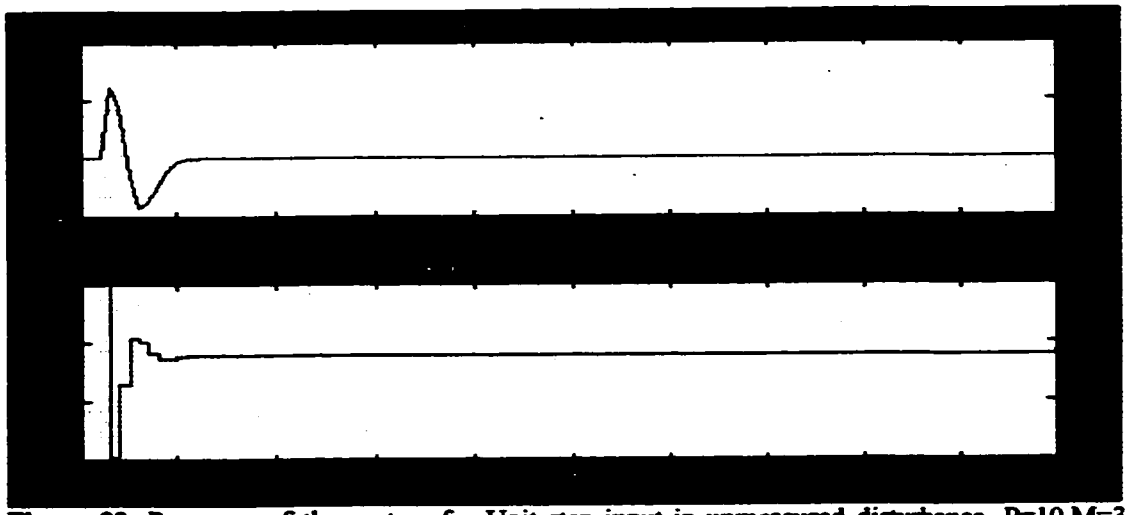


Figure 22: Response of the system for Unit step input in unmeasured disturbance, P=10,M=3, setpoint output 0, estimator type Kalman.

With the better estimator the system compensates much faster for the unmeasured disturbances. The problem of measured disturbances is relatively easier since the known models for disturbances that can be evaluated through measurements give the opportunity to reduce their influence directly by the design.

To test the system for handling constraints, first very large constraints for the system are introduced. The plots at this stage are for identical systems. The constraints are imposed on the output and the input to see how the limitations appear and find out which of the constraints should be imposed. If the constraints are too tight, the system may not perform properly or the calculations will take an indefinitely long time.

```
ulim=[-inf inf 10];
ylim=[];
setpts=[0.8];                    % Define the step in the setpoint.
plant=imod;                      % First run an unconstrained simulation.
Ks=smpccon(imod,ywt,uwt,blks,nhor);

tend=20;                         % Duration of the simulation
[yl,ul]=smpcsim(plant,imod,Ks,tend,setpts,ulim,K);
```

```
                % Run the constrained simulation and plot the results.
[y,u]=scmpc(plant,imod,ywt,uwt,blks,nhor,tend, ...
                setpts,ulim,ylim,K);

plotall([y yl],[u ul],T);
```

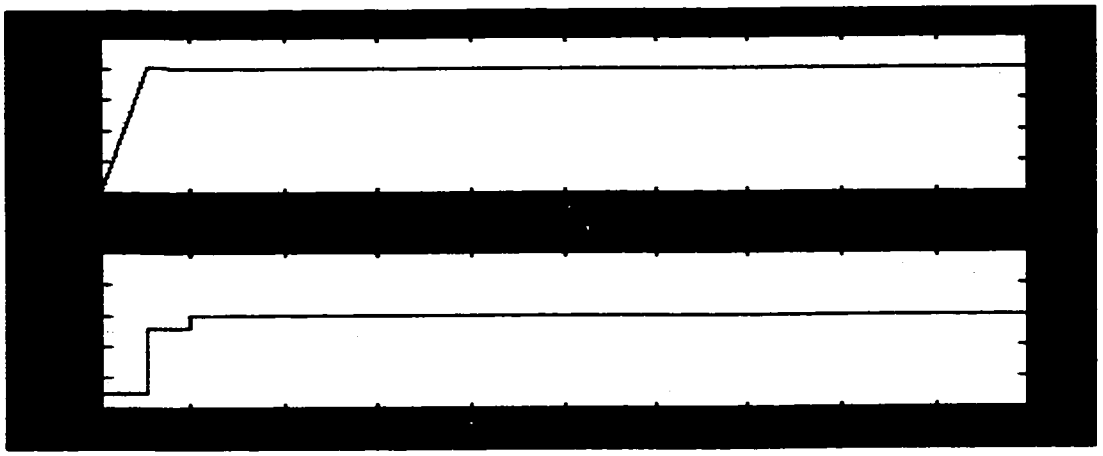


Figure 23: Response of the system for 0.8-step input in the setpoint and large constraints.

The two systems, one without constraints and the other with very large constraints (resulting practically in a system with no constraints), are plotted on the same graph. Figure 23 the two plots are identical (superimposed) and the control signal is the same. The next step, for introducing the constraints, is to impose a constraint close to the output and then another constraint on the input to evaluate the capabilities of the controller to handle these restrictions that may be imposed in the design phase of the project.

```
% Add upper bound of 0.7 on output, accounting for 3 samples time delay

ylim =[-inf  inf
       -inf  inf
       -inf  inf


       -inf  0.7];
[y,u]=scmpc(plant,imod,ywt,uwt,blks,nhor,tend, ...
                setpts,ulim,ylim,K);
```

```
plotall(y,u,T);
```

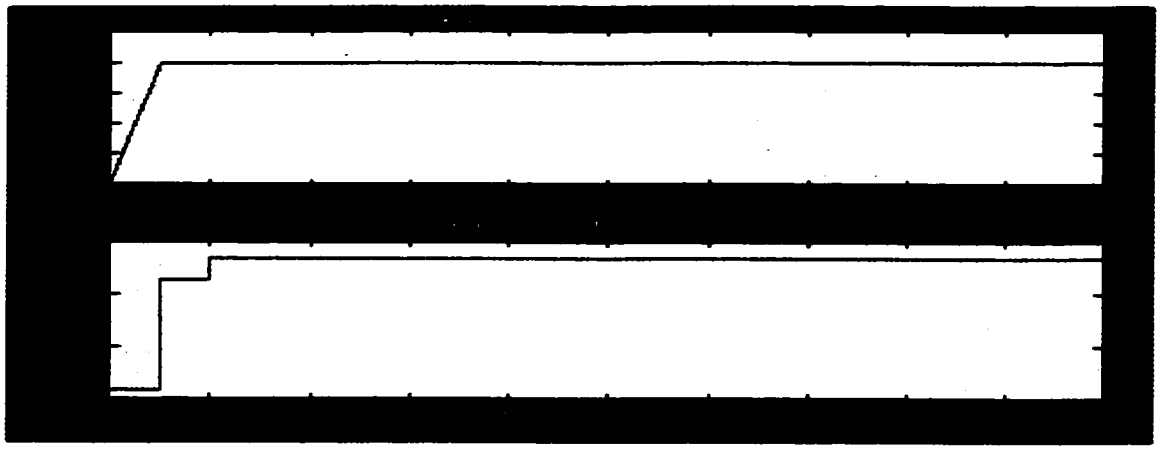


Figure 24: Response of the system for 0.8-step input in the setpoint and output constraint at 0.7

As seen from Figure 26.g, the constraint on output (angular position) is respected only for the first only one or two step. After introducing the constraint condition for more steps of the control horizon, the constraints are, then, respected. The constraints strictly imposed are respected after a minimal number of samples and are presented on Figures 23-24.

```
ylim=[-inf inf
      -inf 0.7
      -inf 0.7
      -4.5 0.7
      -inf 0.7];

[y,u]=scmpc(plant,imod,ywt,uwt,blks,nhor,tend, ...
                setpts,ulim,ylim,K);
plotall(y,u,T);

pause;
ulim=[ -10.5 0.6 0.5
       -10.5 0.6 0.5];

 [y,u]=scmpc(plant,imod,ywt,uwt,blks,nhor,tend, ...
                setpts,ulim,ylim,K);

plotall(y,u,T)
```

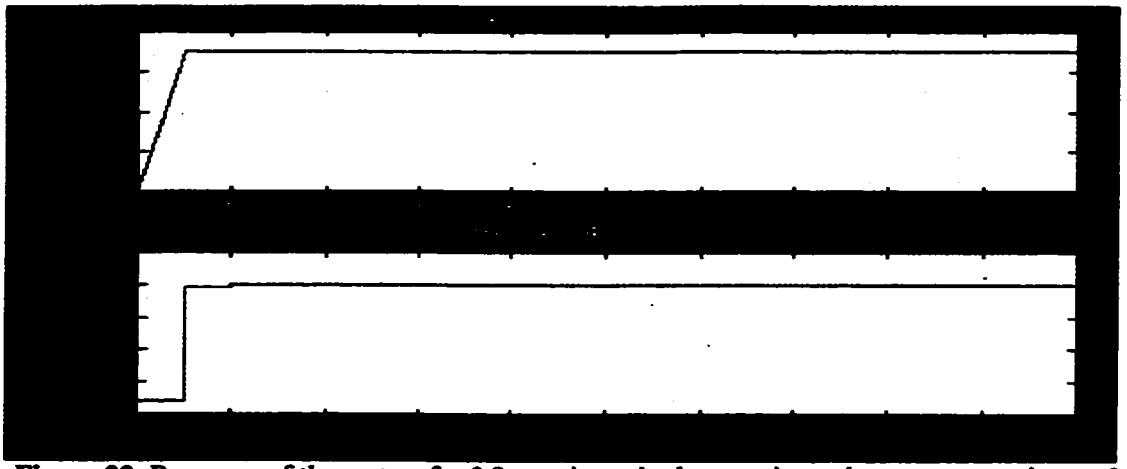


Figure 23: Response of the system for 0.8-step input in the setpoint and output constraints at 0.7 for output and 0.6 for control signal.

The capability of satisfying the constraints in inputs/outputs, characteristic only for the predictive control, allows the designer to create controllers with special functions for particular processes and applications. Figure 24 displays the total conditioning of the input and output by the desired constraints. With these basic design conditions insured, a complete MPC of the nonlinear plant can be constructed for the real time processes with realistic values concerning noise, disturbances and constraints.

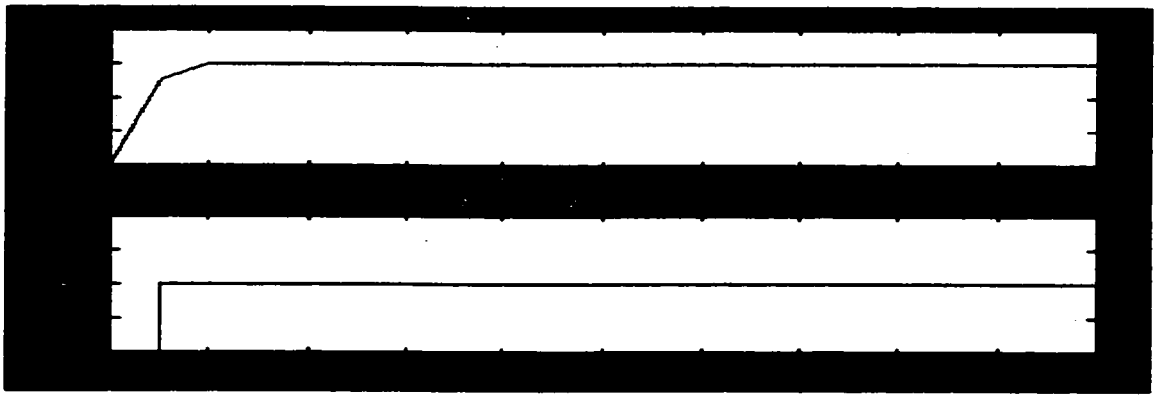


Figure 24: Response of the system for 0.8-step input in the setpoint and output constraints at 0.7 for output and 0.6 for control signal, with constraints reinforced for all steps after the first 3 sampling periods.

In Figure 25, the influence of constraints for both controller signals plotted on the same graph, is obviously in different stages of the process.

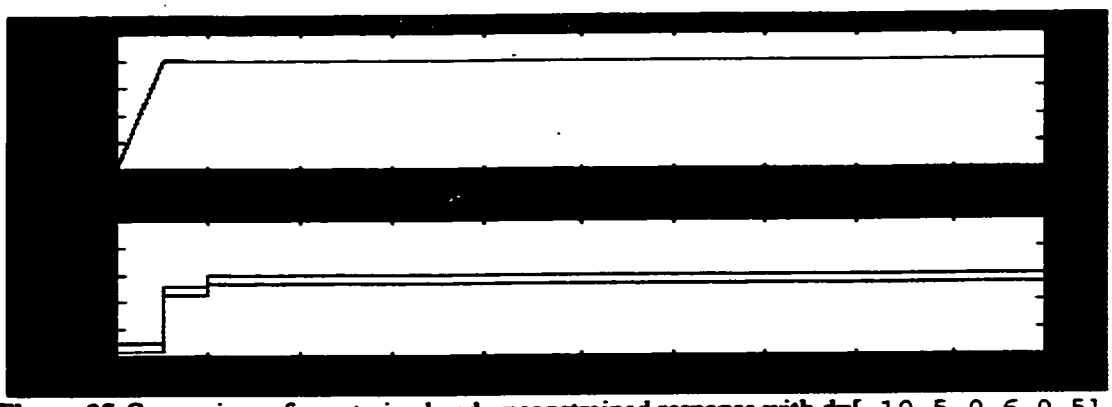


Figure 25:Comparison of constrained and unconstrained response with d=[-10.5 0.6 0.5].

In general, the MPC Toolbox offers the possibility of testing rapidly different, complex controllers for systems with large number of manipulated and controlled variables. It also allows the design of controllers for systems with significant constraints, making the predictive method unique in handling the constraint problem. MPC toolbox also offers some solutions for systems with time delay, which are relatively difficult to solve with other methodologies. The steps taken in using the MPC toolbox consist of:

- identifying the model (transfer function, state-space, disturbances, constraints, continuos/discrete);
- designing the predictive parameters list, i.e. gains, weights, horizons, by testing rapidly, through programming code, in simulations and by using different methods for the selection of parameters;
- applying the conditions imposed by the constraints, introducing observers, filters and other tools specific to the control domain and to the predictive methods in particular;
- writing S-files and using established MPC blocks from MPC and Simulink

4.1.2 – MPC, DMC/AF System Design

From the nonlinear model described in [2], a linear state space model was completed for control design, for the driving motor torque M_d, the drive-side (motor) and the load-side (ballscrew) angular positions φ_d and φ_l, velocities Ω_d and Ω_l , and acceleration $\alpha_l = \dot{\Omega}_l$ are chosen as state variables [1].

$$\dot{\underline{x}}_p = {}^c\underline{\underline{A}}_p \underline{x}_p + {}^c\underline{b}_{pc} u_{pc} + {}^c\underline{b}_{pd} u_{pd}$$

$$\begin{bmatrix} \dot{\varphi}_d \\ \dot{\Omega}_d \\ \dot{\varphi}_l \\ \dot{\Omega}_l \\ \dot{\alpha}_l \\ \dot{M}_d \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{c_{dl}}{J_d} & -\frac{b_d + b_{dl}}{J_d} & \frac{c_{dl}}{J_d} & \frac{b_{dl}}{J_d} & 0 & \frac{1}{J_d} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -\frac{c_{dl} b_{dl}}{J_d J_l} & -\frac{(b_d + b_{dl}) b_{dl}}{J_d} \frac{1}{J_l} + \frac{c_{dl}}{J_l} & \frac{b_{dl} c_{dl}}{J_d J_l} & \frac{b_{dl}^2}{J_d J_l} - \frac{c_{dl}}{J_l} & -\frac{b_d + b_l}{J_l} & \frac{b_{dl}}{J_d J_l} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{servo}} \end{bmatrix} \cdot \begin{bmatrix} \varphi_d \\ \Omega_d \\ \varphi_l \\ \Omega_l \\ \alpha_l \\ M_d \end{bmatrix} +$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{J_d} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -\frac{1}{J_l} & 0 \\ \frac{k_{servo}}{T_{servo}} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{servo} \\ M_l + M_{fl} \\ M_{fd} \end{bmatrix}$$

(4.1.2a)

The state space model is given by the differential equations (4.1.2a) of the motor, where the tachometer contributes to the drive-side moment of inertia J_d, the screw, the encoder, the accelerometer, the carriage, the friction wheel and load motor inertia add up to J_l at the load side. The compliance between the motor and the positioning unit is modeled by a spring with stiffness c_{dl} and damping constant b_{dl} for material damping. Viscous friction with damping constants b_d and b_l is allocated to th the drive and the load side. A load-side friction torque Mfrl, which in the linear model is considered to be external disturbance input, accounts for the dry friction in the ball screw drive. External forces F_l acting at the carriage are counteracted by the input torque M_l .

The current controlled servo amplifier is modeled as a first order system with input u_{servo} and output M_d .Its parameters are the gain k_{servo} and the time constant T_{servo} .In the measurement output equations (4.1.2b) the DC tacho generator, the accelerometer and the incremental encoder are simply modeled by their stationary gains k_Ω, k_α and k_φ , respectively.

The state-space model was extended from its classical form in [2], with only position and velocity measurements, to the new one with measurement output equations including acceleration, and finally to its DMC/AF form given by the equations (4.1). Another equation needed for optimal control design is the output equation (4.1.2c) to provide the objective variables of the plant, which will be used to formulate the design objectives for the carriage motion.

These objective variables are the load-side angular position φ_l, velocity Ω_l, and acceleration $\alpha_l = \dot{\Omega}_l$. In the general form of the above equations, the subscripts p, c, d, m,

o are abbreviations of the words plant, control, disturbance, measurement and objective, respectively. With these the vector $\underline{x}_p$ denotes the state vector, the vector $\underline{y}_{po}$ the vector of the objective outputs of the plant. The scalar input u_{pd} is the disturbance input to the plant; and the column vector $\underline{d}_{pod}$ is the corresponding feedthrough matrix from the disturbance input to the objective output.

The superscript c at the plant system and input matrices denotes the continuous character of the system model. For the above linear model, a LQG position control has been designed in [2] with full feedback of the state vector $\underline{x}_p$ of the plant, and feedforward of reference values for the objective variables in $\underline{y}_{po}$. For friction compensation, it has been augmented by a feedforward term of an estimate for the disturbing load-side torque at the input u_{pd}. Due to active vibration damping provided by the state feedback, this approach yields a high control bandwidth. In the analysis of the case with the state observer design, the controller shows a high robustness against uncertain plant properties [8].

The simulation and experimental results obtained with this LQG control are used as a basis for comparison with the simple MPC and DMC/AF approach. For this purpose the linear quadratic regulator in the LQG approach, which contains the gains of above mentioned state feedback, reference and disturbance feedforwards, will be replaced by the gains of the MPC. The Kalman filter from the LQG approach is kept to estimate the unknown plant and disturbance model states. More details about the LQG control, with which the comparisons were made, are in [2].

$$\underline{y}_{pm} = \underline{C}_{pm}\,\underline{x}_p$$

$$\begin{bmatrix} u_{acc} \\ u_{tacho} \\ count \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & k_\alpha & 0 \\ 0 & k_\Omega & 0 & 0 & 0 & 0 \\ 0 & 0 & k_\varphi & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \varphi_a \\ \Omega_a \\ \varphi_i \\ \Omega_i \\ \alpha_i \\ M_d \end{bmatrix}$$

(4.1.2b)

$$\underline{y}_{po} = \underline{C}_{po}\,\underline{x}_p + \underline{d}_{pod}\,u_{pd}$$

$$\begin{bmatrix} \varphi_1 \\ \Omega_1 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \varphi_d \\ \Omega_d \\ \varphi_1 \\ \Omega_1 \\ \alpha_i \\ M_d \end{bmatrix}$$

(4.1.2c)

The method used for building the MPC control was the Dynamic Matrix Control (DMC) form [10]. From the state-space model, a simple MPC control was developed and it was tested with acceleration feedback separately from the state position and velocity feedback This model was compared to the LQG approach, and then the new state model with acceleration at the load side as a state variable was introduced. Several methods of MPC can be found in the literature [11]. A brief summary of the derivation and structure of MPC/DMC and DMC/AF is given below.

Starting from the DMC design, as in [1], the discrete plant model

$$\underline{x}_{p,k+1} = \underline{A}_p\,\underline{x}_{p,k} + \underline{b}_{pc}\,u_{pc,k} + \underline{b}_{pd}\,u_{pd,k} \qquad (4.2a)$$

is derived from the continuous model from equation (4.1.2a) by step-invariant discretization. The output equations needed for the design

$$y_{po\,i,k} = \underline{c}_{po\,i}^{T}\,\underline{x}_{p,k} + d_{pod\,i}\,u_{pd,k} + d_{poc\,i}\,u_{pc,k} \quad , \quad i = 1 \,..\, q \tag{4.2b}$$

represent a more general form of equation (4.1.2c). The q = 3 output variables in equation (4.1.2c) are not directly affected by the control input to the plant u_{pc}. The feedthrough matrix $\underline{d}_{poc}$ is equal to zero, so that the corresponding term does not appear. If a constant control signal $u_{pc,k} = u_{pc,k+1} = \ldots = u_{pc,k+M}$ is assumed over a prediction horizon of length M, a quadratic cost function can be formulated as the sum

$$J_k = \sum_{i=1}^{q} (\underline{\bar{w}}_i - \underline{\bar{y}}_{po\,i})^T\, \underline{Q}_i\, (\underline{\bar{w}}_i - \underline{\bar{y}}_{po\,i}) + \sum_{j=1}^{M} r_j\, u_{pc,k}^2 \tag{4.3a}$$

where

$$\underline{\bar{y}}_{po\,i} = \begin{bmatrix} y_{po\,i,k+1} \\ y_{po\,i,k+2} \\ \vdots \\ y_{po\,i,k+M} \end{bmatrix} \tag{4.3b}$$

is the vector of future values of the objective output $y_{po\,i}$,

$$\underline{\bar{w}}_i = \begin{bmatrix} w_{i,k+1} \\ w_{i,k+2} \\ \vdots \\ w_{i,k+M} \end{bmatrix} \tag{4.3c}$$

is the vector of the corresponding reference values, and

$$\underline{Q}_i = \mathrm{diag}(q_{i1}, q_{i2}, \ldots, q_{iM}) \tag{4.3d}$$

is a weighting matrix for the error of the ith elements of the q objective outputs, and r_j are weightings for the constant control signal over the prediction horizon.

Expressing the elements of $\bar{\underline{y}}_{poi}$ with the model equations (4.2) and minimizing the resulting cost function from equation (4.3a) with respect to $u_{pc,k}$,i.e. solving $\partial J_k / \partial u_{pc,k} \equiv 0$, the control law becomes:

$$u_{pc,k} = \left[\sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \underline{s}_{pi} + \sum_{j=1}^{M} r_j \right]^{-1} \cdot \left[\sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \cdot \bar{\underline{w}}_i - \sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \underline{T}_{pi} \cdot \underline{x}_{p,k} - \sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \underline{S}_{di} \cdot \bar{\underline{u}}_{pd} \right] \tag{4.4a}$$

where

$$\underline{s}_{pi} = \begin{bmatrix} \underline{c}_{poi}^T \underline{b}_{pc} + d_{poci} \\ \underline{c}_{poi}^T \left[\underline{A}_p \underline{b}_{pc} + \underline{b}_{pc} \right] + d_{poci} \\ \vdots \\ \underline{c}_{poi}^T \left[\underline{A}_p^{M-1} \underline{b}_{pc} + \underline{A}_p^{M-2} \underline{b}_{pc} + \ldots + \underline{b}_{pc} \right] + d_{poci} \end{bmatrix}, \tag{4.4b}$$

$$\underline{T}_{pi} = \begin{bmatrix} \underline{c}_{poi}^T \underline{A}_p \\ \underline{c}_{poi}^T \underline{A}_p^2 \\ \vdots \\ \underline{c}_{poi}^T \underline{A}_p^M \end{bmatrix}, \tag{4.4c}$$

$$\underline{S}_{di} = \begin{bmatrix} \underline{c}_{poi}^T \underline{b}_{pd} & d_{podi} & 0 & \cdots & 0 & 0 \\ \underline{c}_{poi}^T \underline{A}_p \underline{b}_{pd} & \underline{c}_{poi}^T \underline{b}_{pd} & d_{podi} & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & d_{podi} & \vdots \\ \underline{c}_{poi}^T \underline{A}_p^{M-1} \underline{b}_{pd} & \underline{c}_{poi}^T \underline{A}_p^{M-2} \underline{b}_{pd} & \vdots & \cdots & \underline{c}_{poi}^T \underline{b}_{pd} & d_{podi} \end{bmatrix}, \tag{4.4d}$$

and $$\bar{\underline{u}}_{pd} = \begin{bmatrix} u_{pd,k} \\ u_{pd,k+1} \\ \vdots \\ u_{pd,k+M} \end{bmatrix}. \tag{4.4e}$$

As described in [2] the sum of the load and the friction torque M_l+M_{frl} of the disturbance input u_{pd} can well be approximated by a constant function of time. Thus, it can be modeled as the output of a discrete integrator disturbance model

$$\begin{aligned} x_{d,k+1} &= x_{d,k} \\ y_{d,k} &= x_{d,k} \end{aligned} \tag{4.5}$$

with appropriate non-zero initial condition $x_{d,0}$. Substituting $u_{pd,k} = y_{d,k}$ in equations (4.4e) the control law (4.4a) yields:

$$u_{pc,k} = \left[\sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \underline{s}_{pi} + \sum_{j=1}^{M} r_j\right]^{-1} \cdot \left[\sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \cdot \overline{\underline{w}}_i - \sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \underline{T}_{pi} \cdot \underline{x}_{p,k} - \sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \underline{s}_{di} \cdot x_{d,k}\right] \qquad (4.6a)$$

where

$$\underline{s}_{di} = \begin{bmatrix} \underline{c}_{poi}^T \underline{b}_{pd} + d_{pod\,i} \\ \underline{c}_{poi}^T \left[\underline{A}_p \underline{b}_{pd} + \underline{b}_{pd}\right] + d_{pod\,i} \\ \vdots \\ \underline{c}_{poi}^T \left[\underline{A}_p^{M-1} \underline{b}_{pd} + \underline{A}_p^{M-2} \underline{b}_{pd} + \ldots + \underline{b}_{pd}\right] + d_{pod\,i} \end{bmatrix}. \qquad (4.6b)$$

In this form of the control law there are still i = 1.. q reference variables $\underline{w}_i$ and their future values in the vector $\overline{\underline{w}}_i$. For the given position control, the variables are the reference position, velocity, and acceleration, which can not be chosen independently. The last results from the previous one by differentiation. A discrete double integrator reference model can finally express this as for the desired acceleration input u_r:

$$\begin{aligned} \underline{x}_{r,k+1} &= \underline{A}_r \underline{x}_{r,k} + \underline{b}_r u_{r,k} \\ y_{ri,k} &= \underline{c}_{ri}^T \cdot \underline{x}_{r,k} + d_{ri} u_{r,k} \end{aligned} \qquad (4.7)$$

where the first output is the desired position, equal to the first state, the second output is the desired velocity, equal to the second state, and the third output is the acceleration, equal to the input u_r. By substituting the vectors $\overline{\underline{w}}_i$, of future reference values, by the vectors $\overline{\underline{y}}_{ri}$ of future values of the reference model outputs, and rewriting the latter with the reference model, equations (4.7) become:

$$\overline{\underline{w}}_i = \overline{\underline{y}}_{ri} = \underline{T}_{ri} \cdot \underline{x}_{r,k} + \underline{S}_{ri} \cdot \overline{\underline{u}}_r \qquad (4.8a)$$

$$\text{with } \underline{T}_{ri} = \begin{bmatrix} \underline{c}_{ri}^T \underline{A}_r \\ \underline{c}_{ri}^T \underline{A}_r^2 \\ \vdots \\ \underline{c}_{ri}^T \underline{A}_r^M \end{bmatrix}, \qquad (4.8b)$$

and

$$\underline{S}_{ri} = \begin{bmatrix} \underline{c}_{ri}^T \underline{b}_r & d_{ri} & 0 & \cdots & 0 & 0 \\ \underline{c}_{ri}^T \underline{A}_r \underline{b}_r & \underline{c}_{ri}^T \underline{b}_r & d_{ri} & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & d_{ri} & \vdots \\ \underline{c}_{ri}^T \underline{A}_r^{M-1} \underline{b}_r & \underline{c}_{ri}^T \underline{A}_r^{M-2} \underline{b}_r & \vdots & \cdots & \underline{c}_{ri}^T \underline{b}_r & d_{ri} \end{bmatrix}, \tag{4.8c}$$

This leads to the final control law:

$$u_{pc,k} = \left[\sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \underline{s}_{pi} + \sum_{j=1}^{M} r_j \right]^{-1} \cdot \left[\sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \underline{S}_{ri} \cdot \bar{\underline{u}}_r + \sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \underline{T}_{ri} \cdot \underline{x}_{r,k} - \sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \underline{T}_{pi} \cdot \underline{x}_{p,k} - \sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \underline{s}_{di} \cdot x_{d,k} \right] \tag{4.9}$$

The remaining unique reference input vector to the system

$$\bar{\underline{u}}_r = \begin{bmatrix} u_{r,k} \\ u_{r,k+1} \\ \vdots \\ u_{r,k+M} \end{bmatrix} \tag{4.10}$$

contains the current and future values of the acceleration command. Corresponding to the double integrator reference model, with piecewise constant acceleration, the reference velocity and reference position become piecewise linear and parabolic signals with respect to time (Figure 19). In the simulation and prototyping environment, a reference profile generator generates the above class of signals for position control applications. Thus, for the simulation studies and implementation of the control, the reference profile generator replaces the reference model.

For control implementation, equation (4.9) is rewritten as

$$u_{pc,k} = \underline{k}_{filt\,r}^T \cdot \bar{\underline{u}}_r + \underline{k}_{ff\,r}^T \cdot \underline{x}_{r,k} - \underline{k}_{fb\,p}^T \cdot \underline{x}_{p,k} - k_{ff\,d} \cdot x_{d,k} \tag{4.11}$$

where

$\underline{k}_{filt\,r}^T = \underline{H} \cdot \sum_{i=1}^{q} \underline{s}_{pi}^T \underline{Q}_i \underline{S}_{ri}$ is the reference filter gain vector for the reference input vector (acceleration),

$\underline{k}_{ff\,r}^{T} = \underline{H} \cdot \sum_{i=1}^{q} \underline{s}_{p\,i}^{T} \underline{Q}_i \underline{T}_{r\,i}$ is the feedforward gain vector for the reference model states (position and velocity),

$\underline{k}_{fb\,p}^{T} = \underline{H} \cdot \sum_{i=1}^{q} \underline{s}_{p\,i}^{T} \underline{Q}_i \underline{T}_{p\,i}$ is the plant feedback gain vector,

$k_{ff\,d} = \underline{H} \cdot \sum_{i=1}^{q} \underline{s}_{p\,i}^{T} \underline{Q}_i \underline{s}_{d\,i}$ is the disturbance feedforward gain,

where $\underline{H} = \left[\sum_{i=1}^{q} \underline{s}_{p\,i}^{T} \underline{Q}_i \underline{s}_{p\,i} + \sum_{j=1}^{M} r_j \right]^{-1}$.

The model was first tested with acceleration feedback separatedly from the plant feedback gain vector. In this case, for q = 2, $\underline{k}_{fbp}^{T} = \underline{H} \cdot \sum_{i=1}^{q} \underline{s}_{pi}^{T} \underline{Q}_i \underline{T}_{pi}$ contained only position and velocity feedback gains, while acceleration feedback gain was introduced separately. However, the plant feedback gain vector can also include the gain for acceleration feedback for q = 3. The block diagram in Figure 26 illustrates the structure of the corresponding control system.

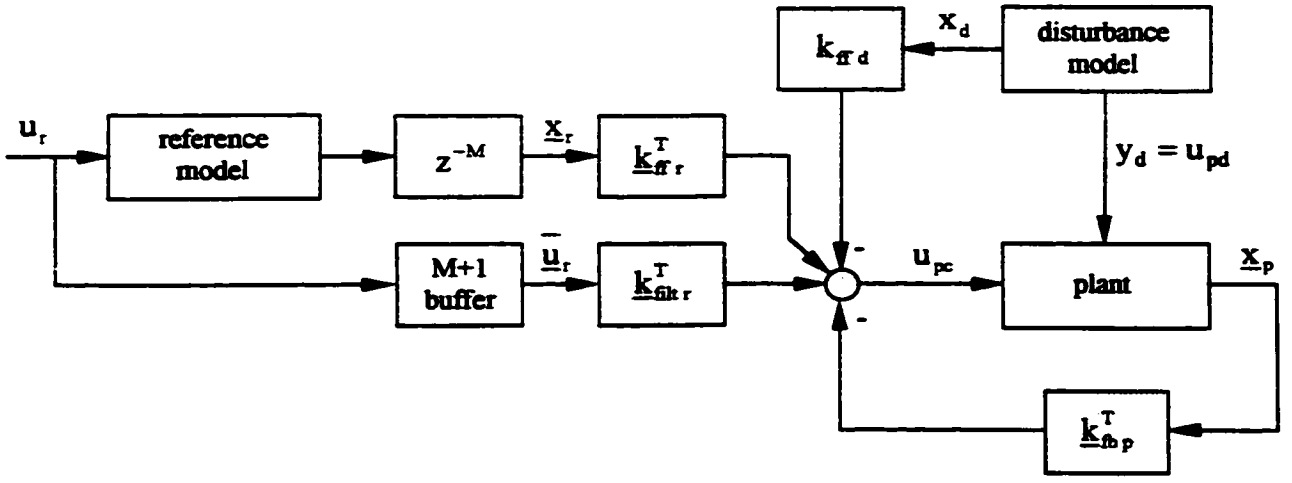


Figure 26: Structure of MPC, DMC/AF.

The shaded blocks describe the control law from equation (4.11). They contain the values of the control signal u_{pc}, the reference model state $\underline{x}_r$, plant state $\underline{x}_p$, and disturbance model state $\underline{x}_d$ at the current time instant k and, as shown with equation (4.10), the current value $u_{r,k}$ as well as M future values $u_{r,k+1}$ to $u_{r,k+M}$ of the reference model input u_r. For implementation, a buffer of length M+1 is required to provide the current and future values of the reference model input in the vector $\underline{\bar{u}}_r$ from equation (4.10). When the controller starts, this buffer is empty and then is filled step by step. Seen from the current time of the control law computations, the current buffer inputs represent the future values of the reference model input u_r. Corresponding to equation (4.11), the reference model states have to be delayed by M steps to synchronize the start of the feedforward of the reference model state $\underline{x}_r$, when the buffer for $\underline{\bar{u}}_r$ is filled, given that both are driven by the same reference model input u_r.

For simulation studies and control implementation, the reference model is replaced by a reference profile generator, which has been developed at the CLM for position control applications [2]. An observer is required to provide the unknown state variables for disturbance feedforward and plant feedback. By using the same Kalman filter as in the LQG approach described in [2], it was possible to perform a correct comparison of the MPC results and the linear quadratic regulator control results from [2]. The control implementation and real-time testing of the control system was automated with a dSPACE real-time extension and the corresponding software environment.

4.2 – Simulink Based Model for DMC with Acceleration Feedback

As in [2], all steps required for control design, analysis, implementation as well as the experimental system analysis have been added to a GUI-driven design suite programmed under MATLAB for user convenience. With this suite, a user is able to vary the control design parameters, i.e. the control function and objective function weights as well as the length of the prediction horizon, to find the best control in short time. The proposed control presented in Figure 26 was tested using MATLAB/SIMULINK models similar to those shown in Figure 27. The observer has been derived from the continuous observer from [2] by discrete redesign for use with the MPC. The robustness achieved with the observer-based MPC is comparable to that of the LQG control reported in [2]. Figure 27 presents the simulation model for the DMC/AF used with different prediction horizons P and sample periods. The model was used in simulations with the MPC structure without acceleration feedback, then was improved with the acceleration, first

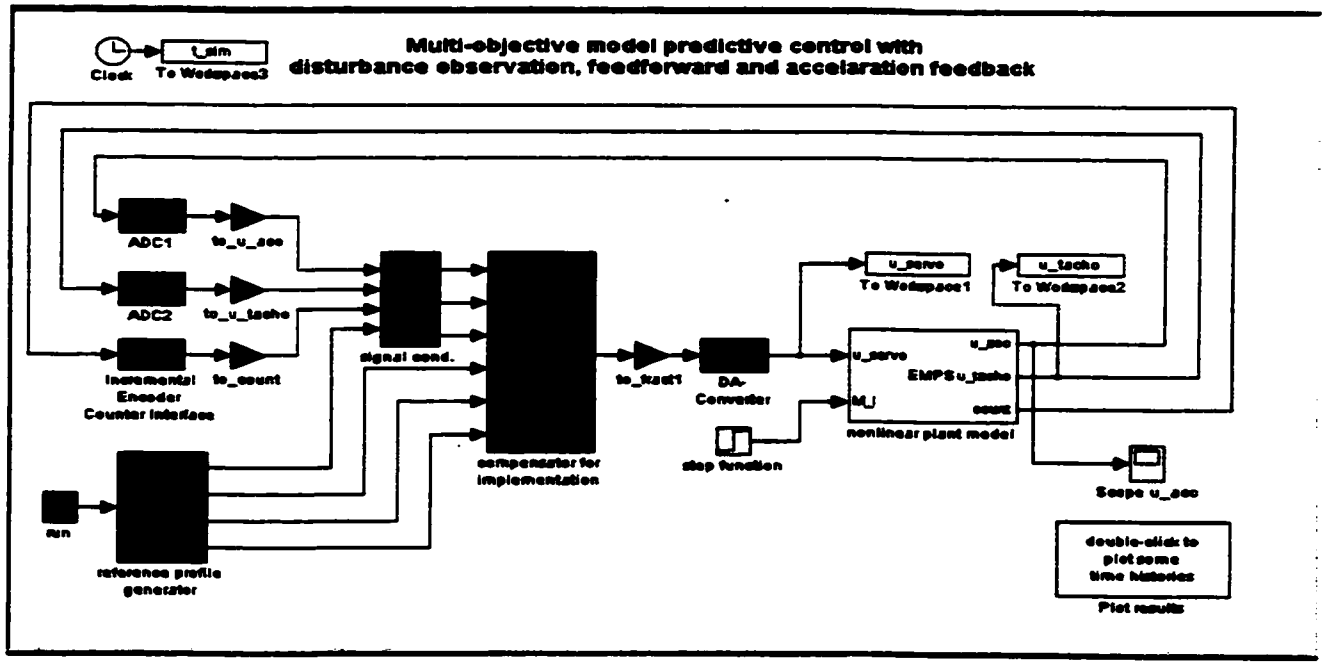


Figure 27: Simulation model for the DMC/AF design, analysis, and implementation

directly to the controller, then with the feedback in the observer. The model was also used for the simulation of extended state-space models which include drive and load accelerations and the derivative of the drive side motor momentum. The model incorporate the routines used for implementation as described while referring to the GUI.

4.3 – Matlab MPC Toolbox

In order to validate the program for the simulation shown in Figure 27, a similar predictive controller has been programmed using the Matlab toolbox MPC. The following sequence of tasks has been pursued:

- linearization of the state space model with the matrices A,B,C,D,
- transformation of the state space continuos time model into a discrete time one and then into mod form,
- choice of prediction and control horizons,
- selection by trial and error of the weighting matrices to have the same values as in the suite presented in chapter 4.2 ,
- identification of the constraints of the system and the setpoints for the outputs in the program,
- re evaluation of the setpoints by introducing values close to those in chapter 4.2 for the feedback and feedforward gains,
- calculation of the responses of the controller for unit step inputs in control signals (manipulated output), measured and not measured disturbances,
- design of the Kalman filter for a better evaluation of disturbances,

- test of a DMC-state estimator (default) and a Kalman estimator as alternatives for both measured and unmeasured disturbances,
- test of the designed controller using the nonlinear plant model,

4.4 –Simulation and Experimental Results

4.4.1- Results of Simulation with MPC/AF GUI Driven Suite

Before control law implementation, extensive simulation studies were carried out for reference and disturbance inputs to the closed loop system. In figure 28 are shown the time histories for a reference input of position, velocity and acceleration. The reference profiles contain four phases of motion: acceleration (0-0.7 sec), constant velocity (0.07-0.23 sec), deceleration (0.23 – 0.29 sec) and stand still (after 0.29 sec). A step force F_l=150 N was applied at t=0.01 sec as disturbance input. In order to show the effect of friction on the system response, two different levels of load-side friction torque (normal and increased friction level, as described in section 1) were used. Simulation results for these inputs and friction levels are presented in figures 29 to 32. Figures 33 to 35 display improvements in error tracking obtained by the MPC (Figures 29 to 32), and DMC/AF (Figures 35 and 36) compared to classical and LQG models.

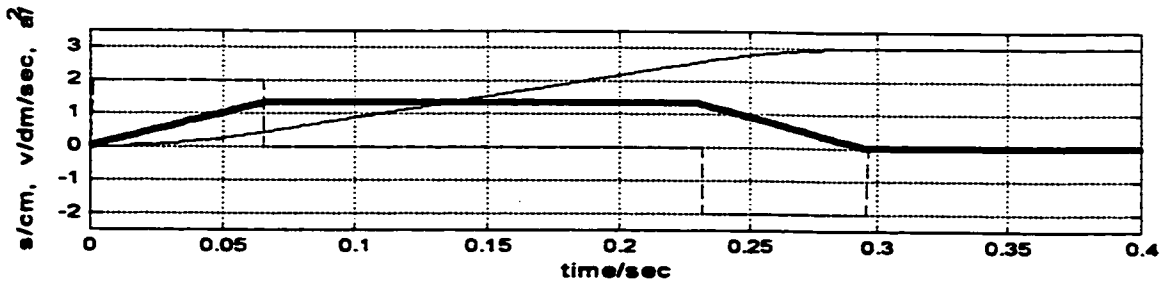


Figure 28: Reference time profiles of position (thin), velocity (thick), and acceleration (dashed) for a uni-directional carriage motion.

The limitations such as the level of noise in the acceleration signal, the phase lag created by the signal processing equipment or the non-collocation of the sensors were known in advance [6]. They were countered by better tuning of the controllers, by the reduction of the backlash value in the screw and better modeling of the accelerometer.

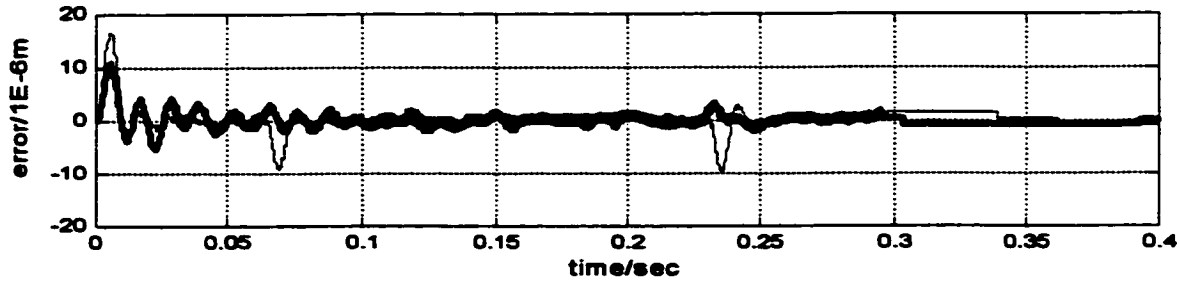


Figure 29: Simulated response of position error to reference input for MPC (thick) and LQG control (thin) and normal friction level.

In Figure 29, the results show smaller transient error amplitudes with MPC than with LQG, at t = 0.005 sec, 0.07 sec and 0.235 sec, after the acceleration changed its value. This can be explained by the smoothing of acceleration feedforward by the MPC reference filter with the gain vector $\underline{k}^T_{filt\,r}$. In terms of damping of the vibrations due to the compliant coupling, however, MPC produces less damping (non-dimensional damping factor 0.1) than LQG (non-dimensional damping factor 0.3), as noticeable in the responses up to t =0.05 sec. During all phases of motion, acceleration, constant velocity, deceleration, and at stand still (for t > 0.29 sec), the steady-state position error (after decay of the transient errors) is close to zero. This is achieved by the feedforward of the reference signals by the reference filter and the gain vector $\underline{k}^T_{ff\,r}$, and the estimation of the disturbance torque M_l+M_{fri} with a disturbance model state x_d in the observer and its

feedforward with the gain k_{ffd}. Due to the compliant coupling at the motor and the load side, where the friction is acting, the servo amplifier dynamics, and the delay of the friction estimate by the observer dynamics, the compensation of the friction is fully effective only in the steady state. From this fact, the step-like change of the friction torque with the start of the motion at t = 0.0 sec produces a large transient position error which decays to zero when the compensation becomes fully effective at t > 0.01 sec.

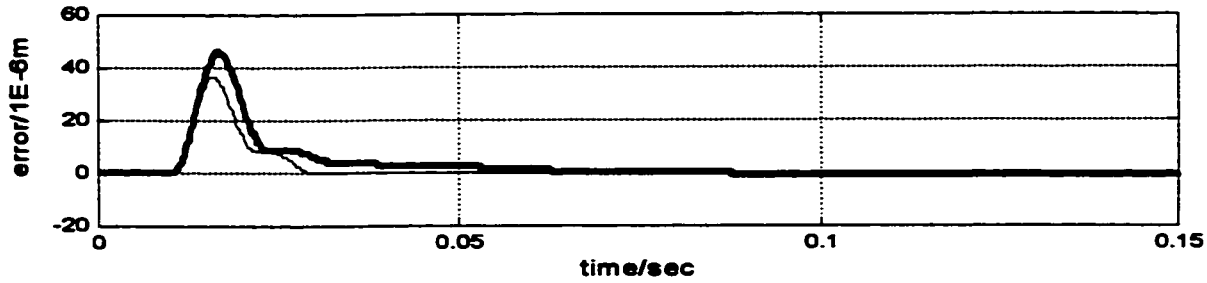


Figure 30: Simulated response of position error to step disturbance input for MPC (thick) and LQG control (thin) and normal friction level.

In figure 30, the results show poorer transient disturbance rejection from MPC compared to LQG. Due to the disturbance estimation and feedforward term, steady-state position accuracy is still good in spite of the presence of disturbing force input F_l.

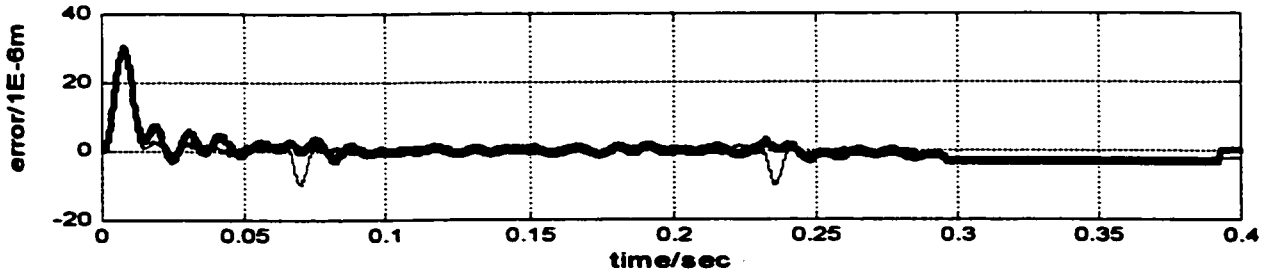


Figure 31: Simulated response of position error to reference input for MPC (thick) and LQG control (thin) and increased friction level

For the case shown in figure 31, the increased friction acts like disturbance input in addition to the reference signal applied to the system. As a consequence, the reduction of the transient error amplitude with MPC at t = 0.005 sec is diminished due to its poorer disturbance response. MPC still results in this case in smaller amplitudes of the transient errors at t = 0.07 sec and 0.235 sec. The effect of higher friction is additional damping to the vibration mode with MPC and results in a comparable damping to the LQG case. In figure 32 the results are comparable to those from figure 30, with an error amplitude and speed of decay with LQG better than with MPC. From t=0.015 sec to about 0.035 sec, a constant position error due to stiction can be seen. For the case of MPC without acceleration feedback, the tracking error was improved, as seen in figures 33 to 35, by applying the acceleration signal directly in a feedback loop.

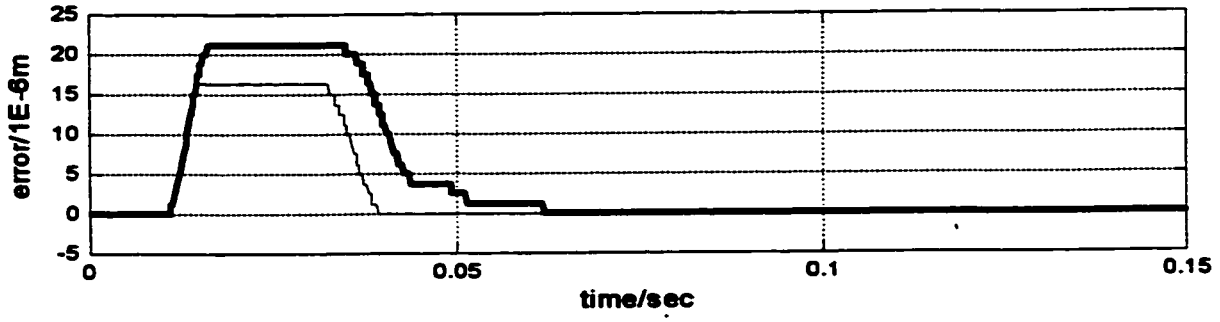


Figure 32: Simulated response of position error to step disturbance input for MPC (thick) and LQG control (thin) and increased friction level.

Figures 33, 34 and 37, 38 show the experimental results for the tracking error in the case of a MPC controller without AF. The graphs with experimental results from figures 33, 34 and 37, 38 correspond to the simulation results from figures 29 to 32 and confirm the validity of the model used for the simulation. The differences in the maximum amplitudes of the position error to step disturbance, shown in figures 32 and

38, can be explained by different initial conditions in the experiments and simulations and by friction parameters in the model which were slightly different from the real values.

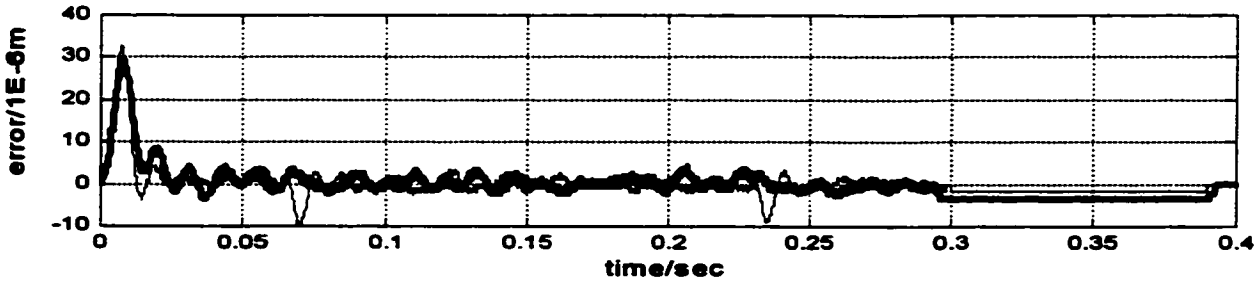


Figure 33: Experimental response of position error to reference input for MPC (thick) and LQG control (thin) and normal friction level

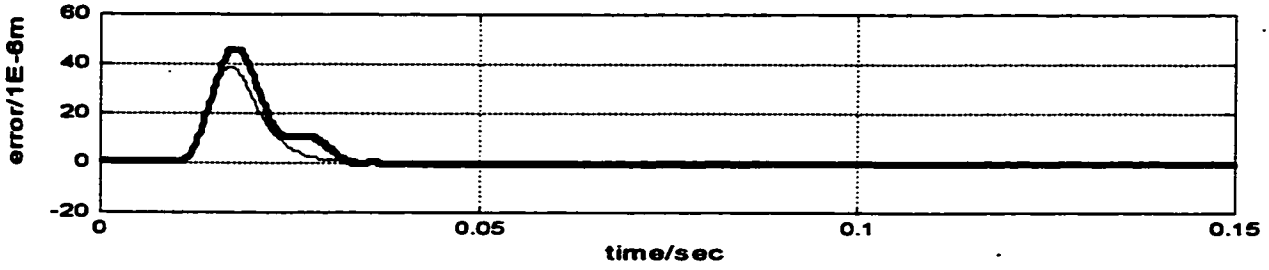


Figure 34: Experimental response of position error to step disturbance input for MPC (thick) and LQG control (thin) and normal friction level.

In DMC/AF, an observer to improve the estimates of position and velocity used the acceleration signal. Finally, both approaches were tested on the same simulation even though the results did not improve over the limits of each method. Variations of the prediction horizons and the sampling rate also contributed to a better tracking error and smoother variations of u_{servo}, u_{tacho}.

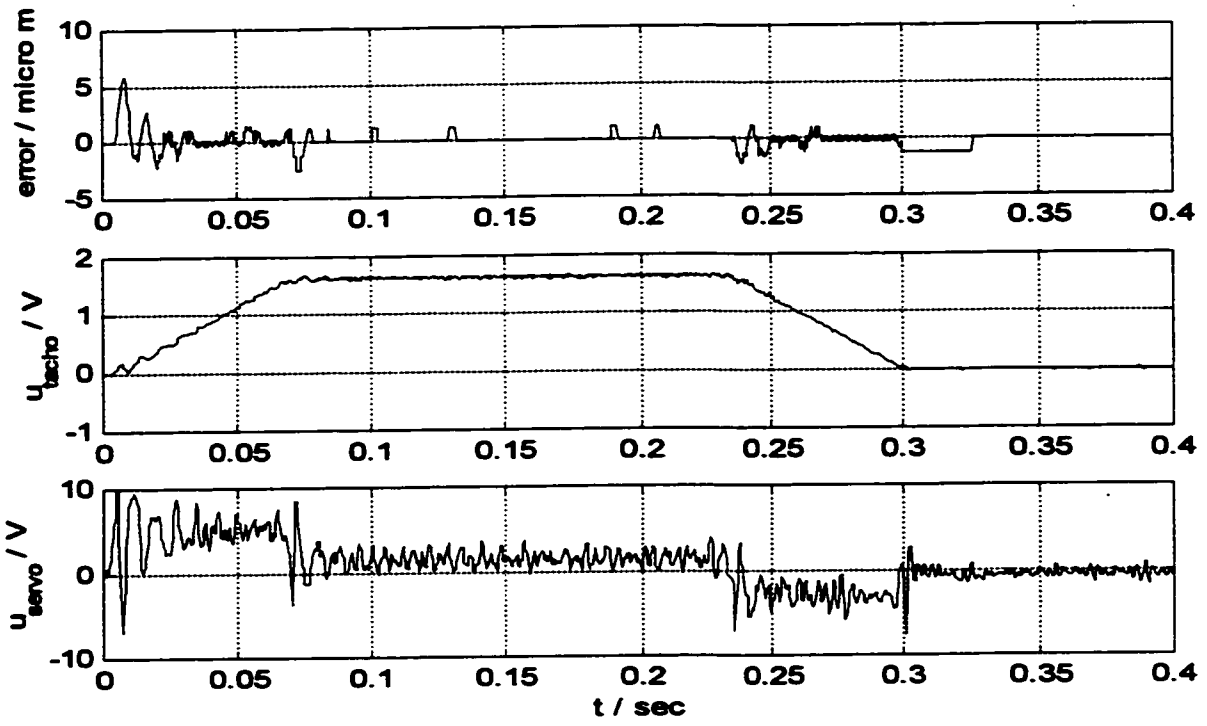


Figure 35: Simulated response of position error, u_{servo} and u_{tacho} for MPC with acceleration feedback to the observer, prediction horizon P=8, and sampling period ts = 5e-4 sec.

Compared to the simulations for MPC without acceleration feedback or LQG control (Figure 29) an important improvement was observed in error tracking: from about 20 μm with LQG and 10 μm with MPC, to slightly over 6 μm in DMC/AF.

The results with classical simple PID [2], shown in Figure 7 have a position error with a transient maximum over 60 μm. Compared to PID results, MPC and MPC/AF show significant improvements. The best simulation results were obtained for the DMC/AF with a prediction horizon P=15 and sampling period ts=2e-4 sec, presented in figure 36. The control signal was very smooth, the position error stays in the same range but somehow not as linear as previously (there is always the trade off while choosing the weights matrix and trying the optimal prediction horizon). The simulation for the extended state space is shown in Fig 37 with some improvements up to a factor of 1.2.

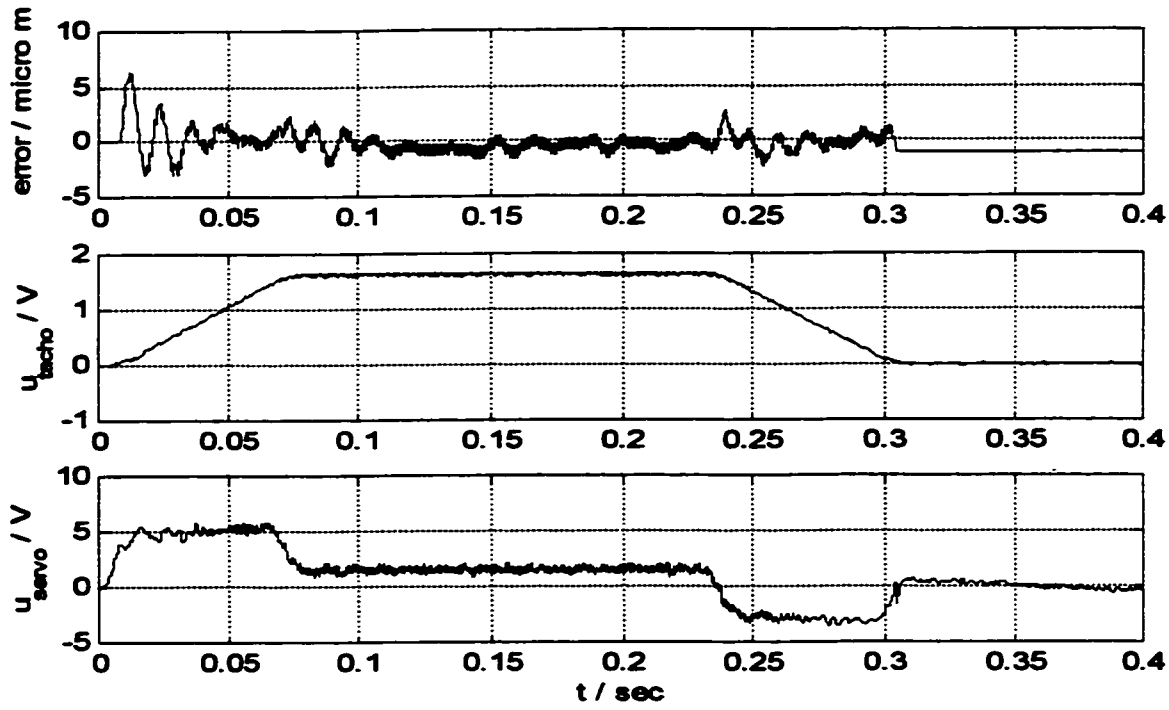


Figure 36: Simulated response of position error, u_{servo} and u_{tacho} for MPC with acceleration feedback to the observer, prediction horizon P=15, and sampling period ts = 2e-4 sec.

The extended state space model, including the drive and load side accelerations and the derivative of the drive side motor momentum dM_d/dt, was developed based on a similar use of derivatives in input-output linearization approach [9]. The error is even smaller in Figure 37, but the experiments showed some reduced improvements. In simulation the error even dropped under 5 μm with enough smoothness on the control signal and a small delay at the start of the process. The experiments were less precise because of the high level of noise and again due to the non-collocation of the sensors. Figure 37 shows the best results obtained in the simulation using both drive and load side accelerations and the derivative of the momentum dM_d/dt.

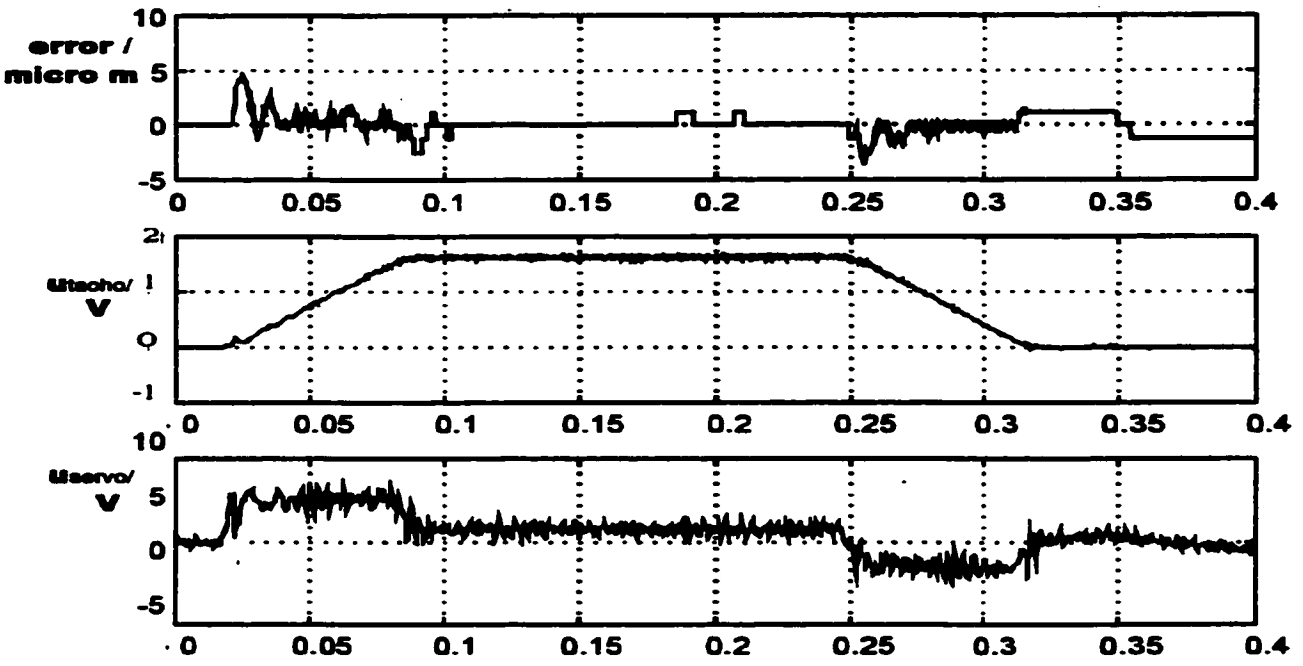


Figure 37: Simulated response of position error, u_{servo} and u_{tacho} for DMC/AF with acceleration feedback to the observer, prediction horizon P=12, and sampling period ts = 3e-4 sec.

4.4.2- Results of Simulations with MATLAB MPC Toolbox

The simulations were carried out only for step inputs in order to obtain a set of results different from the results presented in chapter 4.2 for the reference profile shown in Figure 28. The MATLAB program listings are given in Appendix. The MATLAB MPC toolbox was used for three different controllers described in Appendix A:

A1) PID; A2) MPC with velocity and position measurements; A3) MPC with velocity and position measurements used for position and velocity feedback and acceleration measurement for state variables observer. The comments in these MATLAB files, following %, explain the details of the procedures in these files.

The results obtained permit to compare the positioning precision with both methods (the MPC suite for DMC/AF and the Matlab MPC toolbox approach) and have

shown minimal differences. It appeared that the DMC/AF suite may have a better approach for setting up the weighting matrices and, consequently the positioning precision with the suite was slightly higher. The system was tested first by studying the step responses, the controllability, and then the constraints were set to the same functions from the profile generator used by the suite. Figure 38 below shows the position and velocity response of EMPS to unit step open loop control u_{servo}. As expected, the velocity y1 response tends to stabilize to a steady state velocity while the position y2 increases

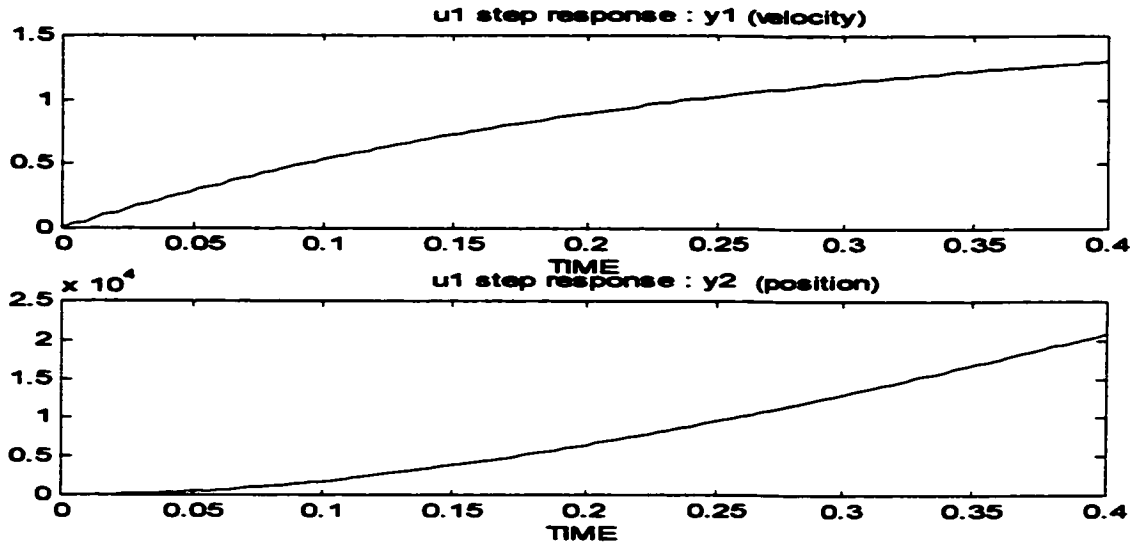


Figure 38: Open loop response of EMPS, position (y2) and velocity (y1) to unit step uservo.

Following the steps described in chapter 4.3, the subsequent simulations determine the responses of the closed-loop system to unit step inputs in setpoints for the outputs using equal and then unequal weights. This way was determined the interdependence between the outputs, and the possibility of improving one of the output parameters at the expense of the other. Later, the same procedure has been applied to the system with 3 outputs, including acceleration, in different versions. Differences in the responses of the system in each of these cases were identified. The disturbances have been considered unmeasured

M_{frd} (friction on the drive side) and measured M_l+M_{frl} (friction and load-side momentum). The results shown in Figures 39 to 44 are for the controller applied to a linear plant model. Figure 39 shows the response of the closed-loop system to the unit step set point for velocity for equal weighting, and 40 displays the same response for the unit step set point in position for different weighting.

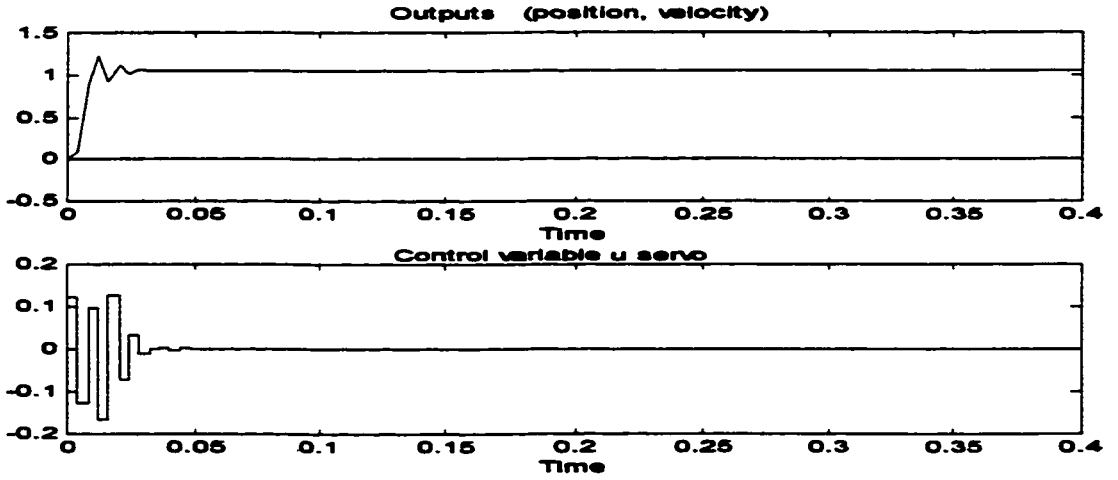


Figure 39: Velocity response and control variable ($u_{\_servo}$) of the closed-loop system to unit step in velocity setpoint for equal weights.

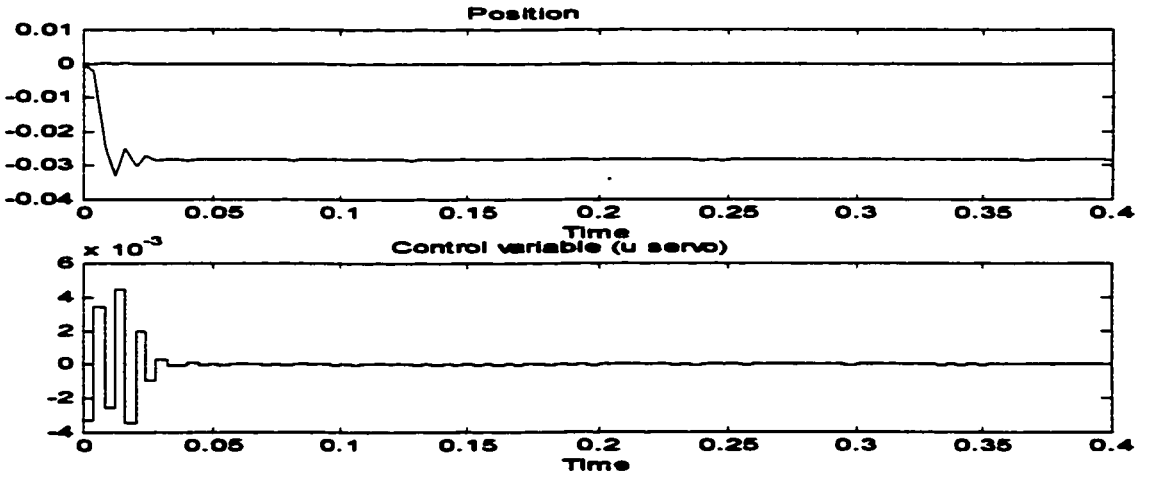


Figure 40: Position response for the closed-loop system and control variable u_servo to unit step position setpoint for unequal output weights and constraint on position (limit at 0.03m).

The results for the unit step reference input when the velocity influence was evaluated through the manipulation of the gains are shown in Figure 40. Both outputs were evaluated when one of them was set to zero to determine the influence of each in the controller.

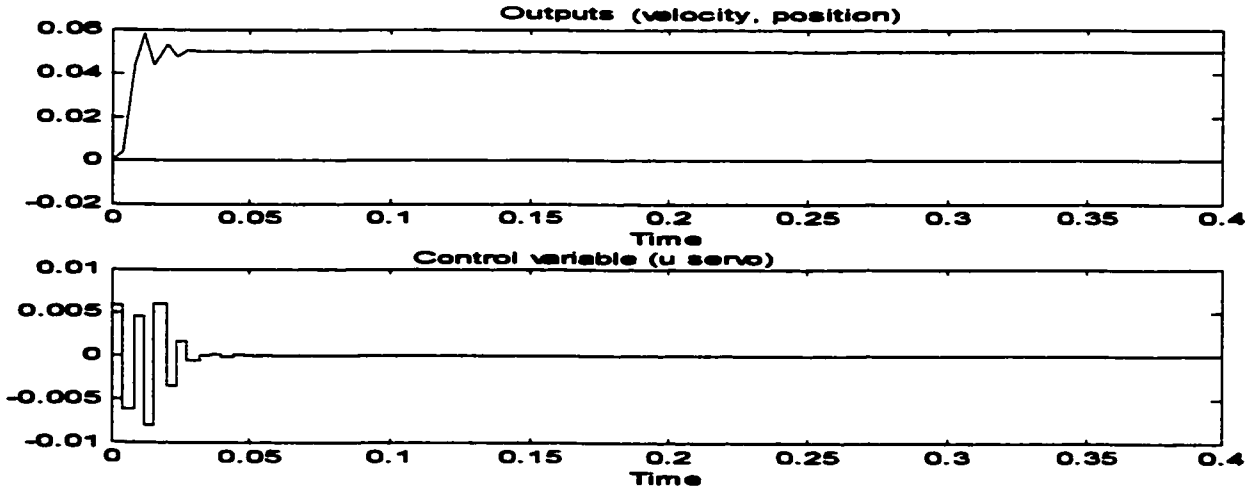


Figure 41: Response of the closed-loop system to unit step velocity setpoint for unequal weights.

Figures 42-45 show the responses of the closed-loop system with a default DMC-type state estimator and with a Kalman estimator to the unit step input in unmeasured disturbance keeping the unequal output weights and one output setpoint at 0. The system using the Kalman estimator displayed better reaction to the disturbances and a faster reaction than the default estimator. The results of tests carried out for the controller applied to a nonlinear plant model are presented in Figures 45 to 46. The nonlinear model is described in Appendix A4, where the step size for numerical integration, the initial values of the plant state and control parameter, and different sampling periods are indicated. The steps have been reworked separately with acceleration feedback in observer or directly calculated in distinct cases where acceleration is and is not a state.

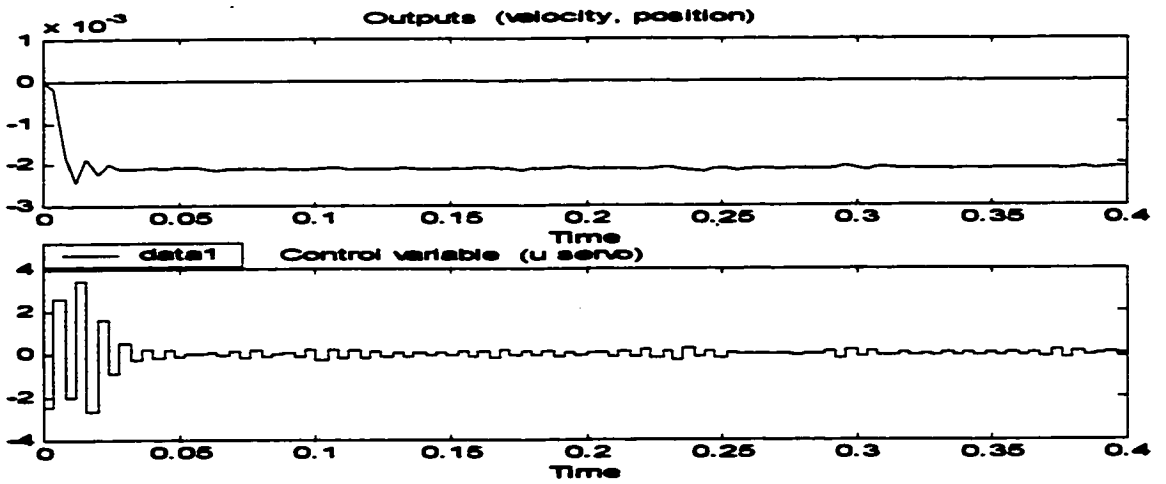


Figure 42: Response of the closed-loop system to unit step unmeasured disturbance Mfrd default estimator and unequal weights.

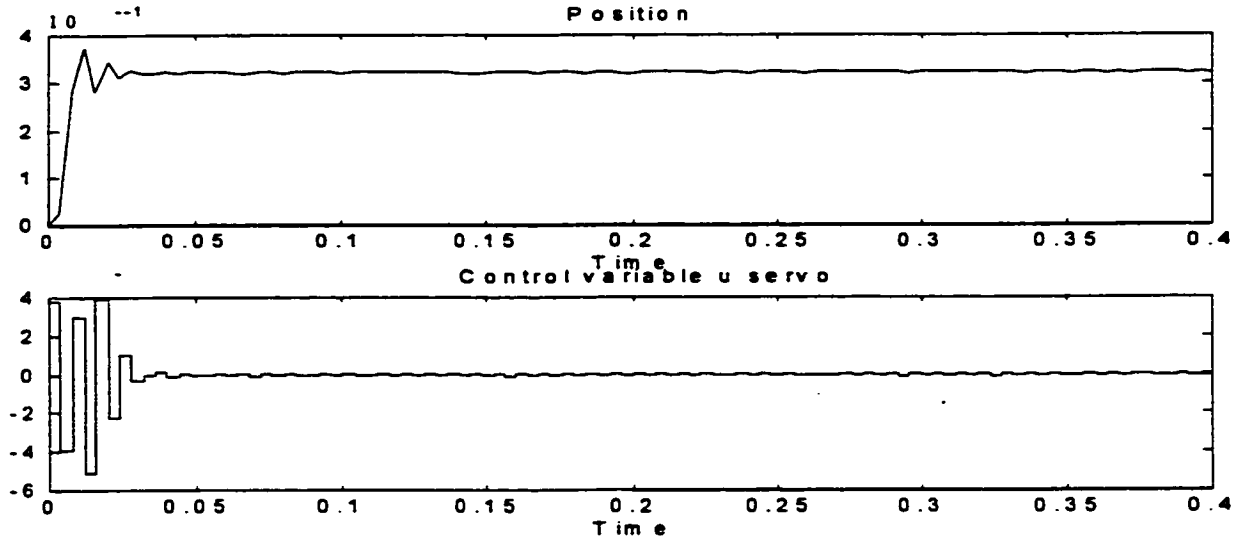


Figure 43: Position response of the closed-loop system to unit step measured disturbance and unequal weights.

Some of the resulting responses are presented below. The comparation was carried out for the two approaches, with the MPC suite and with the Matlab toolboxes. At some values for the disturbance (from 7-8 and up to 12 times higher than expected in practice) the system becomes unstable and the magnitude of the velocity output increases indefinitely (Figure 45).

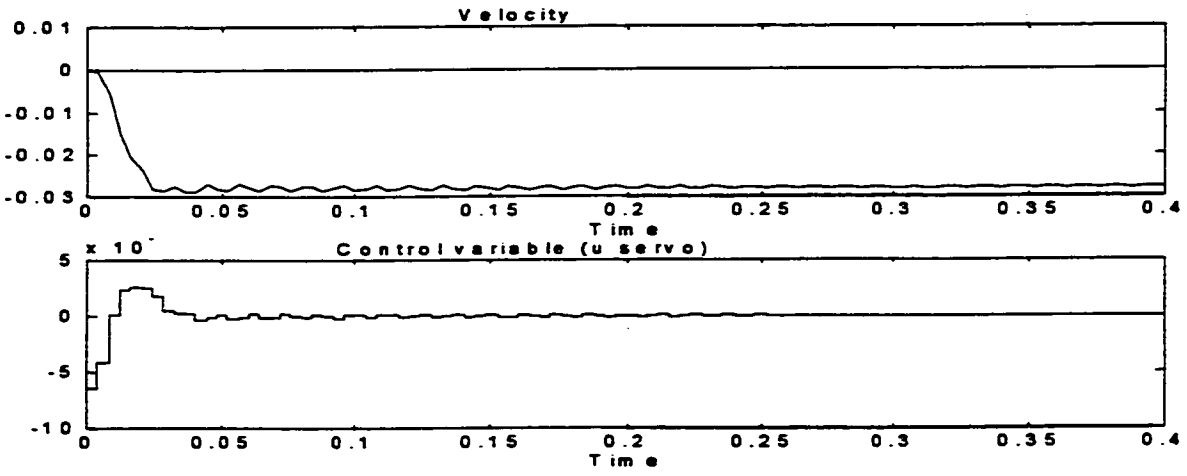


Figure 44: Velocity response of the closed-loop system to unit step position setpoint with constraints and unequal weights.

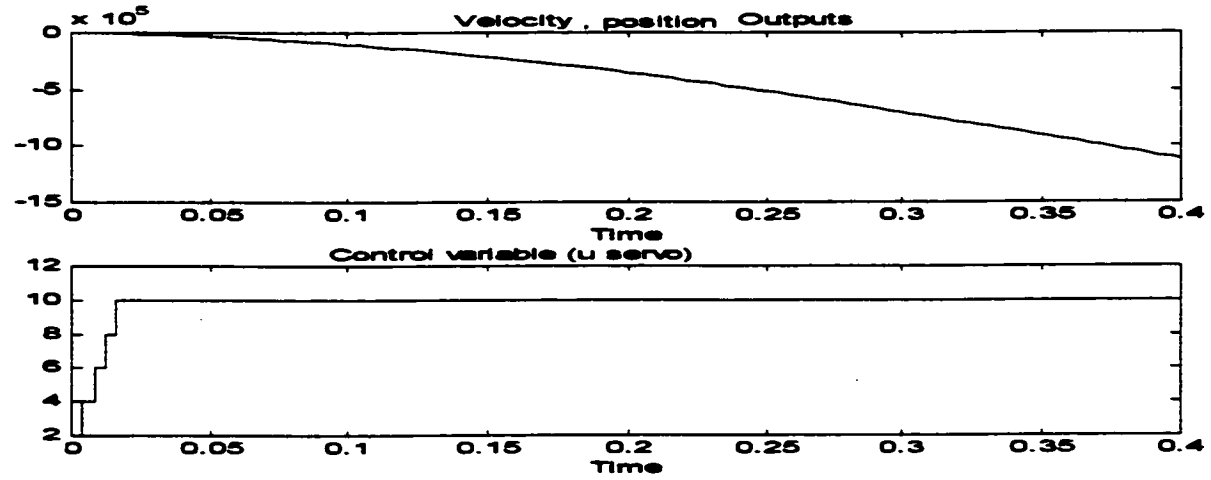


Figure 45: Response of the closed-loop system to unit step in unmeasured disturbance with constraints and unequal weightings (the system becomes unstable).

The introduction of acceleration feedback shows a slight improvement (Figure 46), visible and close to what the MPC suite foresaw, i.e. a position error of maximum 7.5 μm in Figure 47 for step position input vs. less than 5 μm in Figure 48. This can be explained by the noise in signal, less suitable values introduced for the weighting matrices, and a better evaluation of the friction in the MPC suite using Simulink.

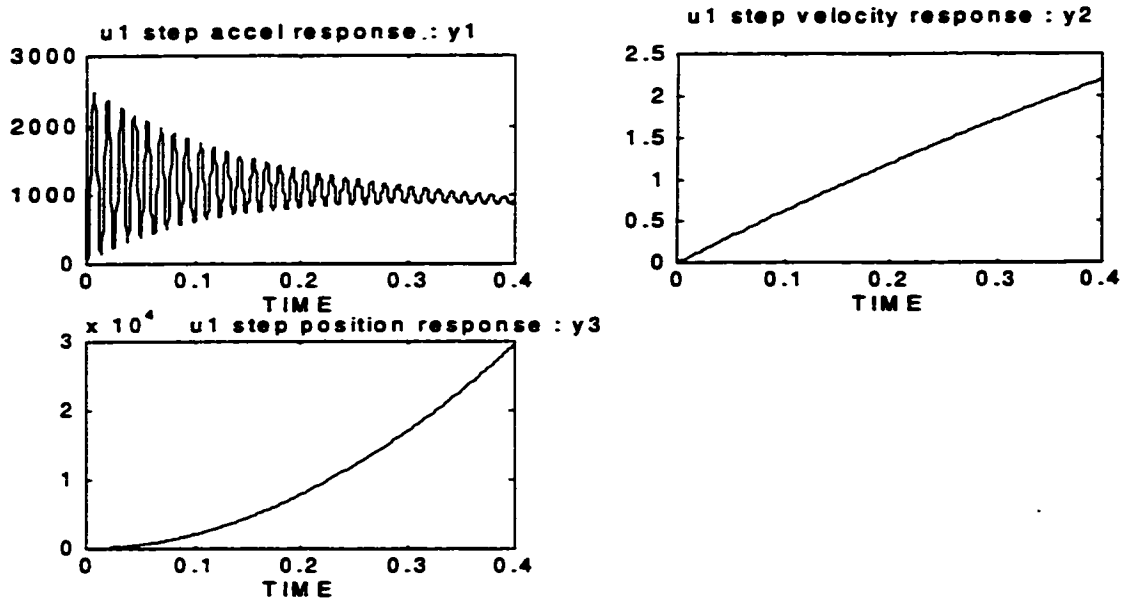


Figure 46: Acceleration (y1), position (y3) and velocity (y2) response of EMPS to unit step u_{servo}.

The similarity of the results obtained using the Simulink based suite (from chapter

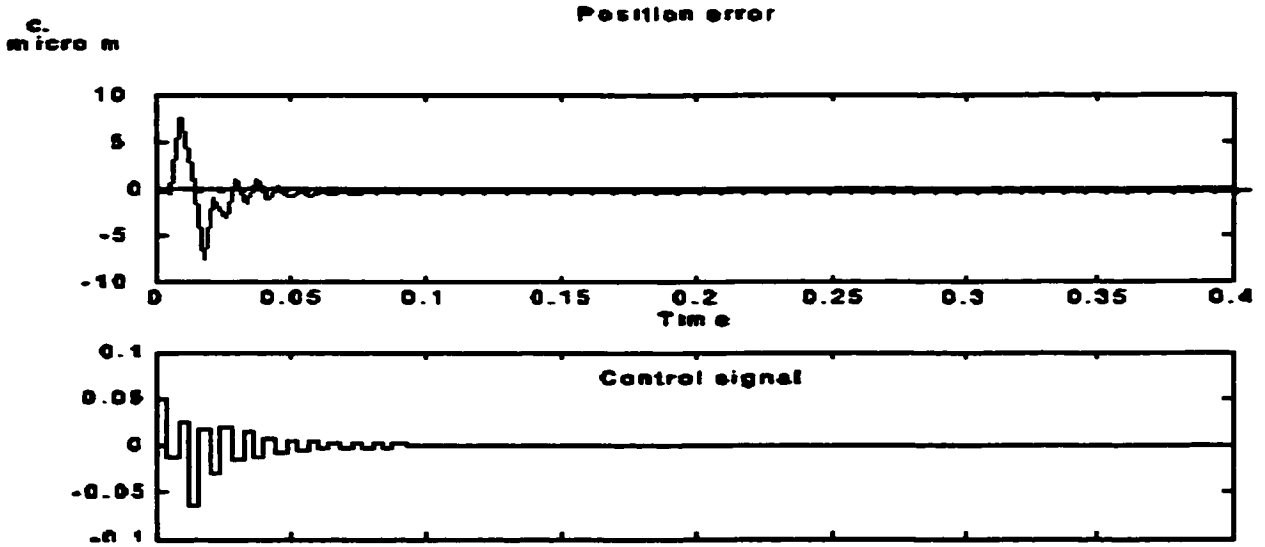


Figure 47: Tracking position error obtained in simulation with acceleration feedback in observer

4.2) and the results obtained using the MATLAB MPC Toolbox, presented in this chapter, confirms that the programming for the MPC controllers and the nonlinear models was correct. The control law given by equation 4.11 was implemented by auto

generation similar to that shown in Appendix A1. The same reference and disturbance inputs to the system were used. The disturbance input was applied, as described in section 1, by the equivalent torque M_l, with the current controlled load motor shown in Figure1.

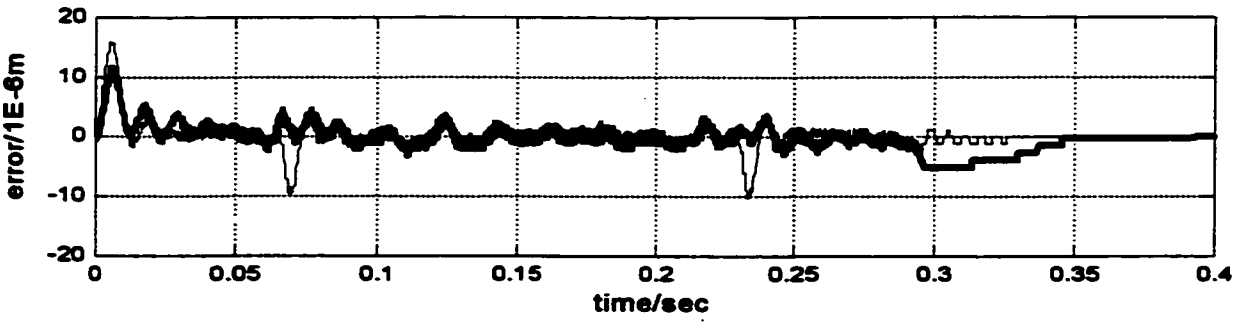


Figure 48: Experimental response of position error to reference input for MPC (thick) and LQG control (thin) and increased friction level

The error tracking using the Matlab/Toolbox method has not introduced the reference model used by the suite, it only applied the desired value for the position as a condition in

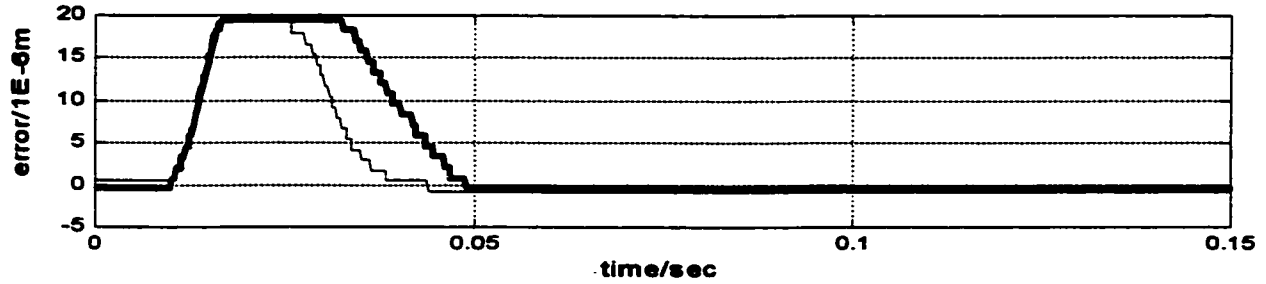


Figure 49: Experimental response of position error to step disturbance input for MPC (thick) and LQG control (thin) and increased friction level

the formulation of the restraints used by the program (Annex A4). In this regard, the shapes of the u_{servo}, u_{tacho} did not appeared, only the error of positioning in micro m was obtained to be compared with that previously found with the suite (chapter 4.2). The results show very similar values giving confirmation of the method. The use of the Matlab MPC Toolbox proved to be a very useful way to validate a more detailed model used by the suite (regarding especially the friction and the noise phenomena).

V – CONCLUSIONS AND RECOMMANDATIONS

1. Acceleration feedback proved lately to be an important means for achieving precise positioning in complex mechanical systems. The use of acceleration feedback has become more attractive in recent years for very diverse applications, especially for systems with significant friction, disturbances or non-linear components. Earlier problems with explicit acceleration feedback in adaptive control have been solved and significant attention was given lately to this feedback in designing robot related control systems. The use of acceleration feedback can improve the tracking performance in many aspects, for example with regard to positioning, force/torque control, robustness or other key performance parameters. For robotic manipulators, or other types of motor driven positioning systems, the use of acceleration signal has become an important means in smoothening the movements, increasing the precision and reducing the oscillations that occur in the interactions with the environment. In many applications, the criterion function to be minimized was modified to include the effect of acceleration feedback and it proved to be an important factor in improving specific aspects regarding performance requirements. The inclusion of acceleration measurements has provided good results for different areas of control from the classical methodology to the optimal, robust and predictive control. With the analytical tool characteristic to each of these domains for different particular applications, various mathematical approaches have been developed to make explicit the influence of the acceleration feedback and to improve the performance of various non-linear systems.

2. For servomechanisms, used often in precise positioning, the potential of improvement through the use of acceleration measurements has been explored and

different methods of implementing have been proposed, specific to the domain in which the application required higher levels of performance. The introduction of acceleration in control schemes has been demonstrated to be advantageous. So far, however, there is no generic method for designing controllers with acceleration feedback. At present various options are used for calculating these gains. These approaches depend on the analytical models, robustness requirements and the type of control approach (optimal, adaptive, predictive controls, etc). Experiments are often conducted to test controllers and to improve certain parameters of the system such as precision positioning, force/torque control, and for eliminating oscillatory regimes, or increasing robustness.

3. In the ElectroMechanical Drive System (EMPS) application presented in this thesis, the acceleration signal has been used in a feedback scheme directly and, in another configuration, with an observer providing better estimates for the output variables. Each of these two methods provided higher accuracy in positioning compared to the results obtained with previous published methods. Given that system is nonlinear and that multiple factors present in the system are affecting a smooth, precise functioning (such as friction, vibrations, instrument tuning, or signal processing influence), the use of the acceleration was appropriate and satisfied the expectations. The DMC (Dynamic Matrix Control) form of MPC, chosen in this thesis, proved to be very efficient in achieving rapid and precise position control of an industrial EMPS. Compared to LQG, for the same drive system and for the same simulation and experimental parameters, MPC showed lower transient errors in tracking a reference position profile with stepwise changes of acceleration. It also displayed good steady-state position accuracy but somewhat lower

transient disturbance rejection in both situations developed, with normal friction and with increased level.

LQG control, verified experimentally on the same system, had better results than those obtained with classical control methods. MPC and MPC/AF have increased further the performance of the EMPS, especially with respect to the positioning precision. While maintaining the rest of performance characteristics of the system, referring to stability and robustness, the DMC/AF proved effective in achieving better damping of the vibrations. Previously, with simple MPC, a lower damping of the compliant system mode resulted, while DMC/AF further increased damping. The control signal has been applied as a constant signal first, then as a variable one over the prediction horizon; the variable control parameter proved more effective.

4. Acceleration feedback has been introduced, as previously mentioned, directly (in a system with no acceleration in the state space model) and indirectly (through an observer for improving the position and velocity estimations). The use of AF was more effective than an increase in the sampling rate and showed an improvement in tracking accuracy by a factor of up to 20%. Acceleration feedback improved the disturbance rejection of internal friction and external disturbing torques. The main factors affecting adversely the use of acceleration measurements have been the disturbances in the signal, the phase lag introduced by the signal processing devices, the time delay created by the sampled data implementation of the controllers, and the non-collocation of the sensors.

5. Future directions for studying precise positioning, using predictive control methods, with good possibilities for improvement, could be, for non-linear systems, the addition of load-side acceleration measurement and of other types of feedback from state

derivative measurements. Processes involving significant inertia-related nonlinearities or even magnetic levitation systems, with very low levels of friction, can also benefit from the use of MPC/AF. The potential for improvement in systems with constraints, offered particularly by predictive control methods, the openness in applying the methodology to different domains and the relative reduced complexity of the design methods for obtaining the gains, could also offer the opportunity to improve control systems used in precise positioning.

REFERENCES

[1] Henrichfreise H., Necsulescu D., Paehle M., Verbit S., "Prototyping of a MPC Compensator for a Compliant Positioning System with Friction and Comparison to a LQG Control", accepted for publication in "Machine Intelligence and Robotic Control" (Japan).

[2] Henrichfreise H, and Oedekoven S., "Practical Issues on Classical and Modern Control of Electromechanical Drive Systems", Technical Report CLM, Cologne, September, 1998.

[3] Necsulescu D., and De Carufel J., "Predictive Control of Actuators in the Presence of Dry Friction", pp.122-135, Seminar IA-EPFL, May 13, 1997.

[4] Tomizuka M., and Lee H., "Robust Motion Controller Design for High-Accuracy Positioning Systems", IEEE Transactions on Industrial Electronics, Vol. 43, No.1, pp. 481-502, February, 1996.

[5] Tomizuka M., and Ishikawa J., "Pivot Friction Compensation Using an Accelerometer and a Disturbance Observer for Hard Disk Drives", IEEE Transactions on Mechatronics, Vol. 3, pp. 118-134, No.3 September, 1998.

[6] Bram de Jager, "Acceleration Assisted Tracking Control", IEEE Control Systems, Vol. 4, pp.20-27, October 1994.

[7] Xu L., and Han J., "Experimental Study of Contact Transition Control Incorporating Joint Acceleration Feedback", IEEE/ASME Transactions on Mechatronics, Vol. 5, No. 3, pp. 560-583, September 2000.

[8] Friedland B., "Control System Design", McGraw-Hill, 1986.

[9] Slotine J., Weiping L., "Applied Nonlinear Control", Prentice Hall, 1991.

[10] Berlin F., and Frank P.M., "Design and Realization of a MIMO Predictive Controller for a a3-tank System", Advances in Model Based Predictive Control, Oxford University Press, pp. 144-48, 1994.

[11] Soeterboek R., "Predictive Control – A Unified Approach", Prentice Hall, 1992.

[12] Camacho E.F., Bordons C., "Model Predictive Control in Process Industry", Springer Verlag Berlin, 1995.

[13] Ohnishi K., Shibata M., Murakami T., "Motion Control for Advanced Mechatronics",IEEE/ASME Transactions on Mechatronics, Vol. 6, pp. 312-320, March, 1996.

[14] Ishikawa J., Tomizuka M., "Pivot Friction Compensation Using an Accelerometer and a Disturbance Observer for Hard Disk Drives", IEEE/ASME Transactions on Mechatronics, Vol.7, pp.412-419, September, 1998.

[15] Umeno T., Hori Y., "Robust Speed Control of DC Servomotors Using Modern Two Degrees-of-Freedom Controller Design", IEEE/ASME Transactions on Mechatronics, Vol. 4, pp. 86-95, October , 1991.

[16] Lee H. S., Tomizuka M., "Robust Motion Controller Design for High-Accuracy Positioning Systems", IEEE/ASME Transactions on Mechatronics, Vol. 8, pp. 486-492, February, 1996.

[17] D. S. Necsulescu, J. De Carufel, C. Canudas De Wit, "Investigation on the Efficiency of Acceleration Feedback in Servomechanism with Friction", Dynamics and Control, #7, Kluwer Academic Publishers, Boston 1997.

[18] K. Ogata, "Modern Control Engineering", Prentice Hall, New Jersey 1997.

[19] D.W. Clarke, "Application of Generalized Predictive Control to Industrial Processes", IEEE Contr. Syst. Mag., vol.8, pp. 49-55, 1988.

[20] J. Richalet, "Industrial Applications of Model Based Predictive Control," Automatica, vol. 29, no. 8, pp. 1251-1274, 1993.

[21] I. D. Landau, J. Langer, D. Rey, and J. Barnier, "Robust Control of a 360° Flexible Arm Using the Combined Pole Placement/Sensitivity Function Shaping Method," IEEE Trans. Contr. Syst. Technol., vol. 4, no.4, pp. 369-383, 1996.

[22] M. Kinnaert, "Adaptive Generalized Predictive Control for MIMO Systems," Int. J. Contr., vol. 50, no. 1, pp. 161-172, 1989.

[23] C. Vaucoret and J. Bordeneuve-Guibe, "Robust Multivariable Predictive Control: Application to an Industrial Test Stand," in Proc. 5th IEEE Conf. Control Applications, Trieste, Italy, 1998.

APPENDIX A1

PID controller design using MATLAB and Control System Toolbox

```
%% transfer function of PID controller with k_i = 1
t_r = 6*t_servo;  %% 1/t_r sufficiently low
t_n = 8*t_r;
numc = [ t_n*t_n  2*t_n  1 ];
denc = [ t_r  1  0 ];

%% transfer function of EMPS from u_servo to incr
[nump,denp] = ss2tf(a_p,b_p,c_p,d_p,1);
nump = nump(2,:);

%% open-loop transfer function for k_i = 1
num = conv(numc,nump);
den = conv(denc,denp);

%% place crossover between logarithmic frequencies
%% for maximum phase margin
log_w_n = log10(1/t_n);
log_w_r = log10(1/t_r);
w_c = 10^((log_w_n+log_w_r)/2);

%% find index of crossover frequency
w = logspace(-1,4,500);
i = 1;
while w(i) < w_c
  i = i + 1;
end

%% open-loop bode plot for k_i = 1
[mag,phase] = bode(num,den,w);
%% controller gains
k_i = 1/mag(i);
```

APPENDIX A2

Matlab File used for the Linear System Simulation with Position and Velocity Feedback Only

```
% EMPS with basic design, 2 outputs measurements (position, velocity),
% no acceleration feedback
% Variables taken from MPC suite

echo off

% Matrices of the linearized EMPS model
% Matrices of the EMPS plant inputs
%
%      u_servo     V
%      M_l+M_frl   Nm
%      M_frd       Nm
%
%    measurement outputs
%      u_tacho     V
%      count       incremental encoder counter value
%
%    states
%      phi_d       rad
%      omega_d     rad/sec
%      phi_l       rad
%      omega_l     rad/sec
%      M_d         Nm
%
% parameters

c_dl=0.72;                              % Nm/rad
b_dl=3.84e-5;                           % Nm/(rad/sec)
b_d=1.0e-6;                             % Nm/(rad/sec)
j_d=3.0e-7 + 69.6e-7 + 4.2e-7;          % kg*m^2
b_l=4.7e-5;                             % Nm/(rad/sec)
j_l=7.5e-7 + 16.1e-7 + 1.0e-7 + 2*4.2e-7 + 10.4e-7 + 0.4e-7;      %g*m^2
t_servo=0.0005;                         % sec
k_ia = 0.3;                             % A/V   (3.0A/10V)
k_servo=0.0525*k_ia;                    % Nm/V  (0.0525Nm/A*k_ia);
k_acc = 1;                              % 1 Vsec^2/m
k_omega=0.52/104.72;                    % 0.52V/1000 rev/min ^= 0.52V/104.72
rad/sec
k_phi=2000/(2*pi);                      %pulses/rad

T_s=0.004;

% matrix coefficients

a21 = -c_dl/j_d;
a22 = -(b_d+b_dl)/j_d;
a23 = c_dl/j_d;
a24 = b_dl/j_d;
a25 = 1/j_d;
```

```
a41 = c_dl/j_l;
a42 = b_dl/j_l;
a43 = -c_dl/j_l;
a44 = -(b_dl+b_l)/j_l;
a55 = -1/t_servo;

b51 = k_servo/t_servo;
b42 = -1/j_l;
b23 = -1/j_d;


% matrices

A = [  0    1    0    0    0
      a21  a22  a23  a24  a25
       0    0    0    1    0
      a41  a42  a43  a44   0
       0    0    0    0   a55 ];

B = [  0    0    0
       0    0   b23
       0    0    0
       0   b42   0
      b51   0    0 ];

C = [ 0    k_omega    0    0    0
      0      0      k_phi   0    0 ];

D = [ 0  0  0
      0  0  0];

% Discretize the linear model and save in MOD form.
% Then calculate & plot step responses.

dt=T_s;

[PHI,GAM]=c2dmp(A,B,dt);
minfo=[dt,5,1,1,1,2,0];
imod=ss2mod(PHI,GAM,C,D,minfo);
plotstep(mod2step(imod,0.4))
pause


% Define controller parameters
P=15;                                % Prediction horizon
M=P;                                 % Control horizon
ywt=[0,1];                           % 1 weighting of y2, no control of y1
uwt=0.34*[1];                        % 1 weighting of u.
ulim=[-10*[1] 10*[1] 2*[1]];         % Constraints on u_servo
ylim=[-inf   -inf   inf   inf
      -inf   -inf   inf   inf
      -inf   -inf   inf   inf
      -inf   -inf   inf   0.03
      -inf   -inf   inf   inf];      % constraints on y setpoint .03m
Kest=[];                             % Default estimator

% Simulation using SCMPC -- no model error
```

```
pmod=imod;              % plant and internal model are identical
setpts=[0 1];           % servo response to step in y2 setpoint
tend=0.4;               % duration of simulation
z=[];                   % measurement noise
v=0;                    % measured disturbance
d=0;                    % unmeasured disturbance

[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 1'), pause
% Repeat for step in setpoint of y1.

setpts=[1 0];       % response to step in y setpoint (velocity)
[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 2'), pause


% Change the output weighting to allow better control of y2 (position)
% at the expense of y1 (velocity).

ywt=[0.05 1];
            % Unequal output weighting (however look for equalweights)
setpts=[1 0];       % response to step in y1 setpoint(velocity)
[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 3'), pause


% Test the new design for a step in setpoint of y2 (position).


ywt=[1 0.005];          % Unequal output weighting
setpts=[0 1];           %  response to step in y2 setpoint position)
[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
              setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 4'), pause


% Test the design for a step in the measured disturbance.

setpts=[0 0];           % Hold outputs at zero
z=[];                   % measurement noise
v=1;                    % unit step in measured disturbance
d=0;                    %unmeasured disturbance
[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 5'), pause


% Test the design for a step in the unmeasured disturbance.
```

```
v=0;                    % Measured disturbance is now zero
d=1;                    % Unit step in unmeasured disturbance
[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 6'), pause


% Design an alternative to the default state estimator.
% This one is the optimal Kalman estimator for the modeled
% disturbance.

Q=10;
R=1*eye(2);
                            %R=1*diag([1 1]);
Kest=smpcest(imod,Q,R);
                            %use scmpc no model error
setpts = [0 0];
d =1;                       %unmeasured disturbance
[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 7'), pause


% Try another alternative estimator.
% This one is based on the output-disturbance model.

taus=[0.005 0.005];          % Time constants of output disturbances
signoise=[100 100];          % Signal-to-noise for each output
[Kest,newmod]=smpcest(imod,taus,signoise);

[y,u, ym]=scmpc(pmod,newmod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 8'), pause
```

APPENDIX A3

Matlab File for the Nonlinear System Simulation with Position and Velocity Feedback and Observer Using Acceleration Measurement

```
% This is the model for EMPS in non linear form
% Matrices of the linearized EMPS model
% parameters

c_dl=0.72;                              % Nm/rad
b_dl=3.84e-5;                           % Nm/(rad/sec)
b_d=1.0e-6;                             % Nm/(rad/sec)
j_d=3.0e-7 + 69.6e-7 + 4.2e-7;          % kg*m^2
b_l=4.7e-5;                             % Nm/(rad/sec)
j_l=7.5e-7 + 16.1e-7 + 1.0e-7 + 2*4.2e-7 + 10.4e-7 + 0.4e-7;   % kg*m^2
```

```
t_servo=0.0005;                 % sec
k_ia = 0.3;                     % A/V    (3.0A/10V)
k_servo=0.0525*k_ia;            % Nm/V   (0.0525Nm/A*k_ia);
k_acc = 1;                      % 1 Vsec^2/m
k_omega=0.52/104.72;            % 0.52V/1000 rev/min ^= 0.52V/104.72 rad/sec
k_phi=2000/(2*pi);              %pulses/rad

T_s=0.004;

% matrix coefficients

a21 = -c_dl/j_d;
a22 = -(b_d+b_dl)/j_d;
a23 = c_dl/j_d;
a24 = b_dl/j_d;
a25 = 1/j_d;
a41 = c_dl/j_l;
a42 = b_dl/j_l;
a43 = -c_dl/j_l;
a44 = -(b_dl+b_l)/j_l;
a55 = -1/t_servo;

b51 = k_servo/t_servo;
b42 = -1/j_l;
b23 = -1/j_d;

% matrices

A = [  0    1    0    0    0
      a21  a22  a23  a24  a25
       0    0    0    1    0
      a41  a42  a43  a44   0
       0    0    0    0   a55 ];

B = [  0    0    0
       0    0   b23
       0    0    0
       0   b42   0
      b51   0    0 ];

C = [ 0    k_omega    0    0    0
      0      0      k_phi   0    0 ];

D = [ 0  0  0
      0  0  0];



% Discretize the linear model and save in MOD form.

dt=T_s;
[PHI,GAM]=c2dmp(A,B,dt);
minfo=[dt,5,1,1,1,2,0];
imod=ss2mod(PHI,GAM,C,D,minfo);

% Define controller parameters.  The controller design is the same
```

```
% as for the linear case.

P=8;                                    % Prediction horizon
M=P;                                    % Control horizon
ywt=[0.023,1];                          % unequal weighting of y1 and y2
uwt=0.34*[1];                           % weighting 1 of u
ulim=[-10*[1 1] 10*[1 1] 2*[1 1]];      % Constraints on u
ylim=[-inf   -inf   inf   inf
      -inf   -inf   inf   inf
      -inf   -inf   inf   inf
      -inf   -inf   0.03  inf
      -inf   -inf   inf   inf];         %  constraints on y

% Estimator design

Q=10;
R=1*diag([1 1]);
Kest=smpcest(imod,Q,R);

% Simulation using SCMPCNL command as in Matlab toolbox.  We first try
% a step in the setpoint of y1(velocity)

pmod='EMPS';      % plant is nonlinear ODE model

%pmod='EMPS';     % plant is nonlinear ODE model -- Simulink version

setpts=[1 0];     % servo response to step in y1 setpoint
tend=0.4;         % duration of simulation
z=[];             % measurement noise
v=0;              % measured disturbance is zero initially
d=0;              % unmeasured disturbance is zero initially
du=[];            % No unmeasured disturbance at inputs;
x0=zeros(5,1);    % initial state of nonlinear plant
u0=zeros(3,1);    % initial manipulated variables & measured disturbance
stepsize=0.005;   % for numerical integration.

% the warning messages in the following simulations
% are warning that SCMPCNL has made a decision that might
% not be correct in general, but it is for this case.

[y,u,ym,x]=scmpcnl(pmod,imod,ywt,uwt,M,P,tend, ...
                   setpts,ulim,ylim,Kest,z,v, ...
                   d,du,x0,u0,stepsize);
plotall(y,u,dt)
disp('Hit any key to continue 1'), pause

% This is essentially the same procedure as for the
% linear case in Appendix A1; the same is true for a change in the y2
% setpoint,
% and the responses to the measured and unmeasured disturbances


setpts=[0 1];       % servo response to step in y2 setpoint(position)
[y,u,ym,x]=scmpcnl(pmod,imod,ywt,uwt,M,P,tend, ...
                   setpts,ulim,ylim,Kest,z,v, ...
                   d,du,x0,u0,stepsize);
plotall(y,u,dt)
```

```
pause
setpts=[0 0];          % hold outputs at zero
v=1;                   % unit step in measured disturbance
[y,u,ym,x]=scmpcnl(pmod,imod,ywt,uwt,M,P,tend, ...
                        setpts,ulim,ylim,Kest,z,v, ...
                        d,du,x0,u0,stepsize);
plotall(y,u,dt)
pause


% To make the nonlinear effects more obvious, a
% step in the unmeasured disturbance is magnified 10 times
% larger than that used in the linear simulation.

setpts=[0 0];        % hold outputs at zero
v=0;                 % no measured disturbance
d=10;                % larger step in unmeasured disturbance
[y,u,ym,x]=scmpcnl(pmod,imod,ywt,uwt,M,P,tend, ...
                        setpts,ulim,ylim,Kest,z,v, ...
                        d,du,x0,u0,stepsize);
plotall(y,u,dt)
disp('Hit any key to continue 2'), pause

% the response is different from the linear case
% a larger disturbance will give

d=12;                    % even larger step in unmeasured disturbance
[y,u,ym,x]=scmpcnl(pmod,imod,ywt,uwt,M,P,tend, ...
                        setpts,ulim,ylim,Kest,z,v, ...
                        d,du,x0,u0,stepsize);
plotall(y,u,dt)
disp('Hit any key to continue 3'), pause

% controllability  has been almost lost; acceptable performance can
% be restored by "de-tuning" the controller; the conventional
% way is to increase the penalty on input variable moves;

uwt=10*[1 1];        % de-tune controller by increasing penalty on
                     % manipulated variable moves.
tend=1;              % increase simulation time.
[y,u,ym,x]=scmpcnl(pmod,imod,ywt,uwt,M,P,tend, ...
                        setpts,ulim,ylim,Kest,z,v, ...
                        d,du,x0,u0,stepsize);
plotall(y,u,dt)
disp('Hit any key to continue 4'), pause


% the disturbance rejection could be improved by
% a reduction in the sampling period; the output
% disturbance estimator design is the same as in the linear
% case

dt=0.00025;                          % change sampling period
[PHI,GAM]=c2dmp(A,B,dt);             % get new discrete-time linear model
minfo=[dt,5,1,1,1,2,0];              % replace old model
imod=ss2mod(PHI,GAM,C,D,minfo);
uwt=0.34*[1];                        % restore original weighting of u
```

```
taus=[0.05 0.05];          % time constants of output disturbances
signoise=[10 10];          % signal-to-noise for each output
[Kest,newmod]=smpcest(imod,taus,signoise);
d=1;                       % return to original disturbance magnitude
[y,u,ym,x]=scmpcnl(pmod,newmod,ywt,uwt,M,P,tend, ...
                        setpts,ulim,ylim,Kest,z,v, ...
                        d,du,x0,u0,stepsize);
plotall(y,u,dt)

% Disturbance rejection has improved partially
```

APPENDIX A4

Matlab File used for the Simulation of the Linear System with Position, Velocity and Acceleration and Feedback

```
% EMPS with basic linear design, 3 output measurements (position,
% velocity and acceleration)
% acceleration feedback through observer
% Variables taken from MPC suite

echo off
% M-FUNCTION
%    Set up linear continuous state space model of EMPS.
%
% SYNTAX
%    [a,b,c,d] = empsabcd
%
% OUTPUT PARAMETERS
%    a .. d  Model state space matrices.
%
% DESCRIPTION
%    The EMPS is modeled by two compliantly coupled inertias.
%    A first order lag system generates the driving motor torque from
%    the servo amplifier input voltage.
%    Its inputs, outputs and state variables are:
%
%    inputs
%      u_servo        V
%      d(M_l+M_frl)
%      ------------   Nm/sec
%          dt
%      M_frd          Nm
%
%    measurement outputs
%      u_acc      V
%      u_tacho    V
%      count      incremental encoder counter value
%
%    states
%      phi_d      rad
%      omega_d    rad/sec
%      phi_l      rad
%      omega_l    rad/sec
```

```
%       alpha_l     rad/sec^2
%       M_d         Nm

% parameters
c_dl=0.72;                              % Nm/rad
b_dl=3.84e-5;                           % Nm/(rad/sec)
b_d=1.0e-6;                             % Nm/(rad/sec)
j_d=3.0e-7 + 69.6e-7 + 4.2e-7;          % kg*m^2
b_l=4.7e-5;                             % Nm/(rad/sec)
j_l=7.5e-7 + 16.1e-7 + 1.0e-7 + 2*4.2e-7 + 10.4e-7 + 0.4e-7;      %
kg*m^2
t_servo=0.0005;                         % sec
k_ia = 0.3;                             % A/V    (3.0A/10V)
k_servo=0.0525*k_ia;                    % Nm/V   (0.0525Nm/A*k_ia);
k_acc = 1;                              % 1 Vsec^2/m
k_omega=0.52/104.72;          % 0.52V/1000 rev/min ^= 0.52V/104.72rad/sec
k_phi=2000/(2*pi);                      %pulses/rad

T_s=0.004;

% matrix coefficients

a21 = -c_dl/j_d;
a22 = -(b_d+b_dl)/j_d;
a23 = c_dl/j_d;
a24 = b_dl/j_d;
a25 = 1/j_d;
a41 = c_dl/j_l;
a42 = b_dl/j_l;
a43 = -c_dl/j_l;
a44 = -(b_dl+b_l)/j_l;
a55 = -1/t_servo;

b51 = k_servo/t_servo;
b42 = -1/j_l;
b23 = -1/j_d;

a21 = -c_dl/j_d;
a22 = -(b_d+b_dl)/j_d;
a23 = c_dl/j_d;
a24 = b_dl/j_d;
a26 = 1/j_d;
a51 = -b_dl*c_dl/(j_d*j_l);
a52 = -b_dl*(b_dl+b_d)/(j_d*j_l) + c_dl/j_l;
a53 = b_dl*c_dl/(j_d*j_l);
a54 = b_dl^2/(j_d*j_l) - c_dl/j_l;
a55 = -(b_d+b_l)/j_l;
a56 = b_dl/(j_d*j_l);
a66 = -1/t_servo;

b61 = k_servo/t_servo;
b52 = -1/j_l;
b23 = -1/j_d;
```

```
% matrices

A = [  0    1    0    0    0    0
      a21  a22  a23  a24   0   a26
       0    0    0    1    0    0
       0    0    0    0    1    0
      a51  a52  a53  a54  a55  a56
       0    0    0    0    0   a66 ];

B = [  0    0    0
       0    0   b23
       0    0    0
       0    0    0
       0   b52   0
      b61   0    0 ];

C = [ 0      0        0     0   k_acc   0
      0   k_omega     0     0     0     0
      0      0      k_phi   0     0     0 ];

D = [ 0  0  0
      0  0  0
      0  0  0 ];


% Discretize the linear model and save in MOD form.
% Then calculate & plot step responses.

dt=T_s;

[PHI,GAM]=c2dmp(A,B,dt);
minfo=[dt,6,1,1,1,3,0];
imod=ss2mod(PHI,GAM,C,D,minfo);
plotstep(mod2step(imod,0.4))
pause


% Define controller parameters

P=15;               % Prediction horizon
M=P;                % Control horizon
ywt=[1,0,1];        % 1 weighting of y1,y3   no control of y2
uwt=0.34*[1];       % 1 weighting of u.
ulim=[-10*[1] 10*[1] 2*[1]];      % Constraints on u_servo
ylim=[];            %-inf   -inf   inf   inf  inf inf
                    %-inf   -inf   inf   inf  inf inf
                    %-inf   -inf   inf   inf  inf inf
                    %-inf   -inf   inf   inf  inf inf
                    %-inf   -inf   inf   inf  inf inf
                    %-inf   -inf   inf   inf  inf inf
                    %-inf   -inf   inf   inf  inf 0.03
                    %-inf   -inf   inf   inf  inf inf];
                    % constraints on y setpoint .03 m

Kest=[];            % Default estimator
```

```
% Simulation using SCMPC -- no model error

pmod=imod;              % plant and internal model are identical
setpts=[1 0 0];         % servo response to step in y1 setpoint
tend=0.4;               % duration of simulation
z=[];                   % measurement noise
v=0;                    % measured disturbance
d=0;                    % unmeasured disturbance

[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
                setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 1'), pause


% Repeat for step in setpoint of y3.

setpts=[0 0 1];         % response to step in y3 setpoint (position)
[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 2'), pause


% Change the output weighting to allow better control of y3 (position)
% at the expense of y1 (acceleration).

ywt=[0.05 0 1];         % Unequal output weighting (however look for
equal weights)
setpts=[1 0 0];         % response to step in y1 setpoint(velocity)
[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 3'), pause


% Test the new design for a step in setpoint of y3 (position).


ywt=[0.005 0 1];        % Unequal output weighting
setpts=[0 0 1];         %  response to step in y3 setpoint (position)
[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 4'), pause


% Test the design for a step in the measured disturbance.

setpts=[0 0 0];         % Hold outputs at zero
z=[];                   % measurement noise
v=1;                    % unit step in measured disturbance
d=0;                    %unmeasured disturbance
[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 5'), pause
```

```
% Test the design for a step in the unmeasured disturbance.

v=0;                    % Measured disturbance is now zero
d=1;                    % Unit step in unmeasured disturbance
[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 6'), pause


% Design an alternative to the default state estimator.
% This one is the optimal Kalman estimator for the modeled
% disturbance.

Q=10;
R=1*eye(3);
                                %R=1*diag([1 1]);
Kest=smpcest(imod,Q,R);
                                %use scmpc no model error
setpts = [0 0 0];
d =1;                           %unmeasured disturbance
[y,u,ym]=scmpc(pmod,imod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 7'), pause


% Try another alternative estimator.
% This one is based on the output-disturbance model.

taus=[0.005 0.005 0.005];   % Time constants of output disturbances
signoise=[10 10 10];
% Signal-to-noise for each output
[Kest,newmod]=smpcest(imod,taus,signoise);

[y,u, ym]=scmpc(pmod,newmod,ywt,uwt,M,P,tend, ...
               setpts,ulim,ylim,Kest,z,v,d);
plotall(y,u,dt)
disp('Hit a key to continue 8'), pause
```

APPENDIX A5

Matlab Files used for the Simulation of the DC positioning servomechanism in the Example, ch.4.4.1

MPCExample1.m

```
% This file designs a MPC controller for a simple system DC positioning
% servomotor without constraints and without disturbances

% The system is that of Problem A-4-3 (K. Ogata-Modern Control
% Engineering p.189)
```

```
% The transfer function for the system is tf=42.3/s^2+7.69s+42.3
% the open loop system is stable

echo on

% The system will be developed throughout the tutorial in MPC based on
% step response models
% Define the process model: the system is a SISO without delay and
% disturbances

T=0.1;
ny =1;
G=poly2tfd(42.3,[1 7.69 42.3],0,0);
pmod=tfd2mod(T,1,G);
pause

% MPC design -- "Perfect Controller"

imod=pmod; % assume perfect modeling
ywt=[];    % default (unity) weight on output
uwt=[];    % default (zero) weight input
P=5;       % prediction horizon
M=P;       % control horizon
Ks=smpccon(imod,ywt,uwt,M,P);

% Simulate unit step in setpoint of output and plot results.

tend=10;   % time period for simulation.
r=[1];     % setpoints for the two outputs.
[y,u]=smpcsim(pmod,imod,Ks,tend,r);
plotall(y,u,T)

[clmod,cmod]=smpccl(pmod,imod,Ks);
smpcpole(cmod)

pause

% Increase the prediction horizon and decrease the control horizon
% to increase the smoothness of the control signal

P=10;
M=3;
Ks=smpccon(imod,ywt,uwt,M,P);
[y,u]=smpcsim(pmod,imod,Ks,tend,r);
plotall(y,u,T)

pause

% Increase smoothness even more using "blocking"

M=[2 3 4];                          % Defines 3 blocks of control moves
Ks=smpccon(imod,ywt,uwt,M,P);
[y,u]=smpcsim(pmod,imod,Ks,tend,r);
plotall(y,u,T)

pause
```

```
% Check the regulatory response of this latest controller

ulim=[];    % default (no) constraints on u variable.
Kest=[];    % default (DMC) state estimator.
r=[0];      % Output setpoint at zero.
z=[];       % default (zero) measurement noise.
v=[];       % default (zero) measured disturbances.
w=[1];      % unit-step in unmeasured disturbance.
[y,u]=smpcsim(pmod,imod,Ks,tend,r,ulim,Kest,z,v,w);
plotall(y,u,T)
```

MPCExample2.m

```
% This file designs a MPC controller for a simple system DC positioning
% servomotor without constraints and with disturbances
% The system is that of Problem A-4-3 (K. Ogata-Modern Control
% Engineering p.189)
% The transfer function for the system is tf=42.3/s^2+7.69s+42.3
% (closed-loop), the
% open loop system is stable

echo on

% The system will be developed throughout the tutorial in State-Space
% MPCC
% Define the process model: the system is a SISO without delay and
% disturbances

T=0.1;
ny =1;
G=poly2tfd(42.3,[1 7.69 42.3],0,0);
pmod1=tfd2mod(T,ny,G);

%Disturbance model: hd=1.2e^(-0.2s)/0.14s+1

Gd=poly2tfd(1.2,[0.14 1],0,0.2);
dmod=tfd2mod(T,ny,Gd);
pmod=addumd(pmod1,dmod);        %Combines the model with the disturbance

pause

% MPC design -- "Perfect Controller"

imod=pmod; % assume perfect modeling
ywt=[];     % default (unity) weight on output
uwt=[];     % default (zero) weight input
P=5;        % prediction horizon
M=P;        % control horizon
Ks=smpccon(imod,ywt,uwt,M,P);

% Simulate unit step in setpoint of output and plot results.

tend=10;    % time period for simulation.
r=[1];      % setpoints for the two outputs.
[y,u]=smpcsim(pmod,imod,Ks,tend,r);
```

```
plotall(y,u,T)

[clmod,cmod]=smpccl(pmod,imod,Ks);
smpcpole(cmod)
pause

% Increase the prediction horizon and decrease the control horizon
% to increase the smoothness of the control signal

P=10;
M=3;
Ks=smpccon(imod,ywt,uwt,M,P);
[y,u]=smpcsim(pmod,imod,Ks,tend,r);
plotall(y,u,T)

pause

% Increase smoothness even more using "blocking"

M=[2 3 4];                          % Defines 3 blocks of control moves
Ks=smpccon(imod,ywt,uwt,M,P);
[y,u]=smpcsim(pmod,imod,Ks,tend,r);
plotall(y,u,T)

pause

% Check the regulatory response of this latest controller

ulim=[];    % default (no) constraints on u variable.
Kest=[];    % default (DMC) state estimator.
r=[0];      % Output setpoint at zero.
z=[];       % default (zero) measurement noise.
v=[];       % default (zero) measured disturbances.
w=[1];      % unit-step in unmeasured disturbance.
[y,u]=smpcsim(pmod,imod,Ks,tend,r,ulim,Kest,z,v,w);
plotall(y,u,T)

pause
% See if we can improve the transient regime using a Kalman filter
% instead of DMC type

[Kest,newmod]=smpcest(imod,[15],[3]);
Ks=smpccon(newmod,ywt,uwt,M,P);
[y,u]=smpcsim(pmod,newmod,Ks,tend,r,ulim,Kest,z,v,w);
plotall(y,u,T)
pause
```

MPCExample3.m

```
% This file designs a MPC controller for a simple system DC positioning
% servomotor with constraints and without disturbances
% This demo illustrates the QP solution of the constrained MPC problem.
```

```
echo on

% First set up a 1 by 1 internal model of the plant:
% Define the process model in the MOD Format.

% The system will be developed throughout the tutorial in State-Space
% MPC
% Define the process model: the system is a SISO without disturbances
% but with constraints

T=1;
ny =1;
G=poly2tfd(42.3,[1 7.69 42.3],0,0);
imod=tfd2mod(1,T,G);

pause

% Base-case MPC tuning parameters:

nhor=10;          % Prediction horizon.
ywt=[];           % Unity weighting on output tracking errors (default).
uwt=[];           % Zero weighting on man. variable moves (default).
blks=[2 3 5];     % Allows 3 moves of manipulated variable.
K=[];             % DMC-type state estimation (default).


ulim=[-inf inf 10];
ylim=[];

% Define the input for the simulation.  We will look at the
% effect of a step change in the setpoint of output

setpts=[0.8];  % Define the step in the setpoint.

% Assume a perfect model, i.e., set the plant model equal
% to the internal model.

plant=imod;                       % First run an unconstrained simulation.
Ks=smpccon(imod,ywt,uwt,blks,nhor);

tend=20;                          % Duration of the simulation
[y1,u1]=smpcsim(plant,imod,Ks,tend,setpts,ulim,K);

% Run the constrained simulation and plot the results.

[y,u]=scmpc(plant,imod,ywt,uwt,blks,nhor,tend, ...
                setpts,ulim,ylim,K);

plotall([y y1],[u u1],T);

pause

% Add upper bound of 0.7 on output, accounting for 3 samples time delay

ylim =[-inf  inf
       -inf  inf
       -inf  inf
```

```
       -inf  0.7];

[y,u]=scmpc(plant,imod,ywt,uwt,blks,nhor,tend, ...
               setpts,ulim,ylim,K);
plotall(y,u,T);
plant=imod;                    % First run an unconstrained simulation.
Ks=smpccon(imod,ywt,uwt,blks,nhor);

tend=20;                       % Duration of the simulation
[y1,u1]=smpcsim(plant,imod,Ks,tend,setpts,ulim,K);

% Run the constrained simulation and plot the results.

[y,u]=scmpc(plant,imod,ywt,uwt,blks,nhor,tend, ...
               setpts,ulim,ylim,K);

plotall([y y1],[u u1],T);

pause

% Since model error and disturbances are both absent in the
% above case, the output constraints are never violated.  We
% can speed up the calculations by only enforcing the constraints
% at one sampling period.  We must account for the minimum delays of
% 3 periods for the output.
% Let's try an upper bound of 0.7 on output
% and an lower bound of -4.5 both to be enforced only at the first
% sampling period following the minimum delay.


ylim=[-inf inf
      -inf 0.7
      -inf 0.7
      -4.5 0.7
      -inf 0.7];

[y,u]=scmpc(plant,imod,ywt,uwt,blks,nhor,tend, ...
               setpts,ulim,ylim,K);

plotall(y,u,T); pause;
plant=imod;                    % First run an unconstrained simulation.
Ks=smpccon(imod,ywt,uwt,blks,nhor);

tend=20;                       % Duration of the simulation
[y1,u1]=smpcsim(plant,imod,Ks,tend,setpts,ulim,K);

% Run the constrained simulation and plot the results.

[y,u]=scmpc(plant,imod,ywt,uwt,blks,nhor,tend, ...
               setpts,ulim,ylim,K);

plotall([y y1],[u u1],T);

pause

% Notice that none of the constraints have been violated.
% Now add some constraints on the manipulated variable.
```

```
% Again, to speed up the calculations we will only constrain
% them for the first 2 sampling period in the prediction horizon.

ulim=[ -10.5 0.6 0.5
       -10.5 0.6 0.5];

% Note that the limits on delta_u must always be finite.  Making
% them too large leads to problems with roundoff error in the
% QP calculations.

[y,u]=scmpc(plant,imod,ywt,uwt,blks,nhor,tend, ...
                setpts,ulim,ylim,K);

plotall(y,u,T)

plant=imod;                         % First run an unconstrained simulation.
Ks=smpccon(imod,ywt,uwt,blks,nhor);

tend=20;                            % Duration of the simulation
[y1,u1]=smpcsim(plant,imod,Ks,tend,setpts,ulim,K);

% Run the constrained simulation and plot the results.

[y,u]=scmpc(plant,imod,ywt,uwt,blks,nhor,tend, ...
                setpts,ulim,ylim,K);

plotall([y y1],[u u1],T);

pause

echo off
```