ISSN 0280-5316
ISRN LUTFD2/TFRT--5715--SE

# Iterative Feedback Tuning with Application to Robotics

Alessandro Bindi

Department of Automatic Control
Lund Institute of Technology
December 2003

**Department of Automatic Control**
**Lund Institute of Technology**
**Box 118**
**SE-221 00 Lund Sweden**

*Document name*
MASTER THESIS

*Date of issue*
December 2003

*Document Number*
ISRN LUTFD2/TFRT--5715--SE

*Author(s)*
Alessandro Bindi

*Supervisor*
Rolf Johansson and Anders Robertsson at Department of Automatic Control in Lund.
Edoardo Mosca at Universitá di Firenze.

*Sponsoring organization*

*Title and subtitle*
Iterative Feedback Tuning with Application to Robotics. (Iterativ trimning av informationsåterföring med robotiktillämpning).

*Abstract*
Many control objectives can be expressed in terms of a criterion function. Generally, explicit solutions to such optimization problem require full knowledge of the plant and disturbances and complete freedom in the complexity of the controller. In practice, the plant and the disturbances are seldom known, and it is often desirable to achieve the best possible performance with a controller of prescribed complexity such as for example a PID controller. The optimization of such control performance criterion typically requires
iterative gradient-based minimization procedures. The major stumbling block for the solution of this optimal control problem is the computation of the gradient of the criterion function with respect to the controller parameters: it is a fairly complicated function of the plant and disturbance dynamics.
When these are unknown, it is not clear how this gradient can be computed. Iterative Feedback Tuning (IFT) is a input-output data-based design method for the tuning of restricted complexity controllers. It does not depend on the plant model, utilizes I/O data only. Therefore IFT is robust against the plant model uncertainty.
At each iteration, an update for the parameters of the controller is estimated from data obtained partly from the normal operation of the closed loop system and partly from a special experiment. No identification procedure is involved. In this thesis tuning of robot joint controllers using IFT is considered.
The different IFT-schemes have been verified in simulation and in real experiments on an industrial robot manipulator ABB Irb-2000.

*Keywords*

*Classification system and/or index terms (if any)*

*Supplementary bibliographical information*

*ISSN and key title*
0280-5316

*ISBN*

*Language*
English

*Number of pages*
110

*Recipient's notes*

*Security classification*

To my family

# Contents

# Acknowledgments

I would like to thank Professor Rolf Johansson for the welcome, the availability and the time that he gave me during this work. His humanity and professionality will be with me forever.

I would like to thank Dr. Anders Robertsson for the support and continuous help that he gave me for the theoretical part and moreover for the days and nights in which we worked together in the robot laboratory where we also formulated a theory summarized as “the best results come between midnight and five in the morning”. He is a person that cares about everyone and I will never forget him.

I would also like to thank my Italian supervisor Professor Edoardo Mosca for his advice to set out on this experience in Lund and also for his kind availability during these months.

I am also grateful to all the people of the department of Automatic Control at Lund Institute of Technology for their help and support.

# 1. Introduction

Many control objectives can be expressed in terms of a criterion function.
Generally, explicit solutions to such optimization problem require full knowledge of the plant and disturbances and complete freedom in the complexity of the controller. In practice, the plant and the disturbances are seldom known, and it is often desirable to achieve the best possible performance with a controller of prescribed complexity. For example, one may want to tune the parameters of a PID controller in order to extract the best possible performance from such simple controller.

The optimization of such control performance criterion typically requires iterative gradient-based minimization procedures. The major stumbling block for the solution of this optimal control problem is the computation of the gradient of the criterion function with respect to the controller parameters: it is a fairly complicated function of the plant and disturbance dynamics. When these are unknown, it is not clear how this gradient can be computed.

The contribution of [Hjalmarsson, Gunnarsson, Gevers, 1994] was to show that an unbiased estimate of the gradient can be computed from signals obtained from closed loop experiments with the present controller operating on the actual system.

For a controller of given (typically low-order) structure, the minimization of the criterion is then performed iteratively by a Gauss-Newton based scheme.

For a two-degree-of-freedom controller, three batch experiments are to be performed at each step of the iterative design. The first and third simply consist of collecting data under normal operating conditions; the only real experiment is the second batch which requires feeding back, at the reference input, the output measured during normal operation. Hence the acronym Iterative Feedback Tuning (IFT) given to this scheme. For a one-degree-of-freedom controller, only the first and second experiments are required. No identification procedure is involved.

As in any numerical optimization routine, a variable step size can be used. This allows one to control the rate of change between the new controller and the previous one and is an important aspect from an engineering perspective. Furthermore a variable step size is the key to establishing convergence of the algorithm under noisy conditions. With a step size tending to zero appropriately fast, ideas from stochastic averaging can be used to show that, under the condition that the signal remain bounded, the achieved performance will converge to a (local) minimum of the criterion function as the number of data tends toward infinity.

The optimal IFT scheme of [Hjalmarsson *et al*, 1994] was initially derived in 1994 and presented at the IEEE CDC 1994.

The IFT method is appealing to process control engineers because, under this scheme, the controller parameters can be successively improved without ever opening the loop.

In addition, the idea of improving the performance of an already operating controller, on the basis of closed-loop data corresponds to a natural way of thinking.

# 2. IFT Description

## 2.1 Two-degree-of-freedom controllers and IFT

The theory in this section is based on [Hjalmarsson, Gevers, Gunnarsson, Lequin, 1998].
We consider an unknow true system described by the discrete model

$$y_t = G_0 u_t + v_t \tag{1}$$

where $G_0$ is the linear time-invariant operator, $y$ is the measured output, $u$ is the control input, $\{v_t\}$ is an unmeasurable process disturbance and $t$ represents the discrete time instants (remark: to easy the notation the time index $t$ is sometimes left out).

We consider that this system is to be controlled by a two-degree-of-freedom (*2DOF*) controller:

$$u_t = C_r(\rho) r_t - C_y(\rho) y_t \tag{2}$$

where $C_r(\rho)$ and $C_y(\rho)$ are linear time-invariant transfer functions parametrized by some vector $\rho \in R^{n_\rho}$, and $\{r_t\}$ is an external deterministic reference signal, independent of $\{v_t\}$.
Notice that it is possible for $C_r(\rho)$ and $C_y(\rho)$ to have common parameters.

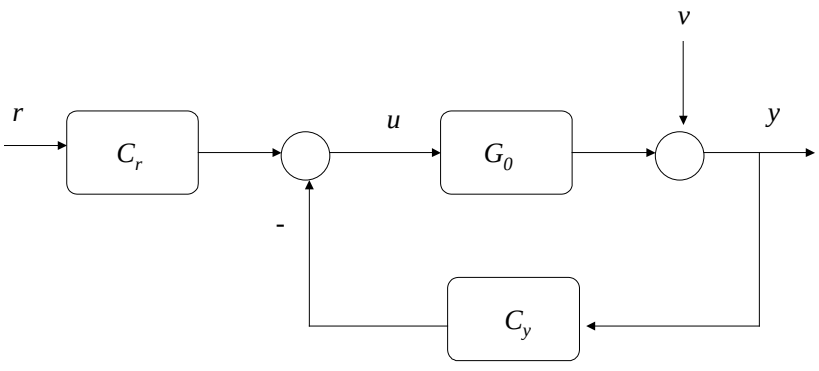



Fig. 2.1 A two-degree-of-freedom controller structure

We will use the notation $y_t(\rho)$ and $u_t(\rho)$ for the output and the control input of the system (1) in feedback with the controller (2), in order to make the dependence of these signals on the controller parameter vector $\rho$ explicit.

Let $y^d$ be a desired output response to a reference signal $r$ for the closed loop system. This response may be defined as the output of a reference model $T_d$ such that

$$y_t^d = T_d r_t \tag{3}$$

but for the IFT method knowledge of the signal $y_t^d$ is sufficient.
The error between the achieved and the desired response is

$$\tilde{y}(\rho) \triangleq y(\rho) - y^d = \left( \frac{C_r(\rho)G_0}{1 + C_y(\rho)G_0} r - y^d \right) + \frac{1}{1 + C_y(\rho)G_0} v$$

So, using (3):

$$\tilde{y}(\rho) \triangleq y(\rho) - y^d = \left( \frac{C_r(\rho)G_0}{1 + C_y(\rho)G_0} - T_d \right) r + \frac{1}{1 + C_y(\rho)G_0} v$$

This error consists of a contribution due to incorrect tracking of the reference signal $r_t$ and an error due to the disturbance $v$.

For a controller of some fixed structure parametrized by ρ, it is natural to formulate the control design objective as a minimization of some norm of $\tilde{y}(\rho)$ over the controller parameter vector ρ.
We will consider the following quadratic control performance criterion:

$$J(\rho) = \frac{1}{2N} E\left[ \sum_{t=1}^{N} (L_y \tilde{y}_t(\rho))^2 + \lambda \sum_{t=1}^{N} (L_u u_t(\rho))^2 \right] \tag{4}$$

where $N$ is the number of samples considered and E[·] denotes expectation with respect to the weakly stationary disturbance $v$.

The filter $L_y$ is a frequency weighting of the error between the desired response and the achieved response. The filter $L_u$ weights the penalty of the control effort. They can of course be set to 1, but they give added flexibility to the design.

The optimal controller parameter $\rho^*$ is defined by:

$$\rho^* = \arg \min_{\rho} J(\rho) \tag{5}$$

The objective of the criterion (4) is to tune the process response to a desired deterministic response of finite length $N$ in a mean square sense.

As formulated, this is a model reference problem with an additional penalty on the control effort.

Let $T_0(\rho)$ ans $S_0(\rho)$ denote the achieved closed loop response and sensitivity function with the controller $\{C_r(\rho), C_y(\rho)\}$:

$$T_0(\rho) = \frac{C_r(\rho)G_0}{1 + C_y(\rho)G_0}$$

$$S_0(\rho) = \frac{1}{1 + C_y(\rho)G_0} \ .$$

Given the independence of $r_t$ and $v_t$, $J(\rho)$ can be written as:

$$J(\rho) = \frac{1}{2N}\sum_{t=1}^{N}\left\{L_y(y_t^d - T_0(\rho)r_t)\right\}^2 + \frac{1}{2N}E\left[\left\{L_y S_0(\rho)v_t\right\}^2\right] + \\ + \lambda\frac{1}{2N}E\left[\sum_{t=1}^{N}(L_u u_t(\rho))^2\right]$$

The first term is the tracking error, the second term is the disturbance contribution, and the last term is the penalty on the control effort.

**CRITERION MINIMIZATION**

We examine now the minimization of the criterion (4) with respect to the control parameters vector $\rho$ for a controller of specified structure.

To obtain the minimum a necessary condition is that the first derivative of $J(\rho)$ w.r.t. the controller parameter $\rho$ is zero:

$$0 = \frac{\partial J}{\partial \rho}(\rho) = \frac{1}{N}E\left[\sum_{t=1}^{N}\tilde{y}_t(\rho)\frac{\partial \tilde{y}_t}{\partial \rho}(\rho) + \frac{\lambda}{N}\sum_{t=1}^{N}u_t(\rho)\frac{\partial u_t}{\partial \rho}(\rho)\right] \qquad (6)$$

where for simplicity we assume $L_y = L_u = 1$.

If the gradient $\frac{\partial J}{\partial \rho}(\rho)$ could be computed, then the solution of the previous equation would be obtained by the following iterative algorithm:

$$\rho_{i+1} = \rho_i - \gamma_i R_i^{-1}\frac{\partial J}{\partial \rho}(\rho_i) \qquad (7)$$

Here $\gamma_i$ is a positive real scalar that determines the step size and $R_i$ is some appropriate positive definite matrix (notice also that, as we will show later, $\gamma_i$

must obey some constraints for the algorithm to converge to a local minimum of the cost function $J(\rho)$).

$R_i$ determines the update direction and is therefore crucial for the performance of the algorithm. Typically we choose a Gauss-Newton approximation of the Hessian of $J$, so:

$$R_i = \frac{1}{N}\sum_{t=1}^{N}\left(\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right]\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right]^T + \lambda\left[\frac{\partial u_t}{\partial \rho}(\rho_i)\right]\left[\frac{\partial u_t}{\partial \rho}(\rho_i)\right]^T\right) \qquad (8)$$

We will see that all the signals (estimates) needed in this expression of $R_i$ will be available from the IFT algorithm.

As stated the problem is intractable since it involves expectations that are unknown.

However, such a problem can be solved by replacing the gradient $\frac{\partial J}{\partial \rho}(\rho)$ by an unbiased estimate.
In order to solve this problem, one needs to generate the following quantities:

1) the signals $\tilde{y}_t(\rho)$ and $u_t(\rho)$;
2) the gradients $\frac{\partial \tilde{y}_t}{\partial \rho}(\rho)$ and $\frac{\partial u_t}{\partial \rho}(\rho)$;
3) unbiased estimates of the products $\tilde{y}_t(\rho)\frac{\partial \tilde{y}_t}{\partial \rho}(\rho)$ and $u_t(\rho)\frac{\partial u_t}{\partial \rho}(\rho)$.

These quantities can be obtained by performing experiments on the closed loop system formed by the actual system in feedback with the controller $\{C_r(\rho_i), C_y(\rho_i)\}$.

We will see that (8) can be expressed us follow:

$$R_i = \frac{1}{N}\sum_{t=1}^{N}\left(est\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right]est\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right]^T + \lambda\, est\left[\frac{\partial u_t}{\partial \rho}(\rho_i)\right]est\left[\frac{\partial u_t}{\partial \rho}(\rho_i)\right]^T\right).$$

(here and in the sequel, $est\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right]$ and $est\left[\frac{\partial u_t}{\partial \rho}(\rho_i)\right]$ denote the estimates of $\frac{\partial y_t}{\partial \rho}(\rho_i)$ and $\frac{\partial u_t}{\partial \rho}(\rho_i)$)

**COMPUTATION OF THE GRADIENT**

Noting that $\frac{\partial \tilde{y}_t}{\partial \rho}(\rho) = \frac{\partial y_t}{\partial \rho}(\rho)$, we can write:

$$\frac{\partial \tilde{y}_t}{\partial \rho}(\rho) = \frac{\partial y_t}{\partial \rho}(\rho) = \frac{G_0}{1 + C_y(\rho)G_0}\frac{\partial C_r}{\partial \rho}(\rho)r - \frac{C_r(\rho)G_0^2}{\left(1 + C_y(\rho)G_0\right)^2}\frac{\partial C_y}{\partial \rho}(\rho)r -$$

$$- \frac{G_0}{\left(1 + C_y(\rho)G_0\right)^2}\frac{\partial C_y}{\partial \rho}(\rho)v$$

$$= \frac{1}{C_r(\rho)}\frac{\partial C_r}{\partial \rho}(\rho)T_0(\rho)r - \frac{1}{C_r(\rho)}\frac{\partial C_y}{\partial \rho}(\rho)\left(\left[T_0\right]^2(\rho)r + T_0(\rho)S_0(\rho)v\right).$$

From the expression above we see that the gradient depends also on the not computable quantities $T_0(\rho)$ and $S_0(\rho)$ since they depend on the unknown system.
Therefore, unless an accurate model of the system is assumed to be available, the signal $\frac{\partial y_t}{\partial \rho}(\rho)$ can only be obtained by running experiments on the actual closed loop system.

Notice that the last two terms in the expression above involve a double filtering of the signal *r* and *v* throught the closed loop system:

$$\left[T_0\right]^2 r + T_0 S_0 v = T_0 y.$$

So we can write:

$$\frac{\partial y_t}{\partial \rho}(\rho) = \frac{1}{C_r(\rho)}\left[\frac{\partial C_r}{\partial \rho}(\rho)T_0(\rho)r - \frac{\partial C_y}{\partial \rho}(\rho)T_0(\rho)y\right]$$

$$= \frac{1}{C_r(\rho)}\left[\left(\frac{\partial C_r}{\partial \rho}(\rho) - \frac{\partial C_y}{\partial \rho}(\rho)\right)T_0(\rho)r + \frac{\partial C_y}{\partial \rho}(\rho)T_0(\rho)(r - y)\right]$$

The term $T_0(\rho)(r\text{-}y)$ can be obtained by substracting the output signal from one experiment of the closed loop system from the reference and by using this error signal as reference signal in a new experiment.
In each iteration *i* of the controller tuning algorithm, we will use three experiments, each of duration *N*, with the fixed controller $C(\rho_i) \triangleq \{C_r(\rho_i), C_y(\rho_i)\}$ operating on the actual plant.

We will see that only the second experiment is a *special* experiment, the first and the third just consist of collecting data under normal operating conditions.

We denote the *N*-length reference signals by $r_i^j$, $j$ =1,2,3, and the corresponding output signals by $y^j(\rho_i)$, $j$ =1,2,3.

We have:

$$r_i^1 = r$$
$$r_i^2 = r - y^1(\rho_i)$$
$$r_i^3 = r$$

So the expressions for the output signal are:

$$y^1(\rho_i) = T_0(\rho_i)r + S_0(\rho_i)v_i^1$$
$$y^2(\rho_i) = T_0(\rho_i)(r - y^1(\rho_i)) + S_0(\rho_i)v_i^2$$
$$y^3(\rho_i) = T_0(\rho_i)r + S_0(\rho_i)v_i^3$$

Here $v_i^j$ denotes the disturbance acting on the system during experiment *j* at iteration *i* (these disturbance can be assumed to be mutually independent since they come from different experiments).

With these experiments

$$\tilde{y}_i = y_i^1 - y_d$$

is a perfect realization of $\tilde{y}_i$ (where the subscript *i* denotes the iteration number), while

$$est\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right] \hat{=} \frac{1}{C_r(\rho)}\left[\left(\frac{\partial C_r}{\partial \rho}(\rho) - \frac{\partial C_y}{\partial \rho}(\rho)\right)y^3(\rho_i) + \frac{\partial C_y}{\partial \rho}(\rho_i)y^2(\rho_i)\right]$$

is a perturbed version (by the disturbance $v_i^2$ and $v_i^3$) of $\frac{\partial y_t}{\partial \rho}(\rho)$.

So now we have an estimate of the gradient that can be used in (6) and so in (7) for the updating control parameter law.

In an analogous way, we can obtain an estimate of the gradient $\frac{\partial u_t}{\partial \rho}(\rho)$.

From

$$u(\rho) = \frac{C_r(\rho)}{1 + C_r(\rho)G_0}r - \frac{C_y(\rho)}{1 + C_y(\rho)G_o}v = S_0(\rho)\left[C_r(\rho)r - C_y(\rho)v\right]$$

we can see that the three experiments described above generate the corresponding control signals:

$$u^1(\rho_i) = S_0(\rho_i)\left[C_r(\rho_i)r - C_y(\rho_i)v_i^1\right]$$
$$u^2(\rho_i) = S_0(\rho_i)\left[C_r(\rho_i)(r - y^1(\rho_i)) - C_y(\rho_i)v_i^2\right]$$
$$u^3(\rho_i) = S_0(\rho_i)\left[C_r(\rho_i)r - C_y(\rho_i)v_i^3\right]$$

These signals can similarly be used to generate the estimates of the input related signals required for the estimation of the gradient (6).

Indeed $u^1(\rho_i)$ is a perfect realization of $u(\rho_i)$,

$$u(\rho_i) = u^1(\rho_i)$$

while

$$est\left[\frac{\partial u_t}{\partial \rho}(\rho_i)\right] \hat{=} \frac{1}{C_r(\rho)}\left[\left(\frac{\partial C_r}{\partial \rho}(\rho) - \frac{\partial C_y}{\partial \rho}(\rho)\right)u^3(\rho_i) + \frac{\partial C_y}{\partial \rho}(\rho_i)u^2(\rho_i)\right]$$

Resuming, with these two gradient estimates an experimentally based estimate of the gradient of *J* can be formed by taking:

$$est\left[\frac{\partial J}{\partial \rho}(\rho_i)\right] = \frac{1}{N}\sum_{t=1}^{N}\tilde{y}_t(\rho)est\left[\frac{\partial y_t}{\partial \rho}(\rho)\right] + \frac{\lambda}{N}\sum_{t=1}^{N}u_t(\rho)est\left[\frac{\partial u_t}{\partial \rho}(\rho)\right]$$

**ONE-DEGREE-OF-FREEDOM CONTROLLERS**

In the case where the simplified controller structure $C_r = C_y \hat{=} C$ is used, i.e.,

$$u_t = C(\rho)(r - y),$$

the algorithm simplifies because the third experiment becomes unnecessary.

Therefore, in the case of a one-degree-of-freedom controller, the first two experiments are run with the same reference signals as the case of a two-degree-of-freedom controller, i.e.,

$$r_i^1 = r$$
$$r_i^2 = r - y^1(\rho_i)$$

and the gradient estimates are obtained by:

$$est\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right] = \frac{1}{C(\rho)}\frac{\partial C}{\partial \rho}(\rho_i)y^2(\rho_i)$$

$$est\left[\frac{\partial u_t}{\partial \rho}(\rho_i)\right] = \frac{1}{C(\rho)}\frac{\partial C}{\partial \rho}(\rho_i)u^2(\rho_i)$$

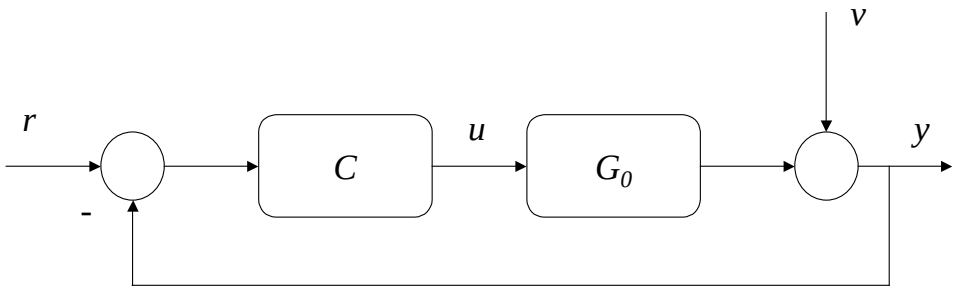



Fig. 2.2 A one-degree-of-freedom controller structure

### 2.1.1 Two-degree-of-freedom controllers – EXAMPLE 1

Consider the plant

$$G_0(s) = \frac{1}{s(s+1)}$$

and the controllers

$$C_r(q) = \frac{1.6535}{q + 0.8469}$$

$$C_y(q) = \frac{2.6854q - 1.0319}{q + 0.8469}$$

where the operator $q$ is the forward shift operator, that is $qf_t = f_{t+1}$.

We add a disturbance $v$ as in (1) given by a noise $n$ which is the output sequence of white (Gaussian) noise with zero mean and variance 0.01 filtered by the following frequency weighting function

$$W(s) = \frac{10}{s^2 + 10s}$$

The purpose is to improve the step response (tracking problem).

We take the following parametrization of the two controllers:

$$C_r(\rho) = \frac{\rho_1}{q + 0.8469}$$

$$C_y(\rho) = \frac{\rho_2 q + \rho_3}{q + 0.8469}$$

so that we have three parameters, $\rho_1, \rho_2, \rho_3$ to tune, starting from:

$$\rho_1 = 1.6535$$
$$\rho_2 = 2.6854$$
$$\rho_3 = -1.0319$$

We use the following quadratic criterion:

$$J(\rho) = \sum_{t=1}^{N} (y_t - y^d)^2$$

and the same fixed step size for all the parameters, $\gamma_i = 0.1$, and a Gauss-Newton approximation of the Hessian of $J$ for the matrix $R_i$.
Performing the IFT algorithm described above we get, after 15 iterations, the final parameters

$$\rho_1 = 33.1330$$
$$\rho_2 = 247.6248$$
$$\rho_3 = -214.6256$$

and the following results:

| Number of iterations | Value of the final criterion $J$ |
|---|---|
| 1 | 11.3131 |
| 2 | 10.7929 |
| 3 | 10.2780 |
| 4 | 9.7635 |
| 5 | 9.2482 |
| 6 | 8.7339 |
| 7 | 8.2239 |
| 8 | 7.7219 |
| 9 | 7.2314 |
| 10 | 6.7555 |
| 11 | 6.2971 |
| 12 | 5.8591 |
| 13 | 5.4442 |
| 14 | 5.0561 |
| 15 | 4.7000 |

(initial value of the criterion = 11.8443)

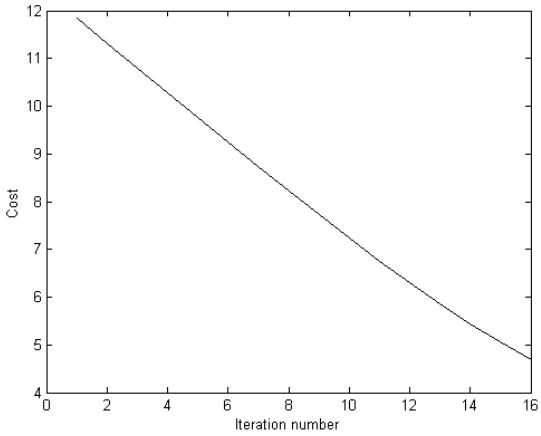



Fig. 2.3

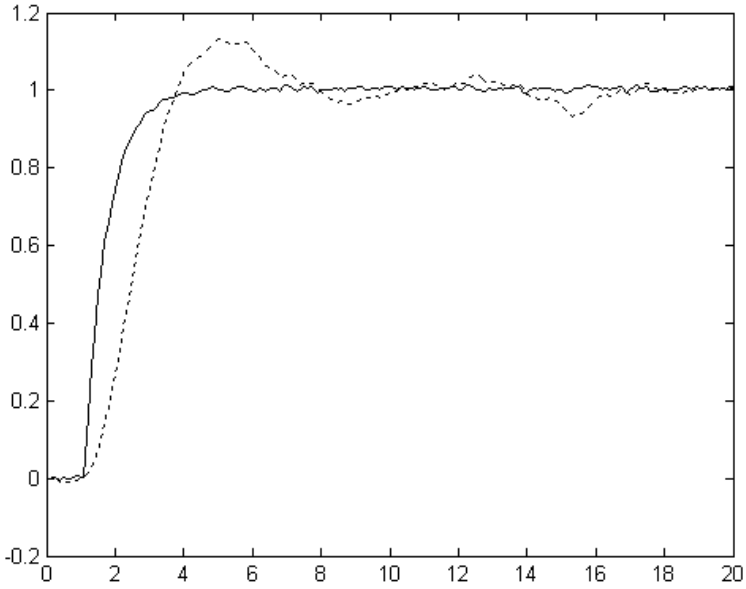


Fig. 2.4 Step Response (step at t =1, amplitude 1)
Dotted line: initial - Solid line: final

## 2.1.2 Two-degree-of-freedom controllers – EXAMPLE 2

We now apply the IFT scheme to the tuning of the controller for a flexible servo, a "two-mass-system" (two masses connected with a spring).

The following figure show the sketch of the process.

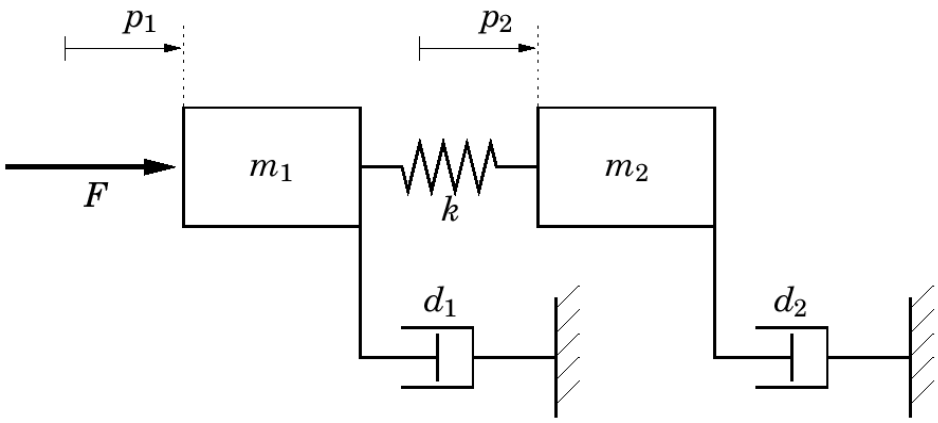



Fig. 2.5

We have the two masses $m_1$ and $m_2$. The spring between the masses has the spring constant $k$. The viscous damping coefficients are $d_1$ and $d_2$, respectively.
One of the masses is driven by a DC-motor. Here we neglect the internal dynamics of the motor. The force from the motor is proportional to the voltage u, that is:

$$F = k_m u$$

Force balance equations give the following dynamical model:

$$m_1 \frac{d^2 p_1}{dt^2} = -d_1 \frac{dp_1}{dt} - k(p_1 - p_2) + F(t)$$
$$m_2 \frac{d^2 p_2}{dt^2} = -d_2 \frac{dp_2}{dt} + k(p_1 - p_2)$$

Introducing the state vector $x = [p_1 \quad \dot{p}_1 \quad p_2 \quad \dot{p}_2]^T$ and $y = p_2$, the system can be written on state-space form,

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t)$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m_1} & -\frac{d_1}{m_1} & \frac{k}{m_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{m_2} & 0 & -\frac{k}{m_2} & -\frac{d_2}{m_2} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{k_m}{m_1} \\ 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} k_{y_1} & 0 & 0 & 0 \\ 0 & 0 & k_{y_2} & 0 \end{bmatrix}$$

We use the following constants and coefficients:

$m_1$ = 2.29 kg
$m_2$ = 2.044 kg
$d_1$ = 3.12 N/m/s
$d_2$ = 3.73 N/m/s
$k$ = 400 N/m
$k_m$ = 2.96 N/V
$k_y$ = 280 V/m

These values correspond to the real process which will be considered in next section.

The purpose is to control the position $p_2$.

Assuming that we could measure all the states, using a state feedback control strategy, we could place the poles for the closed loop system freely (controllability matrix has full rank) by using the control law:

$$u(t) = -Lx(t) + l_r y_r(t)$$

by properly choosing the parameters $L = [l_1\ l_2\ l_3\ l_4]$.

Here $y_r$ is the reference value and $l_r$ is a scalar gain affecting the overall gain and it is chosen to get the correct stationary gain (=1).

We choose the following desired poles for the closed loop system (c.l.s.):

$b_{1,2} = -1.4 \pm j15.7$
$b_{3,4} = -7.6 \pm j4.9$

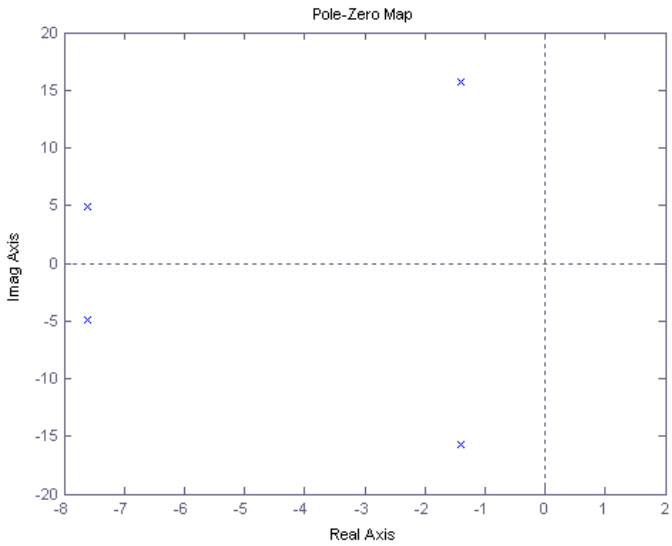



Fig. 2.6

Note the poorly damped first pole-pair, which we have chosen on purpose.

They give an oscillatory response for the position of the second mass and our aim will be to improve the corresponding controller to get a better behaviour.

Assume now that only the position $p_2$ is measurable. Thus we are faced with a control problem where we want to feedback signals which we can not measure.
We need to estimate the process state vector using an observer (Kalman filter) described by:

$$\frac{d\hat{x}(t)}{dt} = A\hat{x}(t) + Bu(t) + K(y_2(t) - C_2\hat{x}(t))$$

where $\hat{x}(t)$ denotes the states of the observer. $K$ can be chosen so that the observer states are approaching the real states with an arbitrarily fast rate of convergence (usually such that the observer dynamics will be 1.5 to 2 times faster than the closed loop system).
We get:

$$L = \begin{bmatrix} -20.9693 & 11.4598 & 101.2845 & 2.2562 \end{bmatrix}$$

$$K = \begin{bmatrix} 0.4428 \\ 1.9066 \\ 0.1172 \\ 3.6203 \end{bmatrix}$$

We will use the observer states in the feedback law instead of the real states which we can not measure.

The Simulink scheme of the controlled system is shown below.

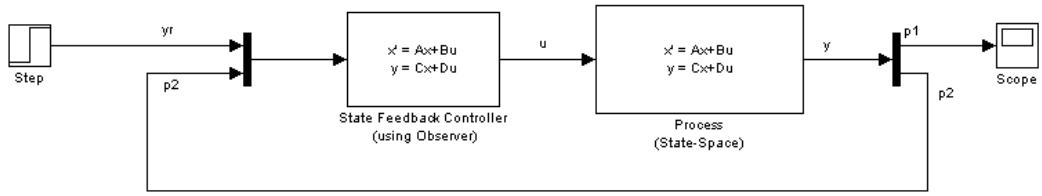



Fig. 2.7

By decomposing the state-space realization we can easily find the transfer functions from $y_r$ to $u$ and from $y_2$ to $u$, so that we could have a two-degree-of-freedom controller structure as below.

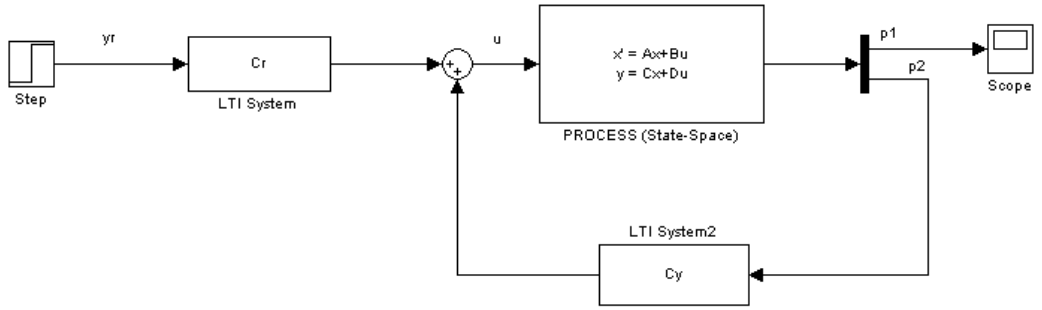



Fig. 2.8

The initial feedforward controller is:

$$C_r(\rho) = \frac{0.2868\,s^4 + 10.3262\,s^3 + 427.7\,s^2 + 9191\,s + 93238}{s^4 + 50.81\,s^3 + 1977\,s^2 + 50480\,s + 694410}$$

The initial feedback controller is:

$$C_y(\rho) = \frac{33\,s^3 - 122.1\,s^2 + 21835\,s + 93238}{s^4 + 50.81\,s^3 + 1977\,s^2 + 50480\,s + 694410}$$

These initial controllers give the following response (position second mass):

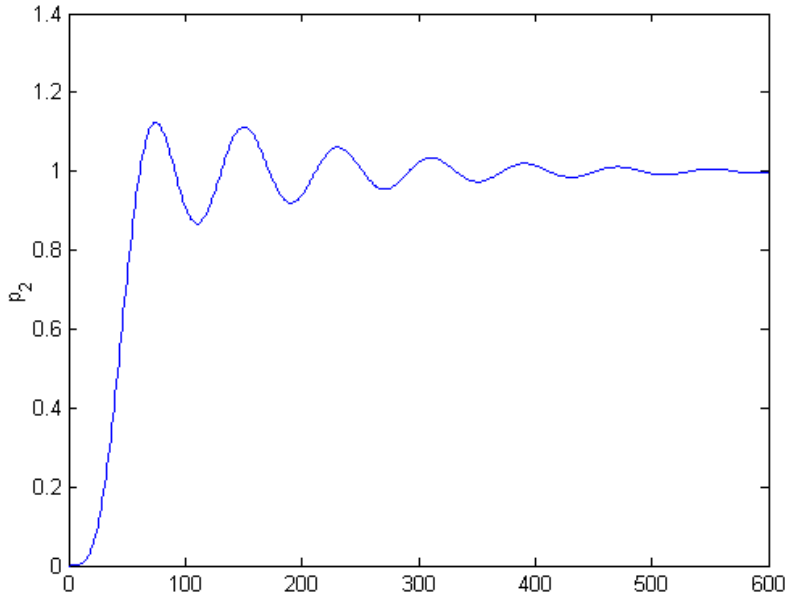



Fig. 2.9 Initial step response

The parametrization chosen to be used for the IFT algorithm is:

$$C_r(\rho) = \frac{\rho_1\,s^4 + \rho_2\,s^3 + \rho_3\,s^2 + \rho_4\,s + \rho_5}{q^4 + \rho_{10}\,s^3 + \rho_{11}q^2 + \rho_{12}\,s + \rho_{13}}$$

$$C_y(\rho) = \frac{\rho_6\,s^3 + \rho_7\,s^2 + \rho_8\,s + \rho_9}{q^4 + \rho_{10}\,s^3 + \rho_{11}q^2 + \rho_{12}\,s + \rho_{13}}$$

Notice the choise to keep the same denominator in the two controllers $C_r$ and $C_y$. This to keep the structure due by the observer and the state feedback control design.

We can now apply the IFT algorithm to update the control parameters of the feedback and the feedforward controller. We will use the following cost function:

$$J(\rho) = \frac{1}{2N}\sum_{t=1}^{N}(y_t - y^d)^2$$

As written above we need three experiments and we will implement them using Simulink. Actually in this example we do not use the disturbance *v* so we do not need the third experiment.

The first and the second experiments have the following Simulink schemes:

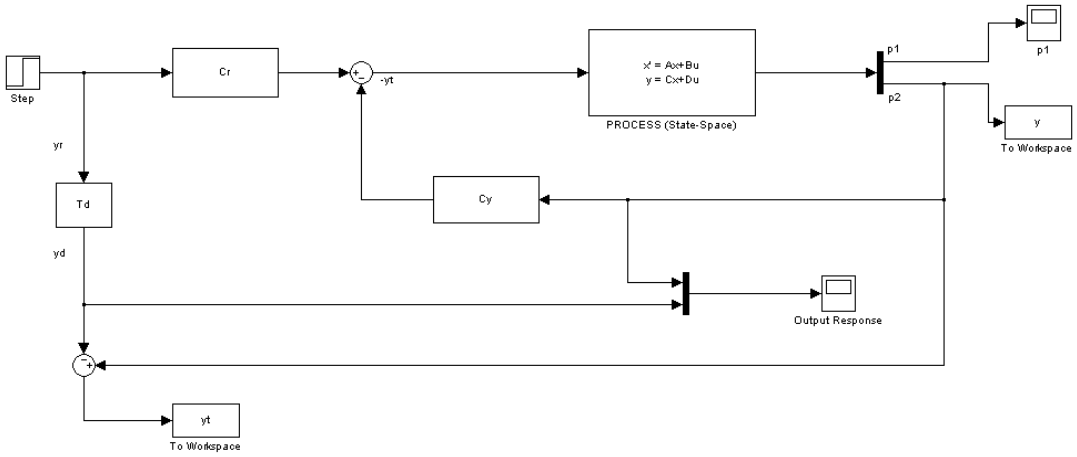



Fig. 2.10 First IFT experiment

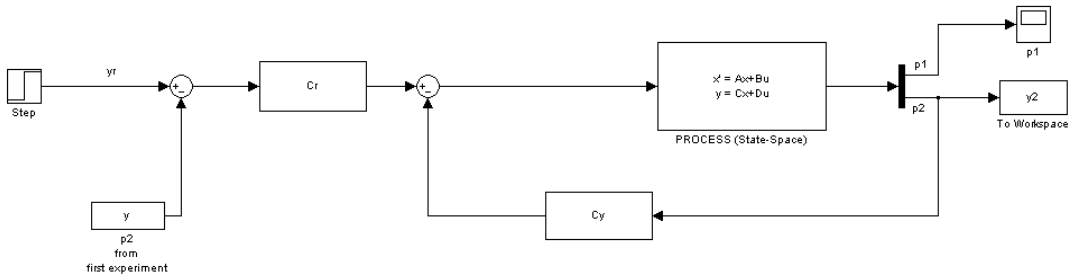



Fig. 2.11 Second IFT experiment: $r^2=r-y^2$

The desired output $y^d$ is chosen as:

$$y^d = T_d r = \left(\frac{a/0.25}{s + a/0.25}\right)^4 r$$

where $a$=3.5 and $r$ as a unit step.

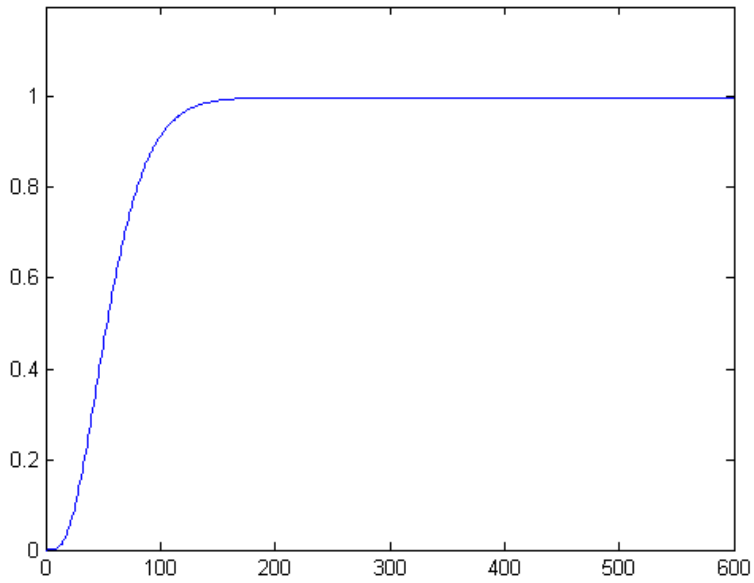



Fig. 2.12 Desired output $y^d$

Starting from an initial criterion

$$J_0 = 0.00497$$

and using a small step size of $\gamma_i = 0.005, \forall i,$ so that we can speak in terms of *fine tuning*, and a Gauss-Newton approximation of the Hessian of $J$ for the matrix $R_i$, after $n$ = 600 iterations, we get the following final parameters:

$\rho_1 = 0.527$
$\rho_2 = 10.8913$
$\rho_3 = 521.356$
$\rho_4 = 3.6184 \cdot 10^3$
$\rho_5 = 9.0616 \cdot 10^4$
$\rho_6 = 33.5995$
$\rho_7 = 147.4887$
$\rho_8 = 2.347 \cdot 10^4$
$\rho_9 = 9.0616 \cdot 10^4$
$\rho_{10} = 72.816$
$\rho_{11} = 2.6459 \cdot 10^3$
$\rho_{12} = 5.8381 \cdot 10^4$
$\rho_{13} = 6.8484 \cdot 10^5$

and the final criterion value

$$J_{n=600} = 1.1455 \cdot 10^{-5}$$

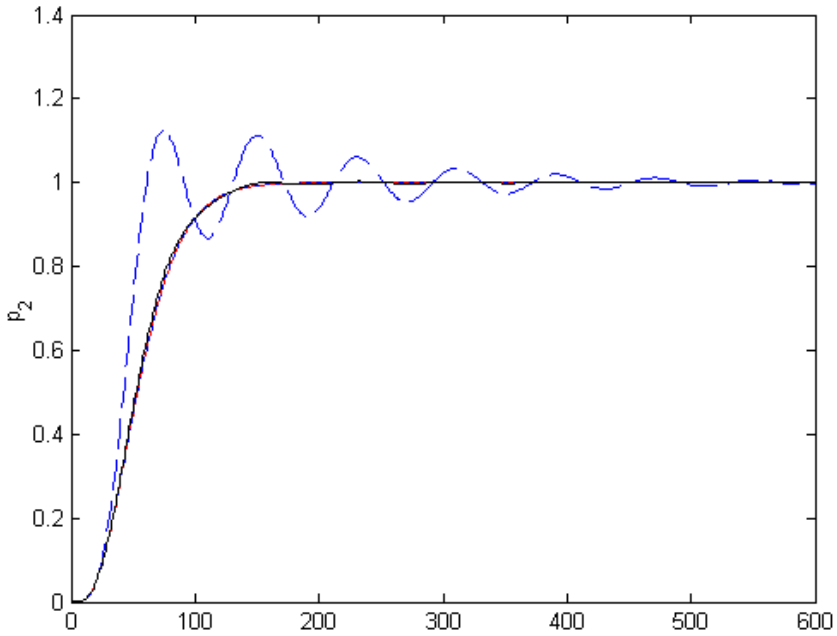
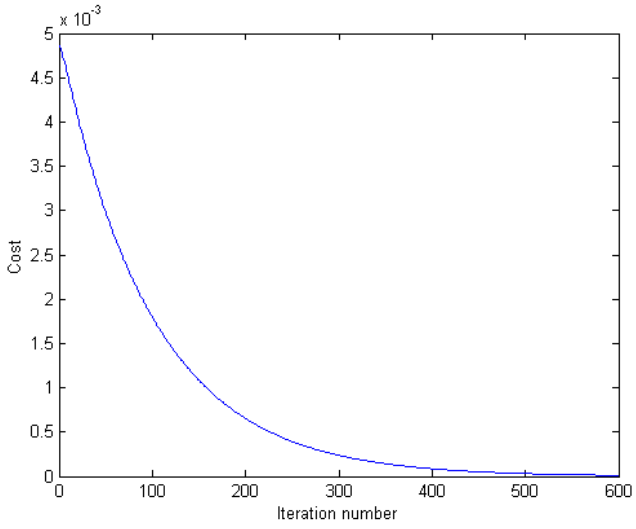



Fig. 2-13 Step Response (step at t =1, amplitude 1)
Solid line: final (after 600 iterations)
Dashed line: initial – Dotted: desired




| Iteration number | Final Cost |
|---|---|
| 100 | 0.001799028 |
| 200 | 0.000653825 |
| 300 | 0.000237864 |
| 400 | 0.000086545 |
| 500 | 0.000031488 |
| 600 | 0.000011455 |

Initial cost: 0.004971208

Fig. 2.14 Final value of the cost function *J* versus the IFT iteration number

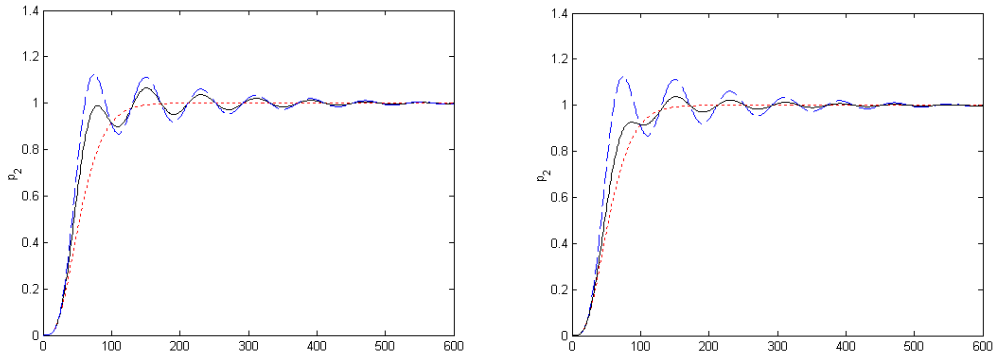



Fig. 2.15 Left: after 100 iterations – Right: after 200 iterations

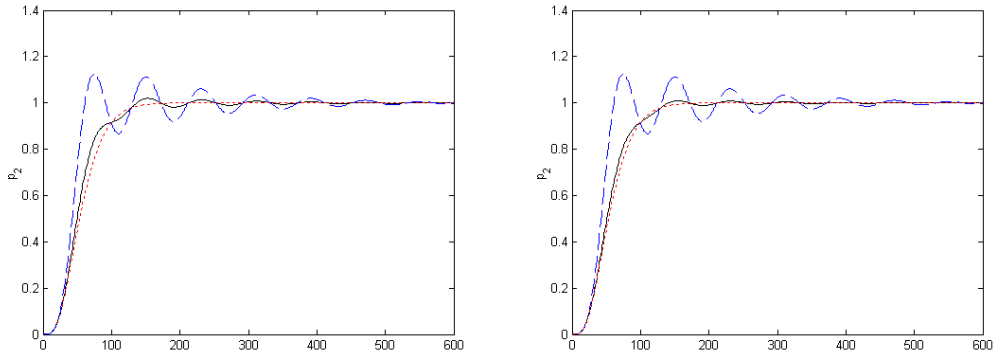



Fig. 2.16 Left: after 300 iterations – Right: after 400 iterations

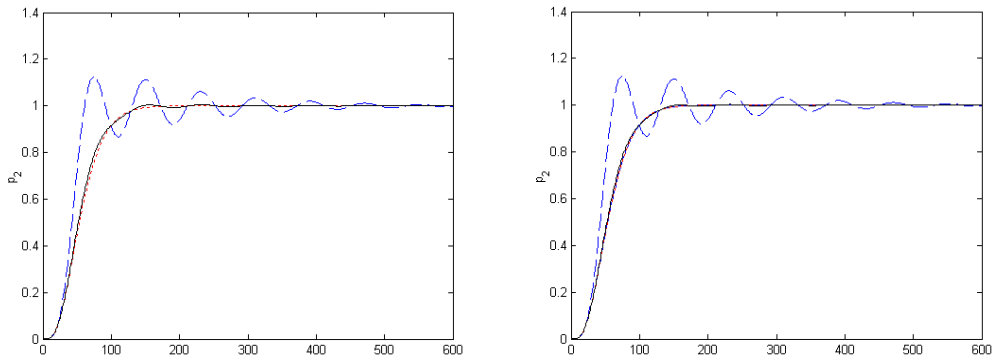



Fig. 2.17 Left: after 500 iterations – Right: after 600 iterations

The parameters updating is shown in the following figures.

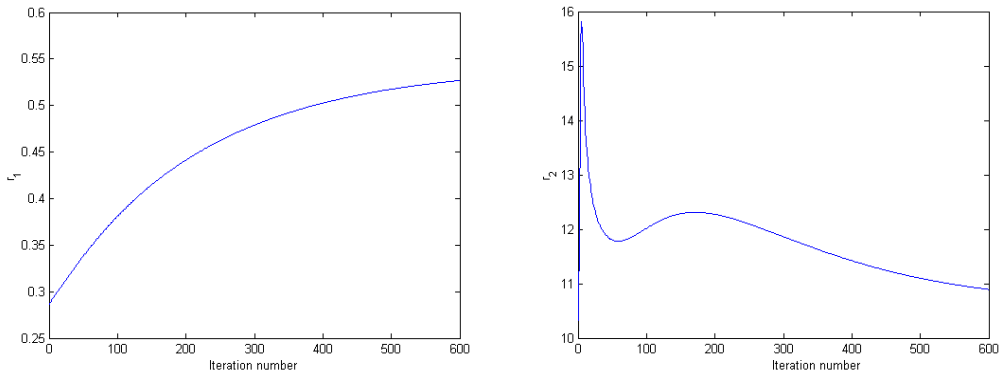



Fig. 2.18

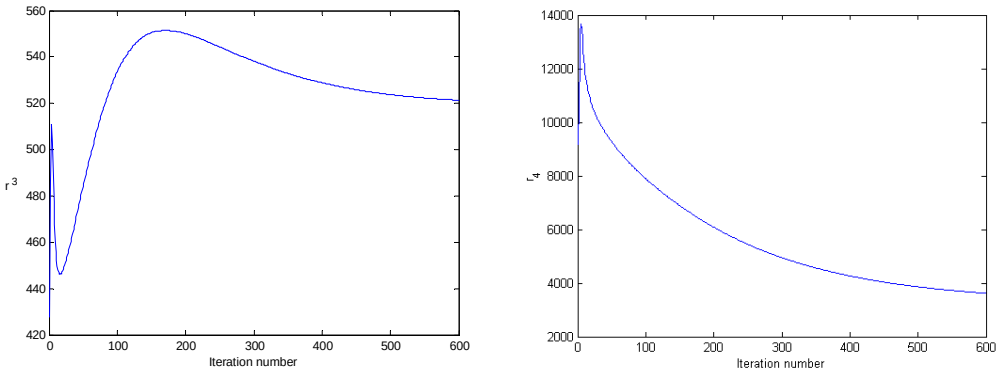



Fig. 2.19

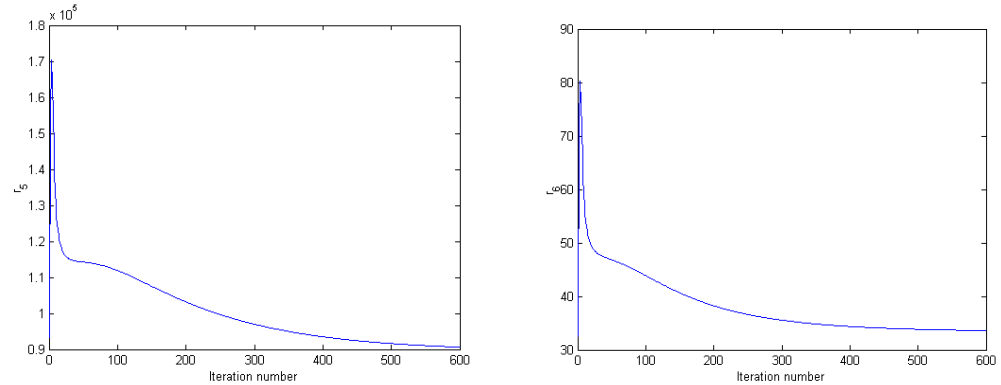



Fig. 2.20

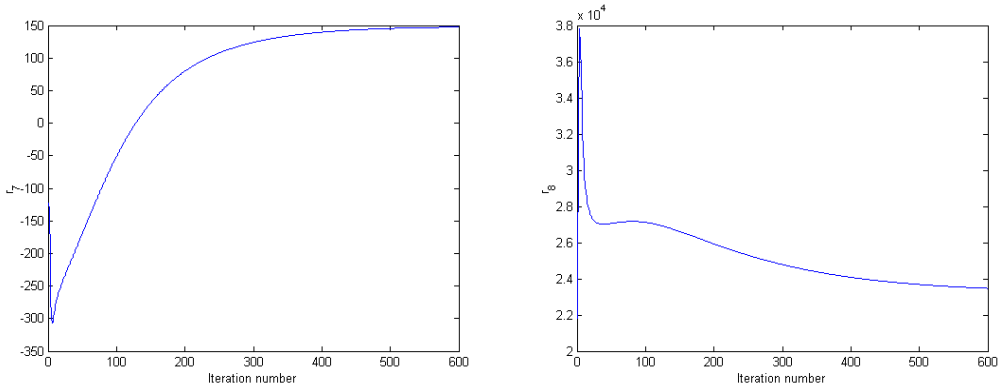



Fig. 2.21

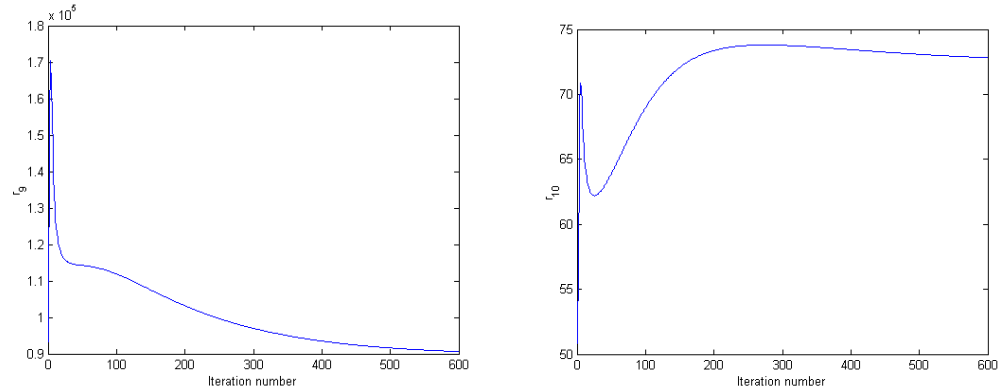



Fig. 2.22

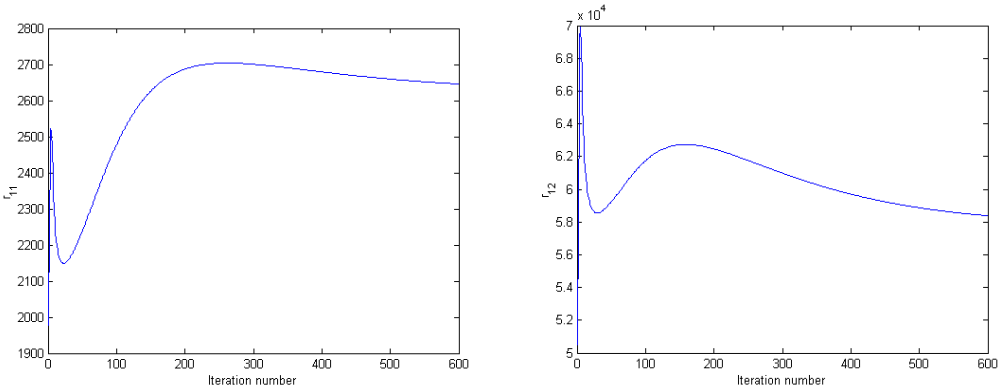
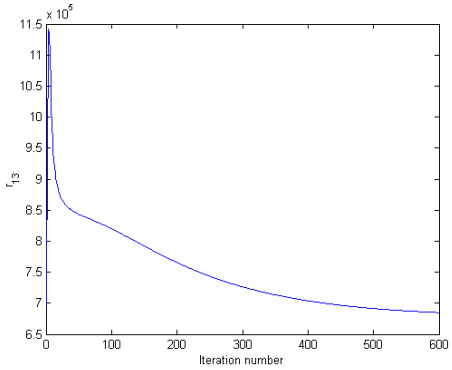



Fig. 2.23


x 10^5
11.5
11
10.5
10
9.5
9
8.5
8
7.5
7
6.5
f13
0
100
200
300
400
500
600
Iteration number
Fig. 2.24

## THE REAL PROCESS AND THE PROBLEM OF FRICTION

An implementation on the real process has been tried but a relevant problem has been encountered: friction.
In this case, a nonlinear contribution (friction) does not allow the IFT method to work properly and in our specific case, the friction was too important and the calculations of $y-y^d$ during the iterations of the algorithm were distorted.

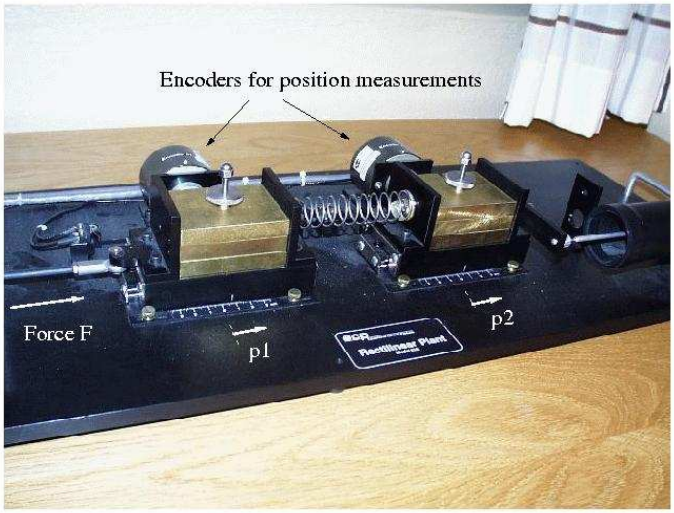



Fig. 2.25

In the following picture, we show the behaviour of the system and we can understand how in this case the implementation of IFT was almost impossible because we could neither check the result of the final controller nor keep stability properties.

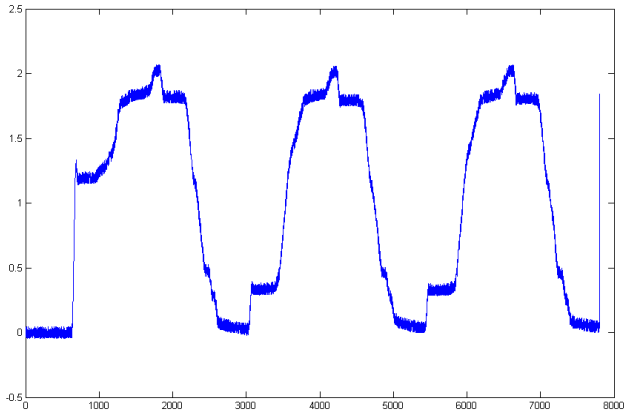



Fig. 2.26 Position response to a square wave reference signal

A problem with friction and a poorly tuned initial controller is that the process will stick at positions which are far from the reference value and which may change a lot from one iteration to another.
The experiments on the real process, even we are in presence of a limit case of huge friction, showed the limit of IFT that usually is based on the measurements of the error between the output of the system and the desired response.
However we will show in the next chapter how it is still possible to apply the IFT method to the case of (deterministic) nonlinear systems.

## 2.2 Convergence of IFT

Consider a discrete time linear time-invariant (LTI) model

$$y_t = G_0 u_t + v_t$$

($v_t$ disturbance)

and let be the system be controlled by the controller

$$u_t = C(\rho)(r - y_t)$$

with $\rho \in \Re^{n_\rho}$.

In this section we state exact conditions for which the controller parameters updated with the IFT algorithm converge to the set of stationary points of the criterion

$$J(\rho) = \frac{1}{2N} E\left[\sum_{t=1}^{N} (y_t(\rho) - y^d)^2\right].$$

Let $D$ be a convex compact subset of $\Re^{n_\rho}$.

We introduce the following conditions on the noise, the controller, the closed loop system and the step sizes of the algorithm, respectively.

V1) In any experiment, the signal sequence $v_t$ ,$t$=1, …, $N$ consists of zero mean random variables which are bounded: $|v_t| \le C$ for all t.
The constant $C$ and the second order statistics of $v_t$ are the same for all experiments, while sequences from different experiments are mutually independent.

C1) There exists a neighbourhood $\Theta$ to $D$ such that $C(\rho)$ is two times continuously differentiable w.r.t. $\rho$ in $\Theta$.

C2) All elements of the transfer functions

$$C(\rho)$$

$$\frac{\partial C(\rho)}{\partial \rho}$$

$$\frac{\partial^2 C(\rho)}{\partial \rho^2}$$

have their poles and zeros uniformly bounded away from the unit circle on $D$.

S1) The linear time-invariant closed loop system is stable and has all its poles uniformly bounded away from the unit circle on *D*.

A1) The elements of the sequence $\{\gamma_i\}$ satisfies $\gamma_i \geq 0$ and $\sum_{i=1}^{\infty} \gamma_i = \infty$.

A2) The elements of the sequence $\{\gamma_i\}$ satisfies $\sum_{i=1}^{\infty} \gamma_i^2 < \infty$.

***Theorem*** (Hjalmarsson, 1998)

Consider the IFT algorithm ( given by $\rho_{i+1} = \rho_i - \gamma_i R_i^{-1} \frac{\partial J}{\partial \rho}(\rho_i)$ ).

Assume that V1), C1), C2), S1), A1) and A2) hold.
Assume that $R_i$ is a symmetric matrix which is generated by the experiments at iteration *i* and satisfies

$$\frac{1}{\delta} I \geq R_i \geq \delta I$$

for some $\delta > 0$.

Then

$$\lim_{i \to \infty} \rho_i = D_c \triangleq \{\rho : J'(\rho) = 0\}$$

on a set $A = \{\rho_i \in D \ \forall i\}$.

The basic requirement for convergence is that the signals remain bounded throughout the iterations (since the result only applies to the set *A* introduced in the theorem).

The power of the theorem is that apart from the assumption of linearity and time-invariance there are no other assumptions on the properties of the system. The same holds for the controller: the complexity of the controller is arbitrary and the result thus applies to simple PID controllers as well as to more complex ones.

It is also important to notice that even though the disturbances have to have the same second order statistics from experiment to experiment, it is not necessary that the disturbances are stationary during one experiment.

## 2.3 The step size: a critical choice

In this section we discuss and show with some expamples, how critical is the choice of the step size γ in the parameter update law

$$\rho_{i+1} = \rho_i - \gamma_i R_i^{-1} \frac{\partial J}{\partial \rho}(\rho_i) .$$

Let's take the following robot model:

$$G_0(s) = \frac{100}{s(s+1)(s^2+15s+100)}$$

controlled by a PD controller:

$$G_C(s) = k_p \left[ \frac{1+sT_d}{1+s\frac{T_d}{N}} \right]$$

with $N$=10.

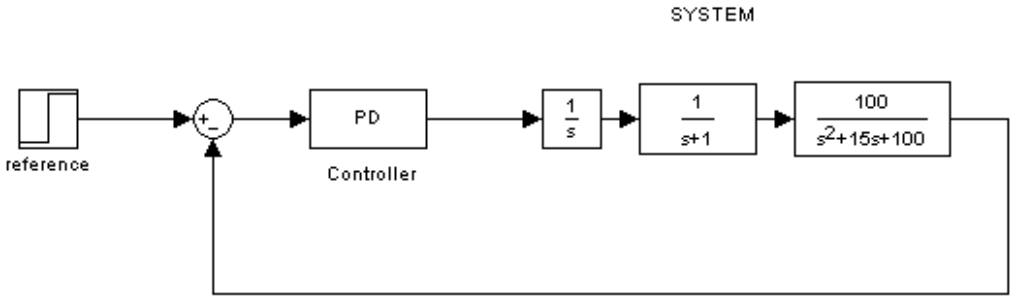



Fig. 2.27

To implement IFT on this system we use the criterion

$$J(\rho) = \frac{1}{2N} \sum_{t=1}^{N} (y_t - y^d)^2$$

and a Gauss-Newton approximation of the Hessian for the matrix $R_i$.

We use as desired output

$$y^d = T_d r = \frac{a}{s-a} r(t)$$

with $r(t)$ as the unit step and $a$=4.

Starting with the initial parameters $k_p$=2 and $T_d$=2 we get the an initial cost $J_0$ = 0.0077 and the following response:

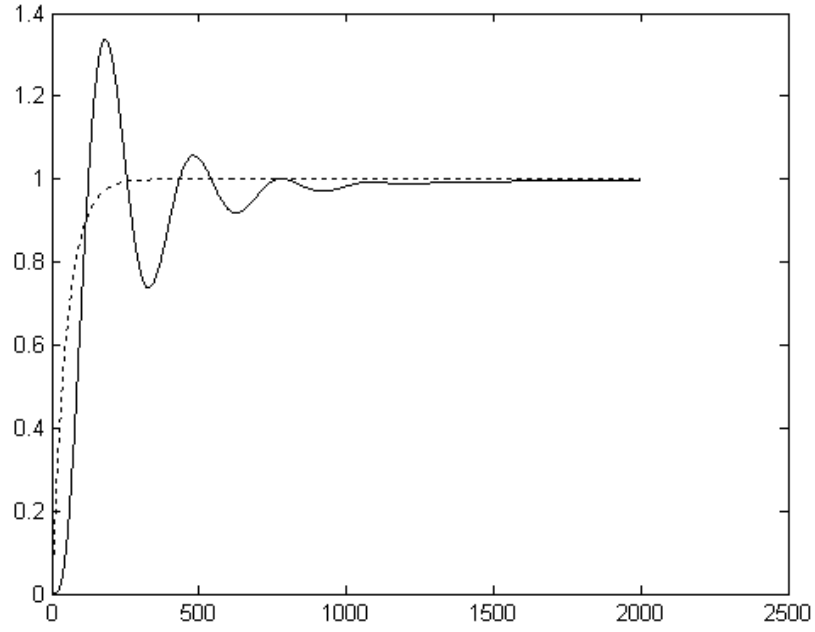



Fig. 2.28 Response with the initial PD parameters(full) and the desired output (dotted)

Performing 50 iterations of IFT using a step size $\gamma$=0.1 we get a final cost $J_{50}$ = 0.0043 and the following response:

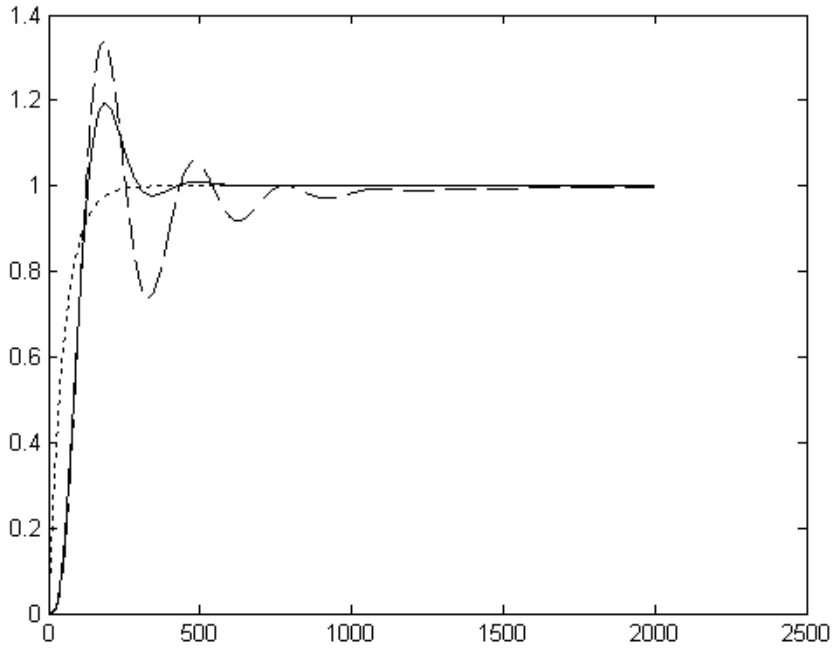



Fig. 2.29 Final response after 50 iterations of IFT using a step size of 0.1(full), initial response(dashed) and desired output (dotted)

We can look at the cost function values as function of the iterations number:

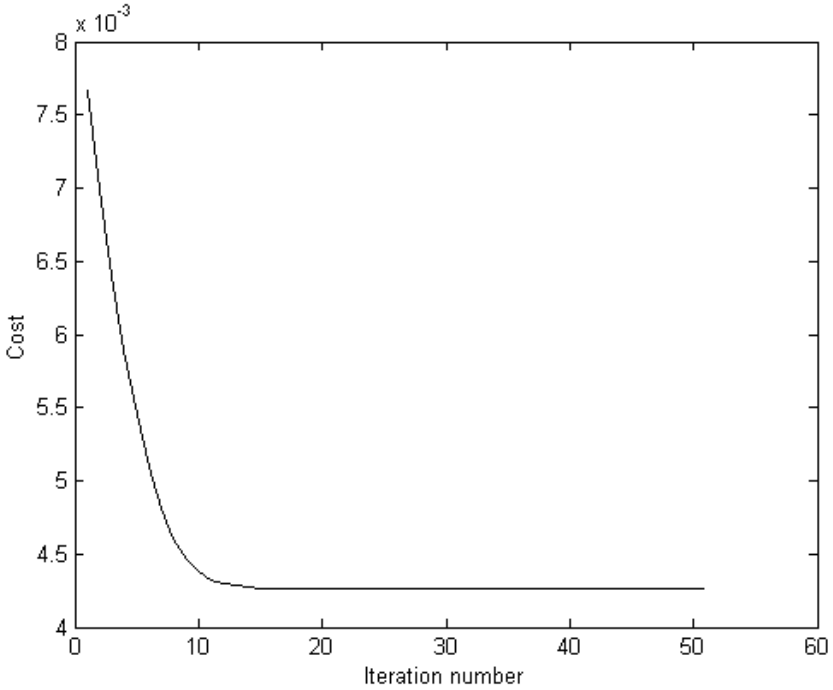



Fig. 2.30 (step size $\gamma$=0.1, final cost $J_{50}$ = 0.0043)

We perform now the same number of iterations, starting form the same initial parameters but using a bigger step size, $\gamma$=0.5.

The response is:

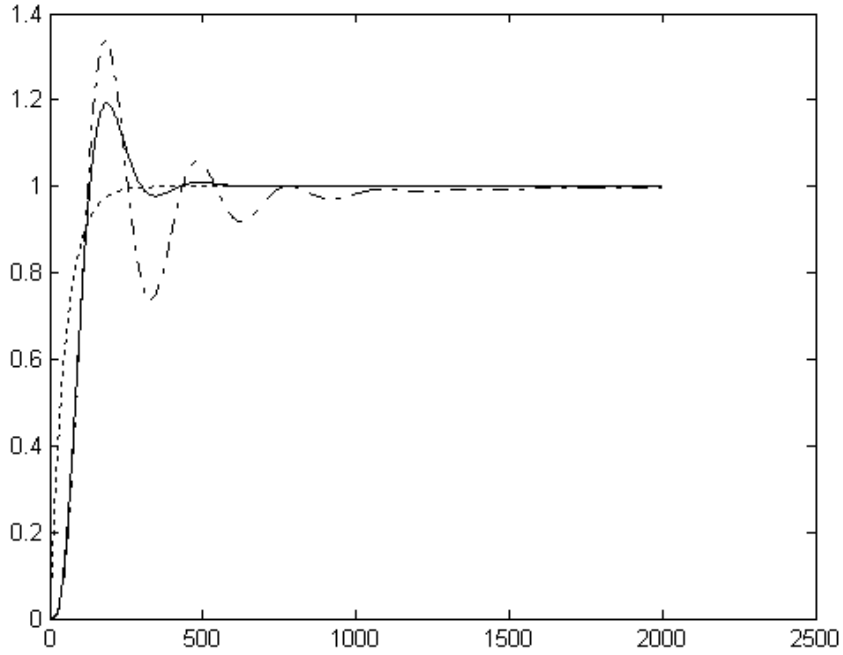



Fig. 2.31

We get the same final cost value $J_{50}$ = 0.0043 but if we look at the graph of $J$ as function of the iteration numbers we can observe an interesting behaviour:

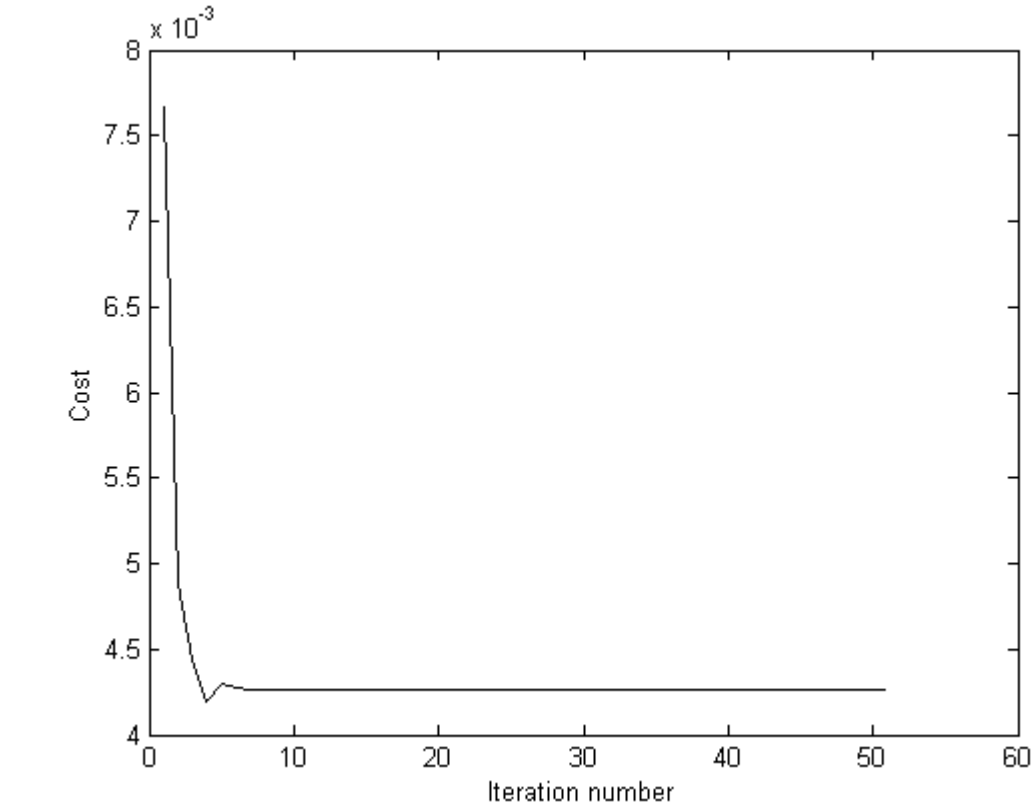



Fig. 2.32 (step size γ=0.5, final cost $J_{50}$ = 0.0043)

We can see that after 3 iteration we get a lower, even if not considerably, cost function value ($J_{50}$ = 0.0042 ).

So it's not useful to continue in the iterations and it's enough to stop the IFT algorithm just after 3 iterations.

This is an important aspect of the iterative tuning methods since it can happen that we find a local minimum just after few iteration and if we don't look at the cost value during the iterations it can happen that we can get lower performance from the parameters updating.
This is due in main part to the "irregular" surface of the cost function $J$ and so it is critical to choose a right step size or at least a good stop rule for the algorithm.
Another important issue is the approximation of the gradients and the form of the matrix $R$. In fact, in case of very irregular surface of $J$, we could need to use a very small step size if the approximation is not accurated. This holds to perform a high number or iterations for the IFT in order to get improvements of the controller.

Interesting is also the behaviour of the IFT implementation after 50 iteration and with a step size equal to 0.9:

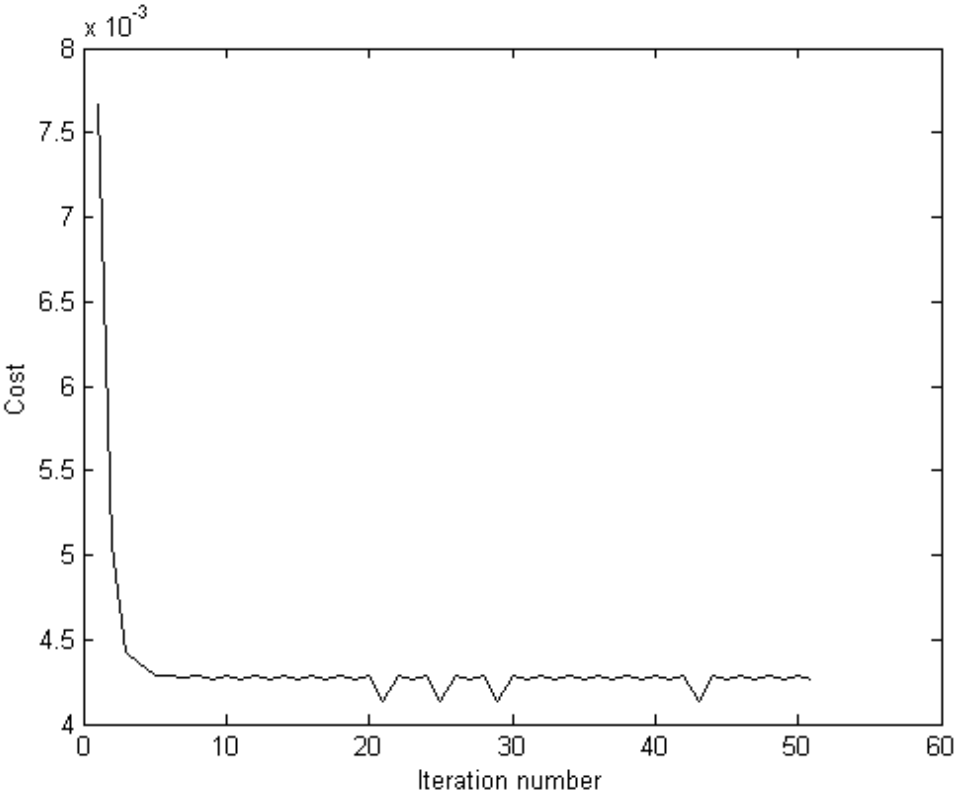



Fig. 2.33 (step size $\gamma$=0.9, final cost $J_{50}$ = 0.0043)

In this case we can see that we find more than one local minimum during the IFT implementation or at least four different iteration numbers in which we could stop instead of arriving at 50 iterations.

# 3. The Nonlinear Case

In this chapter we discuss an analysis of the properties of IFT when applied to nonlinear systems controlled by a under a linear controller.
We will assume that the system to be controlled is given by the following nonlinear state-space model:

$$\begin{aligned} x(t+1) &= f(x(t), u(t), w(t)) \\ y(t) &= h(x(t), v(t)) \end{aligned} \qquad (*)$$

where $f = f(x,u,w)$ and $h = h(x,v)$ are smooth functions, where $x(t)$ represent the state vector at time $t$, where $u(t)$ and $y(t)$ are the scalar inputs and outputs and where $w(t)$ and $v(t)$ are external disturbances.

We will also assume that the system is controlled by the following linear time-invariant controller C($q$,ρ):

$$u(t) = C(q,\rho)(r(t) - y(t))$$

where $r(t)$ is the external reference signal and ρ is the parameters vector.

Proceding in an analogous way of the previous sections, we get, differentiating the system equations wrt $\rho_i$:

$$\begin{aligned} x'(t+1) &= f_x x'(t) + f_u u'(t) \\ y'(t) &= h_x x'(t) \\ u'(t) &= C'(r(t) - y(t)) - Cy'(t) \\ &= C\left(\frac{C'}{C}(r(t) - y(t)) - y'(t)\right) \end{aligned}$$

where

$$x'(t) = \frac{d}{d\rho_i} x(t)$$

$$y'(t) = \frac{d}{d\rho_i} y(t)$$

$$u'(t) = \frac{d}{d\rho_i} u(t)$$

$$f = f(x(t), u(t), w(t))$$

$$f_x = \frac{d}{dx} f(x(t), u(t), w(t))$$

$$f_u = \frac{d}{du} f(x(t), u(t), w(t))$$

As the linear case, if follows that the gradients

$$\frac{d}{d\rho_i} y(t)$$

and

$$\frac{d}{d\rho_i} u(t)$$

can be obtained by first performing a simulation using the system (*) with $r(t)$ as reference signal and collecting the signals $y_1(t)$, $u_1(t)$, $x_1$(t) and $w_1(t)$, $t$ =1,…,N (where the subscript denoted that the signals stem from the first simulation using system (*) ).
With these signals at hand, in a second simulation use the *linear time-varying* feedback system

$$\begin{aligned} x_2(t+1) &= A(t)x_2(t) + B(t)u_2(t) \\ y_2(t) &= H(t)x_2(t) \\ u_2(t) &= C(s(t) - y_2(t)) \end{aligned}$$

with reference signal

$$s(t) = \frac{\frac{d}{d\rho_i} C(q,\rho)}{C(q,\rho)} (r(t) - y_1(t))$$

and with the special choice of time-varying matrices

$$\begin{aligned} A(t) &= \frac{\partial}{\partial x} f(x_1(t), u_1(t), w_1(t)) \\ B(t) &= \frac{\partial}{\partial u} f(x_1(t), u_1(t), w_1(t)) \\ H(t) &= \frac{\partial}{\partial x} h(x_1(t), v_1(t)) \end{aligned}$$

which are function of the signals $x_1(t)$, $u_1(t)$, $w_1(t)$, $v_1(t)$ in the first experiment.

It then holds that

$$u_2 = \frac{d}{d\rho_i} u(\rho)$$

$$y_2 = \frac{d}{d\rho_i} y(\rho)$$

and

$$x_2 = \frac{d}{d\rho_i} x(\rho).$$

By repeating the second simulation with a new reference signal

$$s(t) = \frac{\frac{d}{d\rho_j} C(q,\rho)}{C(q,\rho)} (r(t) - y_1(t))$$

the (*true*) gradients with respect to $\rho_j$ can be obtained.

To obtain the gradient with respect to the complete parameter vector $\rho \in \Re^n$, the second simulation has to be performed *n* times.
A drawback with this method is that the number of experiments is proportional to the number of parameters that are to be tuned.

An alternative approach to avoid this is suggested by Sjöberg and Agarval in [Sjöberg *et al*, 1997] and by De Bruyne, Anderson and Gevers in [De Bruyne *et al*, 1996] where a linear-varying model is identified.

Another approach for the control of non linear systems is the development of IFT due to Hjalmarsson *et al* (1998), that we now describe.

For each iteration *i* in

$$\rho_{i+1} = \rho_i - \gamma_i R_i^{-1} \frac{\partial J}{\partial \rho}(\rho_i)$$

the IFT method uses two experiments, each of duration *N* say, with the fixed controller $C(\rho_i)$ operating on the actual plant. Notice that contrary to the method outlined above the number of experiments is fixed to two regardless of the dimension of the parameter vector ρ.

When *r-y* is used as reference in the second experiment in IFT, the system equations in the second experiment can be written:

$$\begin{aligned} x_2(t+1) &= f(x_2(t), u_2(t), w_2(t)) \\ y_2(t) &= h(x_2(t), v_2(t)) \\ u_2(t) &= C(r(t) - y_1(t) - y_2(t)) \end{aligned}$$

Approximating the first two equations by a first order Taylor expansion around $(x_1(t), u_1(t), w_1(t), y_1(t), v_1(t))$, the closed loop trajectories in the first experiment (with *r* as reference signal), gives

$$\begin{aligned} x_2(t+1) &\approx f_x x_2(t) + f_u u_2(t) + \Delta(t) \\ y_2(t) &\approx h_x x_2(t) + \delta(t) \\ u_2(t) &= C(r(t) - y_1(t) - y_2(t)) \end{aligned} \qquad (**)$$

where

$$\begin{aligned} \Delta(t) &= f_w w_2(t) + f - f_x x_1(t) - f_u u_1(t) - f_w w_1(t) \\ \delta(t) &= h_v v_2(t) + h - h_x x_1(t) - h_v v_1(t) \end{aligned}$$

Above the suppressed arguments are signals from the first experiment:

$$f = f(x_1(t), u_1(t), w_1(t)).$$

Notice that $\Delta(t)$ and $\delta(t)$ can be regarded as external signals since they are functions of variables in the first experiment and $w_2$ and $v_2$ only.
Disregarding these signals we see that the first order approximation is identical to the linear time-varying equations seen previously that generate the true gradient except for the fact that the reference signal $r(t)$-$y(t)$ in the expression of $u'(t)$ is filtered by $C'/C$ whereas in (**) is not. The reason of this last difference is that in the IFT algorithm this filtering is done after and not before the second experiment.

These similarities suggest that it should be possible to improve the performance for nonlinear systems using the standard IFT procedure under the following conditions:

- the first order Taylor approximation is reasonably accurate;
- the signals $\Delta(t)$ and $\delta(t)$ are small compared to $f_x x_2(t) + f_u u_2(t)$ and $h_x x_2(t)$, respectively;
- the error due to commuting the $C'/C$ operator and the closed loop system is small.

It may seem as if these stated conditions are quite restrictive. However, practice, as will be evidenced below, has shown that this does not seem to be the case for many systems. One reason for this is that it suffices to be able to compute a descent direction, the exact gradient is not necessary.
However, it might be necessary to reduce the step-size $\gamma$ when a perturbed gradient estimate is used. Furthermore, care has to be exercised if e.g. a Gauss-Newton update is used since the joint effect of the gradient perturbation and the modification of the search direction caused by $R_i$ is additive and one may end up in an ascent direction.
We conclude that for nonlinear systems, it might be wise to use a small step-size.

## A SIMULATION EXAMPLE

We will consider now the following noise-free nonlinear system which has $x(t)$ and $z(t)$ as states:

$$\begin{aligned} x(t+1) &= x(t) - 0.1x^3(t) - 0.1u(t) \\ z(t+1) &= z(t) - 0.2x^3(t) - 0.2z^3(t) \\ y(t) &= z(t) \end{aligned}$$

Remembering that the subscripts denote experiment number in the IFT procedure, for this system the right-hand side of the first equation of (**) becomes:

$$f_x x_2(t) + f_u u_2(t) \triangleq \begin{pmatrix} \xi(t) \\ \gamma(t) \end{pmatrix} = \begin{pmatrix} (1-0.3x_1^2(t))x_2(t) - 0.1u_2(t) \\ -0.6x_1^2(t)x_2(t) + (1-0.6z_1^2(t))z_2(t) \end{pmatrix}$$

and

$$\Delta(t) \triangleq \begin{pmatrix} \lambda(t) \\ \psi(t) \end{pmatrix} = \begin{pmatrix} 0.2x_1^3(t) \\ 0.4x_1^3(t) + 0.4z_1^3(t) \end{pmatrix}$$

We'll consider the system controlled with the PI controller $C_0 = \dfrac{0.02}{1-q^{-1}}$.

We can prove that $|\xi(t)|$ dominates $|\lambda(t)|$ and also $|\psi(t)| << |\gamma(t)|$.

This indicates that the perturbation $\Delta(t) = \begin{pmatrix} \lambda(t) \\ \psi(t) \end{pmatrix}$ is not very significant. So we can say that $x_2(t+1) = f_x x_2(t) + f_u u_2(t)$ is a reasonable approximation of the second experiment. This indicates also that, provided the commutation $C'/C$ and the closed loop system does not influence the signals too much, it should be possible to use the IFT on this system.

We choose a reference signal as a period of a square-wave with period time 250.

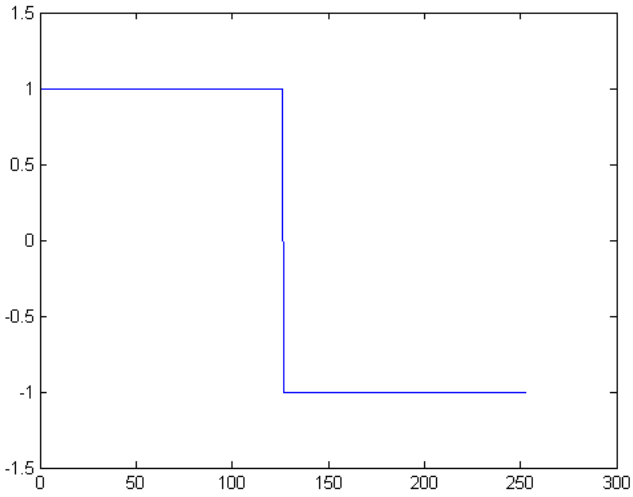



Fig. 3.1 Reference signal for the first IFT experiment

The desired output $y^d$ is taken to be the reference signal of the first IFT experiment and filtered throught the following low-pass filter

$$y^d = T_d r = \frac{0.05q^{-1}}{1-0.95q^{-1}} r(t)$$

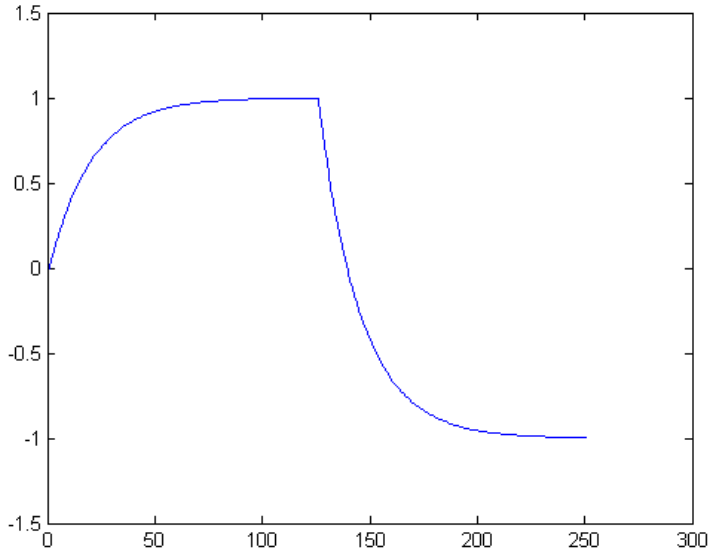



Fig. 3.2 Desired response $y^d$

The parameters $\rho_1$, $\rho_2$, $\rho_3$, $\rho_4$ are adjusted in the following discrete time generalization of a PID controller structure:

$$C_0 = \frac{\rho_1 + \rho_2 q^{-1} + \rho_3 q^{-2} \rho_4 q^{-3}}{1 - q^{-1}}$$

We use for the IFT algorithm:

- a cost function $J(\rho) = \sum_{t=1}^{N} (y_t - y^d)^2$
- a Gauss-Newton approximation of the Hessian for the matrix $R_i$
- a step size $\gamma_i = 0.5 \; \forall i$ .

The initial parameters are:

$\rho_1 = 0.02$
$\rho_2 = 0$
$\rho_3 = 0$
$\rho_4 = 0$

and the corresponding response can be seen in the following figure.

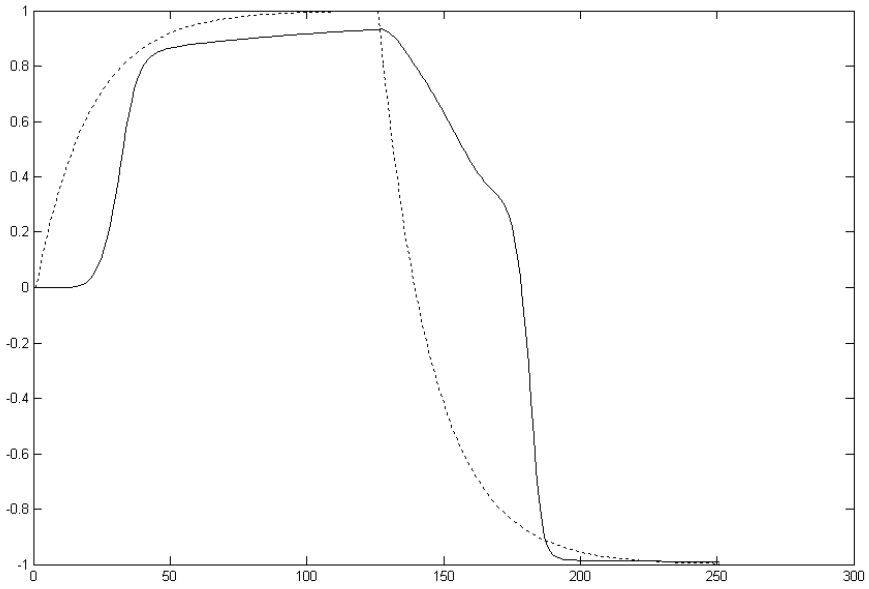



Fig. 3.3 Closed loop response with the initial PID parameters (full) and desired response $y^d$ (dotted)

As can be seen the system exhibits some quite nonlinear behaviours.

Applying the IFT algorithm, we obtain the following response after 50 iterations:

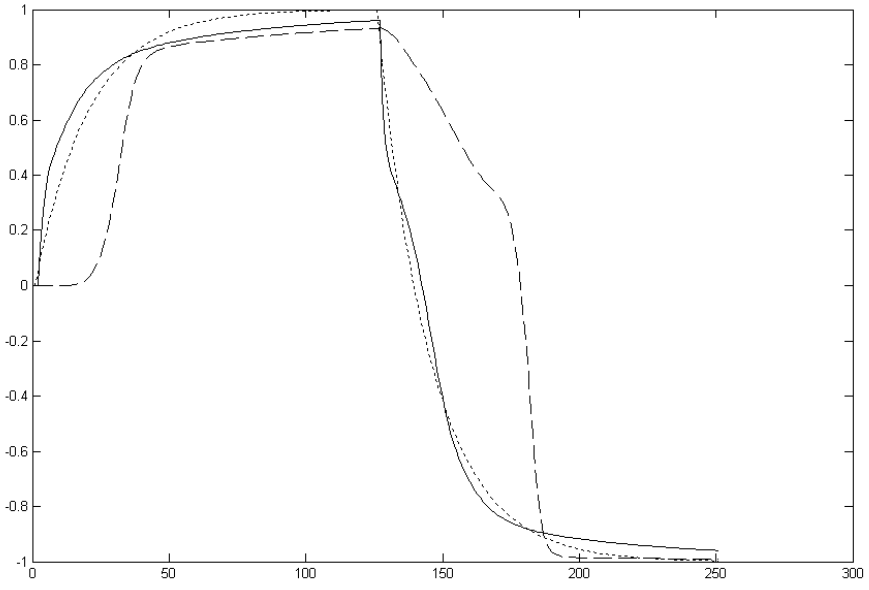



Fig. 3.4 Response(full) after 50 iterations, desired response (dotted) and response with the initial PID parameters (dashed)

The criterion decreases monotonically during the iterations as shown below in the plot and in the table (the subscript denotes after how many iteration the values stem from).

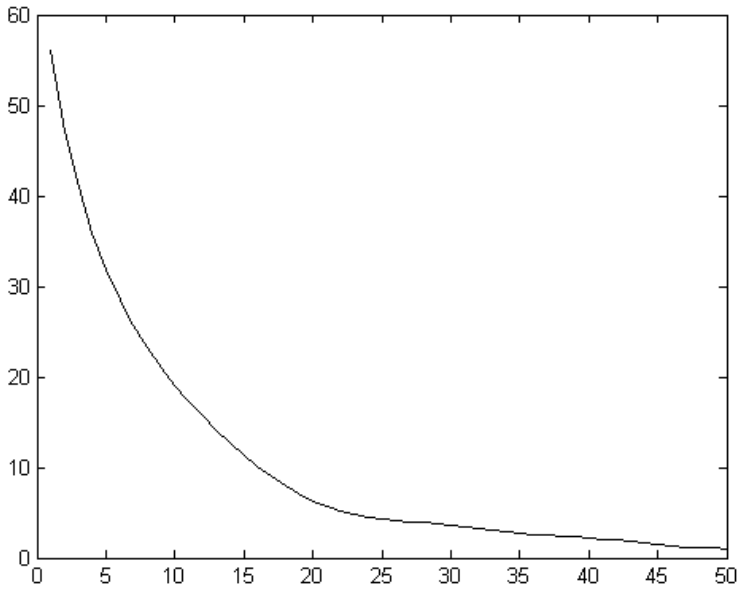



Fig. 3.5 Cost $J$ as function of the number of IFT iterations

| **Criterion values** |
|---|
| $J_{0(\text{initial})}$= 56.0030 |
| $J_{10}$= 19.0633 |
| $J_{20}$= 6.3112 |
| $J_{30}$= 3.5936 |
| $J_{40}$= 2.2127 |
| $J_{50(\text{final})}$= 1.0496 |

A comparison with the initial response shows that the IFT has managed to improve the performance considerably.
It should also be noted that the simple linear controller makes a surprisingly good job on this nonlinear system.

The corresponding final controller is given by:

$$C_{50} = \frac{8.7931 - 16.9071q^{-1} + 8.1999q^{-2} - 0.0447q^{-3}}{1 - q^{-1}} \ .$$

# 4. Modifications and Improvements to IFT

## 4.1 Modified criterion

One of the frequent practical use of controller design is to tune a controller of fixed structure (for example a PID controller) in such a way that the step response of the closed-loop system has a minimal settling time with a small overshoot.
The objective in such applications is to move the output of the closed-loop system quickly from one reference value to another; however, the particular shape of the transient response from the initial reference value to the final value is of no importance, provided that it does not have large overshoot. In addition, without knowledge of the actual system (which is a major reason for using IFT) it is not known in advance how fast a settling time can be achieved for this particular system with this particular controller structure.

By imposing the entire response of the closed-loop system through a specific choice of a desired response $y_d$, rather than just the endpoint of this transient response, the classical IFT criterion leads to controller parameters that realize a compromise between fitting the transient response and fitting the new reference value, even though the user does not care about the exact shape of the transient response. Instead, by imposing a mask on the transient response, the optimization will tune the controller parameters in such a way as to achieve the new desired reference value without focusing on a particular pre-imposed transient response that is perhaps not naturally achieved by the closed-loop system.

We can introduce a variant of the control performance criterion (4) in which the signals $\tilde{y}_t(\rho)$ and $u_t(\rho)$ are time weighted by weightings $w_y(t)$ and $w_u(t)$, respectly.

Thus the criterion

$$J(\rho) = \frac{1}{2N} E\left[ \sum_{t=1}^{N} (L_y \tilde{y}_t(\rho))^2 + \lambda \sum_{t=1}^{N} (L_u u_t(\rho))^2 \right]$$

is replaced by

$$J(\rho) = \frac{1}{2N} E\left[ \sum_{t=1}^{N} w_y(t)(L_y \tilde{y}_t(\rho))^2 + \lambda \sum_{t=1}^{N} w_u(t)(L_u u_t(\rho))^2 \right]$$

where $w_y(\cdot)$ and $w_u(\cdot)$ are any nonnegative numbers. The flexibility offered by the time weightings $w_y(t)$ and $w_u(t)$ is that they allow one to put different weightings on different parts of the time responses. A particularly interesting application is when zero weightings are put on the transient response of the output response to a step change in the reference signal.

In this case the criterion becomes:

$$J_m(\rho) = \frac{1}{2N} E\left[ \sum_{t=t_0}^{N} (L_y \tilde{y}_t(\rho))^2 + \lambda \sum_{t=1}^{N} (L_u u_t(\rho))^2 \right]$$

and we say that a *mask* of length $t_0$ is put on the transient response of the tracking error.

By imposing a mask on the transient response one does not waste the available degrees of freedom in the controller parameters on the matching of a specific and entirely arbitrary transient response. Instead one can focus these parameters entirely on achieving a fast settling time. The cost achieved after the masked interval is always smaller than when no mask is used.

### 4.1.1 Simulations using weighted IFT algorithm – Improving the settling time

Consider the plant

$$P(s) = \frac{1}{s^2 + 0.1s + 1}$$

One wish to tune a PID controller in order to achieve a better settling time for the closed loop system.

Consider the standard form of the PID controller:

$$G_{PID}(s) = k_p \left[ 1 + \frac{1}{T_i s} + T_d s \right]$$

that for the physical realization we change in:

$$G_C(s) = k_p \left[ \frac{1 + T_d s}{1 + \tau s} + \frac{1}{T_i s} \right]$$

with $\tau = 2$ and the following initial PID parameters:

$k_p = 0.025$
$T_i = 2$
$T_d = 1$

so that the initial controller is: $G_C(s) = \dfrac{\rho_1 s^2 + \rho_2 s + \rho_3}{\tau s^2 + s}$

This yields the very sluggish response shown in the figure below.

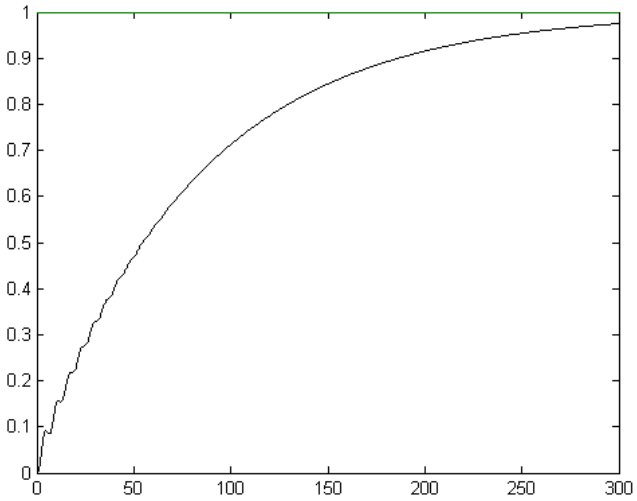



Fig. 4.1 Closed loop step(amplitude 1) response with initial PID parameters

We use in the IFT algorithm a Gauss-Newton approximation of the Hessian for the matrix $R_i$, a step size of 0.1 and the following desired response $y^d$:

$$y^d = T_d r = \frac{0.05}{s + 0.05} r$$

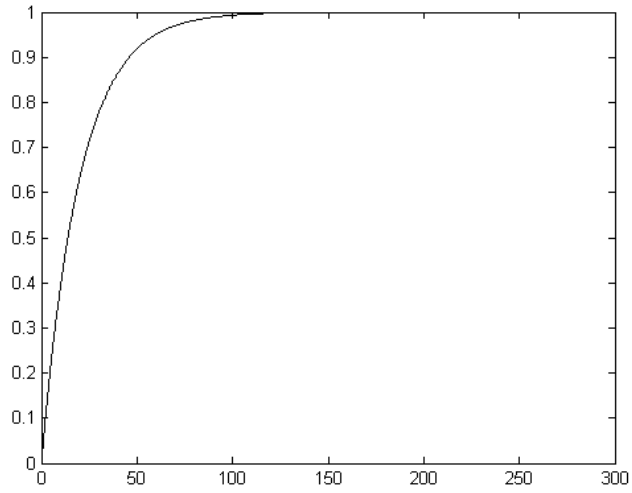



Fig. 4.2 Desired response to a step of amplitude 1

The application of the classical IFT criterion using the cost function

$$J(\rho) = \frac{1}{2N} \sum_{t=1}^{N} (y_t - y^d)^2$$

yields, after 15 iterations, the response:

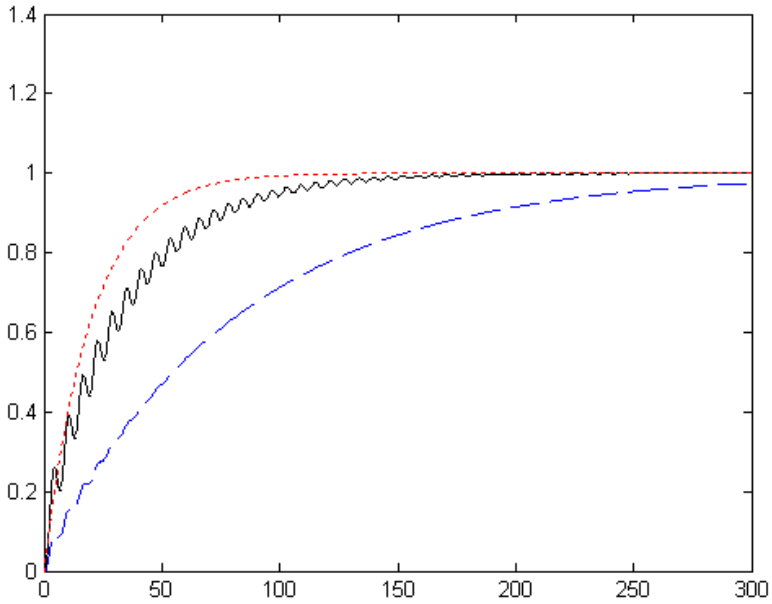



Fig. 4.3 Closed loop step response(full) obtained with
the classical IFT criterion and using the desired response(dotted)
Dashed curve: response with the initial PID parameters.

This response is very unsatisfactory; this is in large part due to an unfortunate choice of initial parameters.

With the use of a fixed mask of length $t_0$ =100 seconds, the minimization of the modified IFT criterion with the same initial parameters leads to the following closed-loop responses, obtained after 15 and 30 iterations respectively.

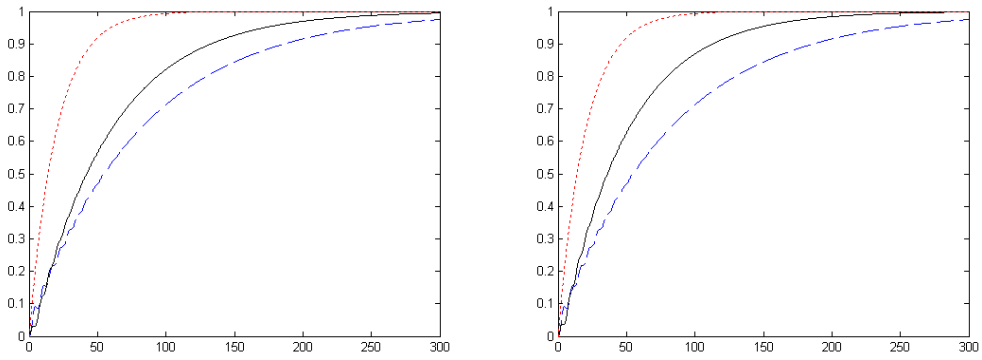


Fig. 4.4 Step response(full) after 15 iterations(left) and after 30 iterations(right) using a mask of length 100, the desired response(dotted) and the step response with the initial PID parameters(dashed)

This response is better than that obtained with a reference trajectory.

Finally a mask of decreasing length is used, with an initial length of 100 seconds, and with the same initial parameters again. At every iteration of the IFT scheme, the length of the mask is decreased by 10 seconds.

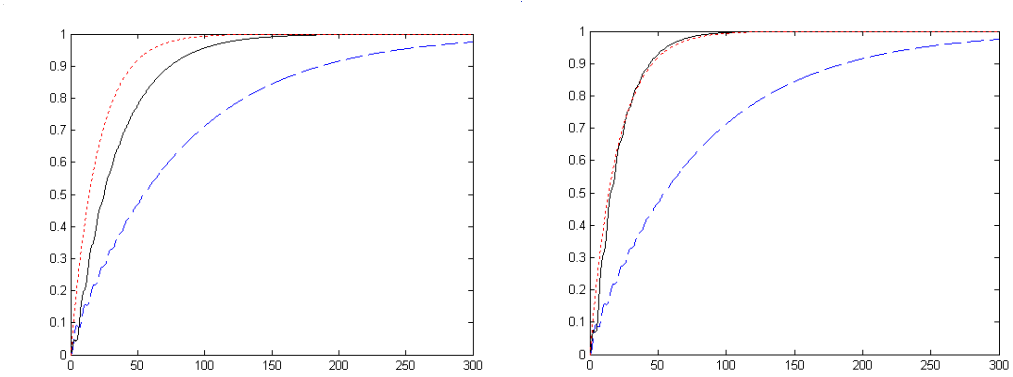



Fig. 4.5 Step response(full) after 15 iterations(left) and after 30 iterations(right) using a mask of decreasing length, the desired response(dotted) and the step response with the initial PID parameters(dashed)

Observe the dramatic improvement of the response due to the use of a mask of decreasing length, leading to a sequence of cost criteria (rather than a one-shot criterion), and to a different sequence of $\rho_i$ parameter vectors than resulted with the direct use of a mask of length $t_0$ =100 seconds.
Resuming, we can use the criterion

$$J(\rho) = \frac{1}{2N}\sum_{t=1}^{N}(y_t - y^d)^2$$

to compare the results.

The initial cost is $J_0$ = 0.0294 and with the implementation of the classical IFT criterion we get $J_{15}$ = 0.0020.

| **Using IFT with fixed mask** | **Using IFT with mask of decreasing length** |
|---|---|
| $J_{15}$ = 0.0169 | $J_{15}$ = 0.0034 |
| $J_{30}$ = 0.0117 | $J_{30}$ = $2.0882 \cdot 10^{-4}$ |

Fig. 4.6 The subscript denotes after how many iterations the values stem from.

The initial parameters are:

$\rho_{1_0} = 0.025$
$\rho_{2_0} = 0.05$
$\rho_{3_0} = 0.0125$

After 15 iteration of the classical IFT criterion:

$\rho_{1_{15}} = 0.0429$

$\rho_{2_{15}} = 0.1643$

$\rho_{3_{15}} = 0.0321$

Using the modified criterions:

| **IFT with fixed mask** | **IFT with mask of decreasing length** |
|---|---|
| $\rho_{1_{15}} = 0.0538\,;\ \rho_{1_{30}} = 0.0630$ | $\rho_{1_{15}} = 0.0898\,;\ \rho_{1_{30}} = 0.1460$ |
| $\rho_{2_{15}} = -0.0085\,;\ \rho_{2_{30}} = -0.0099$ | $\rho_{2_{15}} = -0.0219\,;\ \rho_{2_{30}} = -0.0356$ |
| $\rho_{3_{15}} = 0.0169\,;\ \rho_{3_{30}} = 0.0197$ | $\rho_{3_{15}} = 0.0294\,;\ \rho_{3_{30}} = 0.0478$ |

Fig. 4.7

### 4.1.2 Simulations using weighted IFT algorithm – Improving the overshoot

In this section we will present with an example how the modified criterion using time weightings can improve the response performance in terms of maximum overshoot.

Consider the plant (discrete time):

$$P(z) = \frac{0.1493z + 0.1095}{z^3 - 0.7413z^2}$$

and the discrete controller:

$$G_C(z) = \frac{1.6z^2 - 1.25z + 0.3}{z^2 - 0.608z - 0.392}$$

This yields a step response with a considerable overshoot shown below:

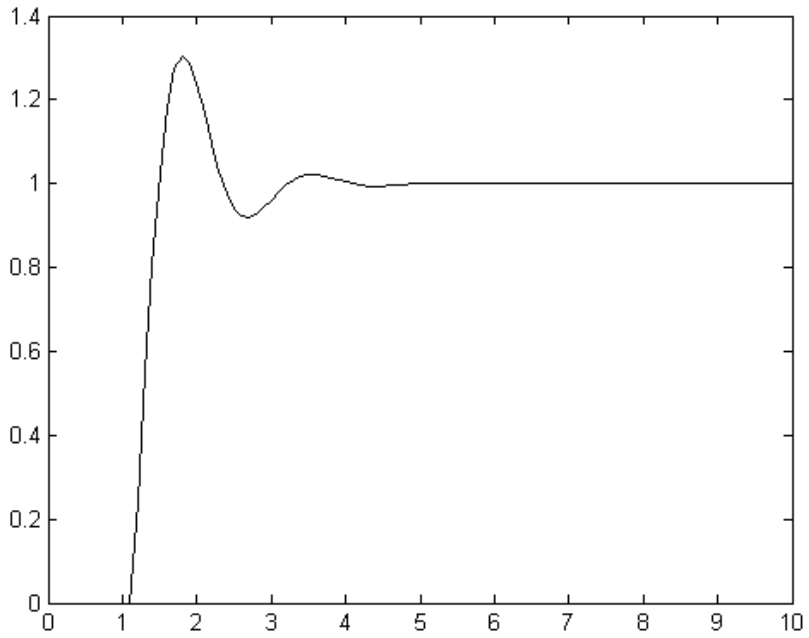



Fig. 4.8 Response to a step of amplitude 1 at t =1
with the initial controller parameters

Using in the classical IFT algorithm:

- a partial parametrization of the controllet (numerator)
- a cost function $J(\rho) = \sum_{t=1}^{N} (y_t - y^d)^2$
- a Gauss-Newton approximation of the Hessian for the matrix $R_i$
- a step size $\gamma_i = 0.2 \; \forall i$

we get, respectively after 4 and 9 iterations, the following step responses.

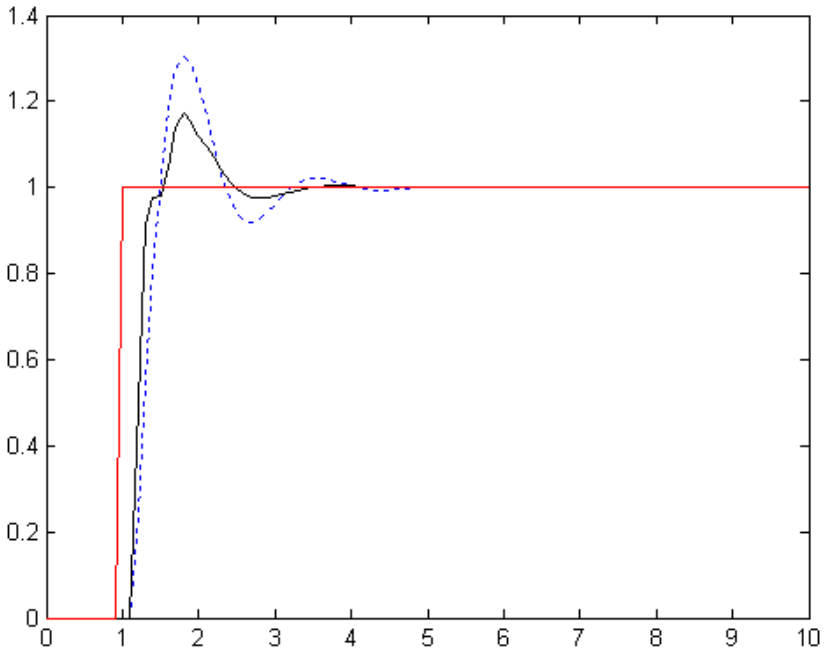



Fig. 4.9 Step response(full) after 4 iterations of the classical IFT algorithm and step response with the initial parameters(dotted)

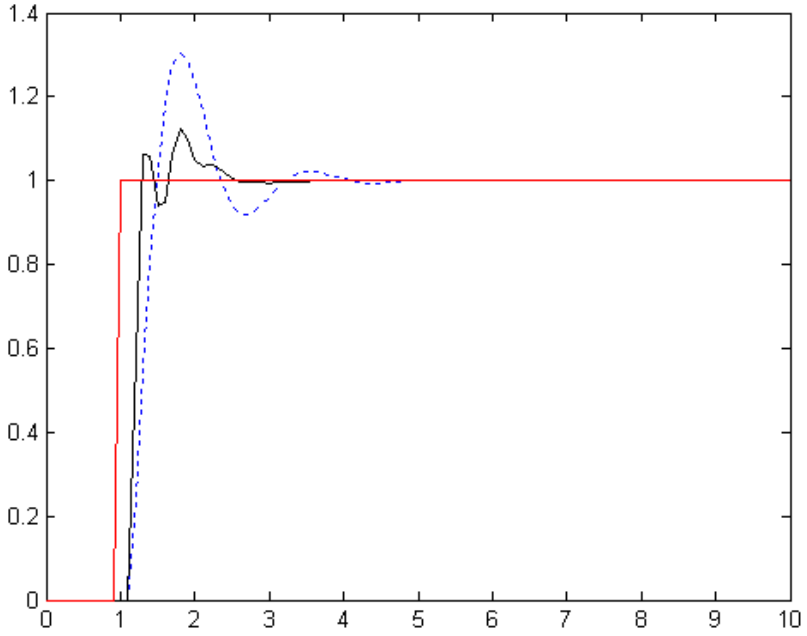



Fig. 4.10 Step response(full) after 9 iterations of the classical IFT algorithm and step response with the initial parameters(dotted)

We see that the classical IFT criterion yields a smaller maximum overshoot (and settling time).
Using a time weighting in the cost function, we will now show how we can get better results.

Considering the cost function

$$J(\rho) = \sum_{t=1}^{N} \left[ w_y(t)(y_t - y^d)^2 \right]$$

and applying a time weighting

$$w(t) = 1, \quad t \in [0, 1.5] \cup [2.25, 10]$$
$$w(t) = 5, \quad t \in (1.5, 2.25)$$

as shown in the following figure

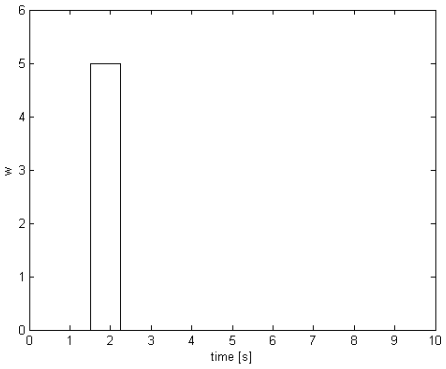



Fig. 4.11

we get the following step response:

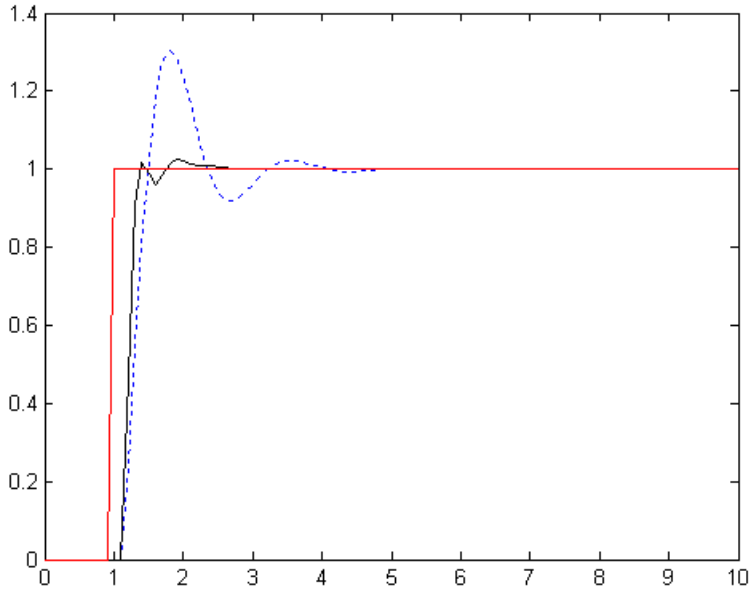


Fig. 4.12

that yields a much better behaviour with respect to the classical IFT criterion and was using only 4 iterations.

## 4.2 The BFGS method for the search direction

The matrix $R_i$ in the parameter update law determines the update direction and it is therefore crucial for the performance of the algorithm.
We have seen that a good choice is to let $R_i$ be an approximation of the Hessian. Especially if $\tilde{y}$ is small, the Gauss-Newton direction

$$R_i = \frac{1}{N}\sum_{t=1}^{N}\left(\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right]\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right]^T + \lambda\left[\frac{\partial u_t}{\partial \rho}(\rho_i)\right]\left[\frac{\partial u_t}{\partial \rho}(\rho_i)\right]^T\right)$$

is a desirable choise and the natural approximation is:

$$R_i = \frac{1}{N}\sum_{t=1}^{N}\left(est\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right]est\left[\frac{\partial y_t}{\partial \rho}(\rho_i)\right]^T + \lambda\, est\left[\frac{\partial u_t}{\partial \rho}(\rho_i)\right]est\left[\frac{\partial u_t}{\partial \rho}(\rho_i)\right]^T\right)$$

Another good choise is the Broyden-Fletcher-Goldfarb-Shanno(BFGS) method, one of the quasi-Newton methods.
One of the merits of the quasi-Newton methods is that the good estimation of Hessian matrix is given from the gradients of the cost function $J$ and the design parameters $\rho_i$. BFGS method is well known as a good optimization method [Hamamoto *et al*, 2003].
The renewal law to estimate the Hessian based on BFGS method is given as follows.

$$B^{(k+1)} = B^{(k)} + \frac{z^{(k)}(z^{(k)})^T}{(z^{(k)})^T s^{(k)}} - \frac{B^{(k)}s^{(k)}(s^{(k)})^T B^{(k)}}{(s^{(k)})^T B^{(k)} s^{(k)}}$$

where

$$B^{(k)} = B(\rho^{(k+1)})$$
$$s^{(k)} = \rho^{(k+1)} - \rho^{(k)}$$
$$z^{(k)} = J'(\rho^{(k+1)}) - J'(\rho^{(k)}) = \frac{\partial J}{\partial \rho}(\rho^{(k+1)}) - \frac{\partial J}{\partial \rho}(\rho^{(k)})$$

The superscript denotes the $k$-th iteration.
The initial matrix value of $B^{(0)}$ is an arbitrary positive definite matrix. Usually $B^{(0)}$ is chosen to be an identity matrix.
The following facts are well known about the BFGS method:

a) if $B^{(k)}$ is symmetric then $B^{(k+1)}$ is symmetric;
b) if $B^{(k)}$ is positive definite and $(z^{(k)})^T s^{(k)}>0$, then $B^{(k+1)}$ is positive definite.

When $(z^{(k)})^T s^{(k)}>0$ is not satisfied, let $B^{(k+1)}= B^{(k)}>0$. However such a case seldom occurs.
Notice that the BFGS method uses the same data as the Gauss-Newton method.

Thus the IFT parameter updating law becomes:

$$\rho_{i+1} = \rho_i - \gamma_i B_i^{-1}\frac{\partial J}{\partial \rho}(\rho_i)\,.$$

## 4.3 The Hamamoto-Fukuda-Sugie IFT Approach

We consider the two d.o.f. control system depicted in the following figure.

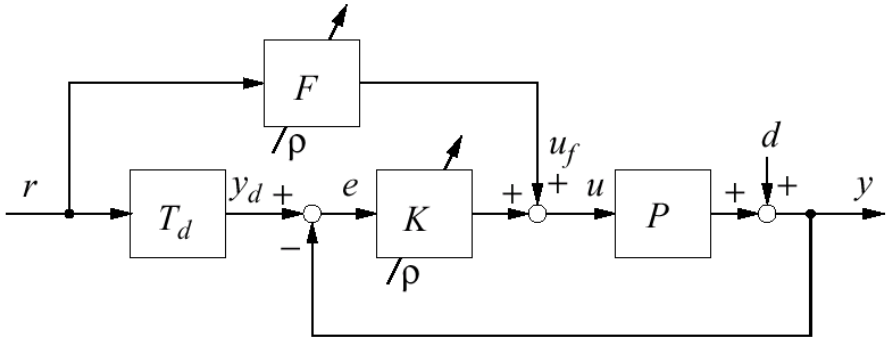



Fig. 4.13 Closed loop system with two-degree-of-freedom controller

Since any two d.o.f. control system can be transformed in to this configuration [Sugie & Yoshikawa, 1986], there is no loss of generality here.
In figures, $P$ is the SISO plant, and $K$ and $F$ denote the controllers to be designed. The scalar u, y and r are the plant input, its output and the reference, respectively.
The signal $d$ denotes an external test signal which is used for the estimation of the performance of the closed loop system and we assume that we can choose it arbitrarily.
Note that all subsystems are known except for the plant $P$, and all the signals are observable.
As we have seen before, the system $T_d$ is the desired closed loop model which is given in advance, and $y_d = T_d r$ is the desired trajectory which $y$ should track.
One distinguishing feature of the control structure shown in the figure is as follows.
Let $T_{yr}$ denote the closed loop transfer function from r to y, then it is known that the so-called conditional feedback property holds, that is,

$$T_{yr} = T_d \quad \forall K$$

is satisfied whenever $F = T_d / P$ holds. In other words, $K$ does not play any role for tracking property in the nominal case, and the main role of $K$ is to suppress the effect of disturbances and plant uncertainties [Sugie & Yoshikawa, 1986]. While apparently, the role of $F$ is to specify the tracking property.

The controllers K and F are supposed to be uniquely determined by the design parameters $\rho_a$ and $\rho_b$ respectively. We use the symbols $K(\rho_a)$ and $F(\rho_b)$ in order to indicate their dependence on the parameters, explicitly. For I/O signals, we use the similar symbols such as $u(\rho_a)$ and $y(\rho_b)$. From the above observation, it would be natural to tune $K(\rho_a)$ and $F(\rho_b)$ do as to achieve low sensitivity and desired tracking property, respectively.
In the following we assume that one stabilizing controller pair $(K(\rho_a^{(0)}), F(\rho_b^{(0)}))$ is given in advance. And each experiment is performed in the finite time interval $[0, t_f]$.

The strategy proposed by Hamamoto, Fukuda and Sugie (*HFS*-method) is the *separeted tuning* of the feedback and feedforward controllers. In particular, as for the feedback controller, they concentrate on tuning to achieve low sensitivity instead of tracking property, while the feedforward controller is tuned from the viewpoint if command tracking property.

## 1 - Feedback controller tuning

In this section we concentrate on the tuning of the feedback controller $K(\rho_a)$.
In order to achieve low sensitivity, this new method try to minimize the weighted sensitivity function $W(s)S(s)$ in a certain case, where $W(s)$ is the given weighting transfer function and

$$S(s) = \frac{1}{(1 + P(s)K(s))}$$

is the sensitivity function.

To achive this goal the *HFS method* performs the following two experiments (A,B).

### Experiment A

Set $u_f(t) = 0$ and $y_d(t) = 0$ in figure 4.13. Then the controlled system is shown as the figure below.

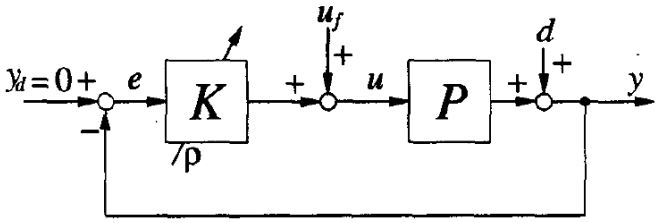



Fig. 4.14

Inject the test signal $d(t)$ which is calculated from

$$d(s) = W(s)n(s)$$

where $n$ is a virtual white signal which has zero mean and an appropriate covariance. Then, let $y(\rho_a)$ and $u(\rho_a)$ be the corresponding I/O signals of the plant.

For these I/O signals, we solve the following problem.

**FB Controller Design Problem**

Find the parameter $\rho_a^*$ which minimizes

$$J_{fb}(\rho_a) = \|y(\rho_a)\|^2 + \lambda_a \|u(\rho_a)\|^2$$

with

$$\|x(t)\|^2 = \langle x(t), x(t) \rangle = \int_0^{t_f} x^T(\tau) x(\tau) d\tau$$

through iteration of experiments, where $\lambda_a$ is a positive constant weighting scalar. Note that the transfer function from *d* to *y* is equal to the sensitivity function S. Therefore, if $t_f \to \infty$ and $\lambda_a \to 0$, the above cost is equal to the square of the $H_2$ norm of *WS* because of the whiteness of the virtual signal *n*.

Now we apply the HFS method, that derives form the standard IFT algorithm based on Hjalmarsson and Birkeland (1998) and Hamamoto and Sugie (1999).

The parameter $\rho_a$ is updated by

$$\rho_a^{(i+1)} = \rho_a^{(i)} - \gamma^{(i)} B(\rho_a^{(i)})^{-1} J_{fb}'(\rho_a^{(i)})$$

where $\rho_a^{(i)}$ denotes the value of $\rho_a$ at the *i*-th renewal, and $J_{fb}'(\rho_a)$ is the gradient of $J(\rho_a)$ with respect to $\rho_a$.
As for the standard IFT method, the matrix $B(\rho_a^{(i)})$ gives the update direction and can be for example a Gauss-Newton approximation of the Hessian of *J* or a matrix found with the BFGS method.
In our discussion and following example the scalar $\gamma^{(i)}$ is considered as a fixed step-size.

First, we calculate $J_{fb}'(\rho_a)$ from the I/O data of experiments without any model of *P*. The *j*-th entry of $J_{fb}'$ is given by

$$J_{fb,j}'(\rho_a) = 2\langle y_j'(\rho_a), y(\rho_a) \rangle + 2\lambda_a \langle u_j'(\rho_a), u(\rho_a) \rangle$$

where $(\cdot)'(\rho_a)$ denotes the derivative with respect to $\rho_a$ and the subscript *j* means the *j*-th element of the vector.
In order to calculate $J_{fb}'(\rho_a)$, we perform the following experiment.

**Experiment B**

Set $d(t) = 0$ and $y_d(t) = 0$ in fig. 4.13 and inject the signal $u_f = y(\rho_a)$ which is obtained by Experiment A, and let $f_u(\rho_a)$ and $f_y(\rho_a)$ be the corresponding I/O signals of the plant.

Since

$$y(\rho_a) = \frac{1}{1+PK(\rho_a)}d$$

$$u(\rho_a) = K(\rho_a)y(\rho_a)$$

hold from Experiment A, the derivative with respect to $\rho_a$ gives us

$$\begin{aligned} y_j^{'}(\rho_a) &= -\frac{PK_j^{'}(\rho_a)}{(1+PK(\rho_a))^2}d \\ &= -K_j^{'}(\rho_a)\frac{P}{1+PK(\rho_a)}\frac{1}{1+PK(\rho_a)}d \\ &= -K_j^{'}(\rho_a)\frac{P}{1+PK(\rho_a)}y(\rho_a) \\ &= -K_j^{'}(\rho_a)f_y(\rho_a) \end{aligned}$$

$$\begin{aligned} u_j^{'}(\rho_a) &= K_j^{'}(\rho_a)y(\rho_a) + K(\rho_a)y_j^{'}(\rho_a) \\ &= K_j^{'}(\rho_a)\frac{P}{1+PK(\rho_a)}y(\rho_a) \\ &= K_j^{'}(\rho_a)f_u(\rho_a) \end{aligned}$$

where we used the relations

$$f_y(\rho_a) = \frac{P}{1+PK(\rho_a)}y(\rho_a)$$

$$f_u(\rho_a) = \frac{1}{1+PK(\rho_a)}y(\rho_a)$$

from Experiment B.

Therefore we can calculate $J_{fb}^{'}(\rho_a)$ from the data $\left(y(\rho_a),\ u(\rho_a),\ y^{'}(\rho_a),\ u^{'}(\rho_a)\right)$ through Experiments A and B.

The procedure of feedback controller tuning is stopped when, for a given scalar $\varepsilon > 0$ in advance,

$$J_{fb}(\rho_a^{(k)}) - J_{fb}(\rho_a^{(k+1)}) < \varepsilon$$

and we regard $\rho_a^{(k)}$ as the sub-optimal parameters $\rho_a^*$.

## 2 - Feedforward controller tuning

Now we fix the feedback controller as $K = K(\rho_a^{(k)})$, and tune the feedforward controller $F(\rho_b)$.
The objective here is to make $y(t)$ track $y_d(t)$ more accurately for the given reference $r(t) = r_0$.
For this purpose we perform the following experiment.

### Experiment C

Inject $r$ to the two d.o.f. control system in fig. 4.13 with $F(\rho_b)$ and $K$, and let $y(\rho_b)$ and $u(\rho_b)$ be the corresponding I/O signals of the plant.
The system is shown in the figure below.

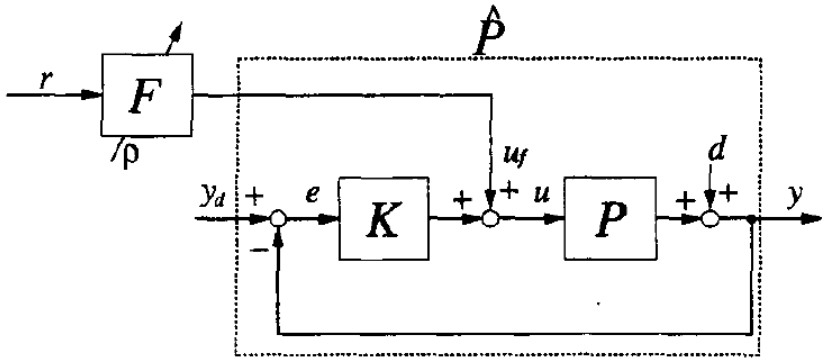



Fig. 4.15

For this system we solve the following problem.

### FF Controller Design Problem

Assume the controllers $K$ and $F(\rho_b^{(0)})$ which stabilize the system shown in fig. 4.13 are given. Find the parameter $\rho_b^*$ which minimizes the cost function

$$J_{ff}(\rho_b) = \left\| y(\rho_a) - y_d \right\|^2 + \lambda_b \left\| u(\rho_b) \right\|^2$$

through iteration of experiments, where $\lambda_b$ denotes the positive constant weighting scalar.
The IFT procedure is almost the same as in the case of FB controller tuning by replacing $J_{fb} \to J_{ff}$, $K \to F$ and $\rho_a \to \rho_b$.
The $j$-th entry of the gradient of the cost function is given by:

$$J'_{ff,j}(\rho_b) = 2\left\langle y'_j(\rho_b), y(\rho_b) - y_d \right\rangle + 2\lambda_b \left\langle u'_j(\rho_b), u(\rho_b) \right\rangle$$

To calculate it we need the following experiment.

**Experiment D**

Set $d(t) = 0$ and $y_d(t) = 0$ in fig. 4.13 and inject the signal $u_f = r$, and let $r_u(\rho_b)$ and $r_y(\rho_b)$ be the corresponding I/O signals of the plant.
From Experiments C and D, the following relations hold:

$$r_u = \frac{1}{1+PK} r$$

$$r_y = \frac{P}{1+PK} r$$

$$u(\rho_b) = \left( \frac{1}{1+PK} F(\rho_b) + \frac{K}{1+PK} T_d \right) r$$

$$y(\rho_b) = Pu(\rho_b)$$

While, from the derivative with respect to $\rho_b$, we have:

$$u_j'(\rho_b) = F_j'(\rho_d) \frac{1}{1+PK} r = F_j'(\rho_b) r_u$$

$$y_j'(\rho_b) = F_j'(\rho_d) \frac{P}{1+PK} r = F_j'(\rho_b) r_y$$

Therefore, the data $\left(u(\rho_b),\ u'(\rho_b),\ y(\rho_b),\ y'(\rho_b)\right)$ are obtained through Experiments C and D, and we can calculate $J_{ff}'$ from these data.

**A SIMULATION EXAMPLE**

Let us introduce a simple model of a robot joint expressed by the Laplace function:

$$G_C(s) = \frac{1}{J \cdot s^2}$$

where $J$ is the joint moment of inertia.

The example is taken for [Scalamogna, 2001]. In that work, the purpose was to improve the performance of the controlled system by adding to the system a suitable external control signal using Iterative Learning Control (ILC). We will examine the same example in terms of IFT.

From $G_c(s)$ we obtain $G(z)$ that is the discrete-time version of $G_c(s)$ including the Zero Order Hold (ZOH):

$$G(z) = \frac{T_s^2}{2J}\frac{z+1}{(z-1)^2} = G_{gain} \cdot \frac{z+1}{(z-1)^2}$$

where $T_s$ is the sampling time.

The joint is controlled by a PD feedback controller $K(z)$ and by a feed-forward controller $F_f(z)$.

$$G(z) = k_p + \frac{k_d}{T_s}\frac{z-1}{z} = \frac{k_p T_s + k_p}{T_s}\frac{z - \frac{k_d}{k_p T_s + k_d}}{z}$$

hence

$$K(z) = F_{gain} \cdot \frac{z - F_{zero}}{z}$$

and

$$F_f(z) = \frac{J}{T_s^2} \cdot \frac{(z-1)^2}{z^2} = F_{f\,gain} \cdot \frac{(z-1)^2}{z^2}$$

with

$J$ = 0.0094 Ns$^2$
$k_p$ = 12.7
$k_d$ = 0.4
$T_s$=0.001 s

The following figure shows the scheme we are considering (notice the disturbance added).

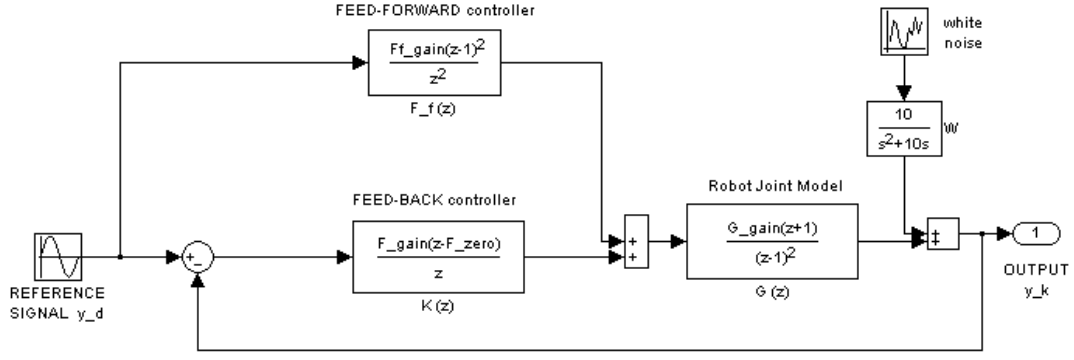



Fig. 4.16

We choose the following parametrizations of the two controllers:

$$K(z) = \frac{\rho_1 z + \rho_2}{z}$$

$$F_f(z) = \frac{\rho_3 z^2 + \rho_4 z + \rho_5}{z^2}$$

with the initial parameters:

$\rho_1 = 412.7$
$\rho_2 = -400$
$\rho_3 = 9400$
$\rho_4 = -18800$
$\rho_5 = 9400$

The reference trajectory chosen in the experiment is $y_d = \sin(2\pi t)$.
The initial behaviour is shown in the pictures below.

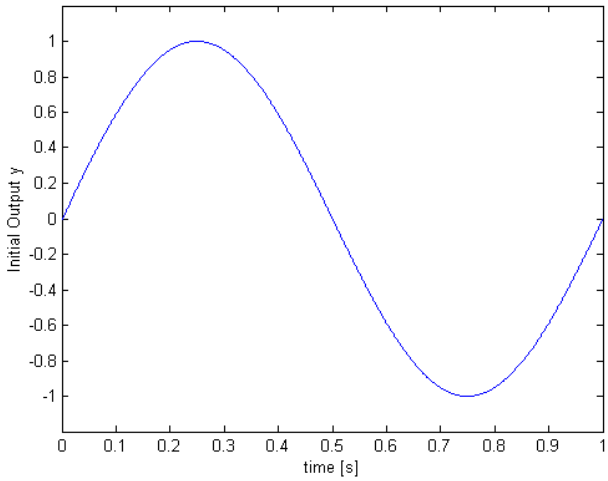



Fig. 4.17

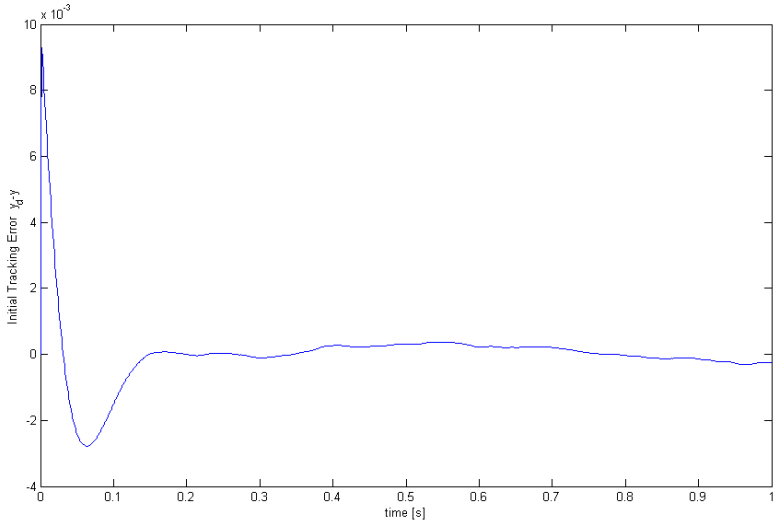



Fig. 4.18

We wish to improve the performance of the controlled system by using the *HFS*-IFT approach.

We perform 13 iteration of the IFT algorithm to tune the feedback controller $K$ with a Gauss-Newton approximation of the Hessian for the matrix $B_i$ , a step size of 0.2 and $\lambda_a = \lambda_b = 0$.
The final parameters are:

$\rho_1 = 4047.9$

$\rho_2 = -3983.7$

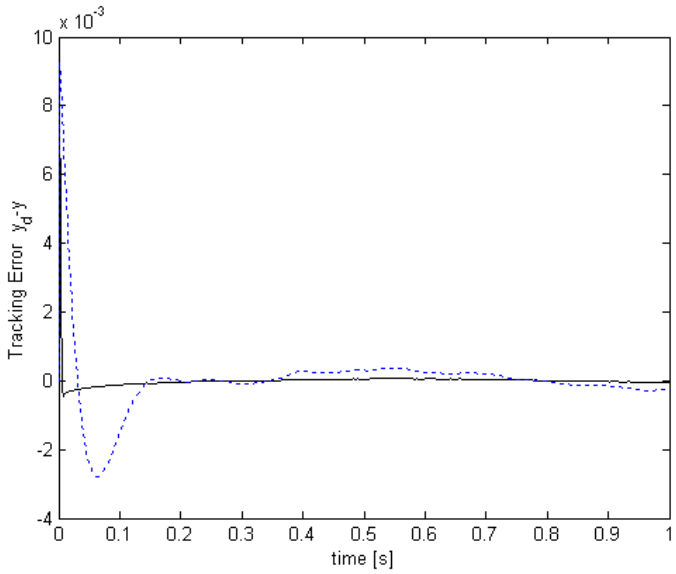



Fig. 4.19

and the corresponding tracking error:

By 13 iteration we pass from an initial cost $J_{fb_0} = 0.2439 \cdot 10^{-5}$ to a final cost $J_{fb_{13}} = 0.0047 \cdot 10^{-5}$.
The following figure shows the transition of the cost $J_{fb}$ by iterations.

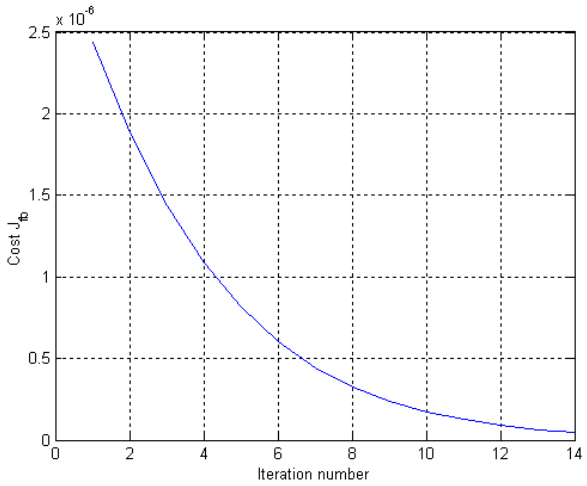



Fig. 4.20

From this figure we can see that the cost $J_{fb}$ decreases by the method proposed by Hamamoto, Fukuda and Sugie.

The following figure shows the output response corresponding to the test signal *d*.

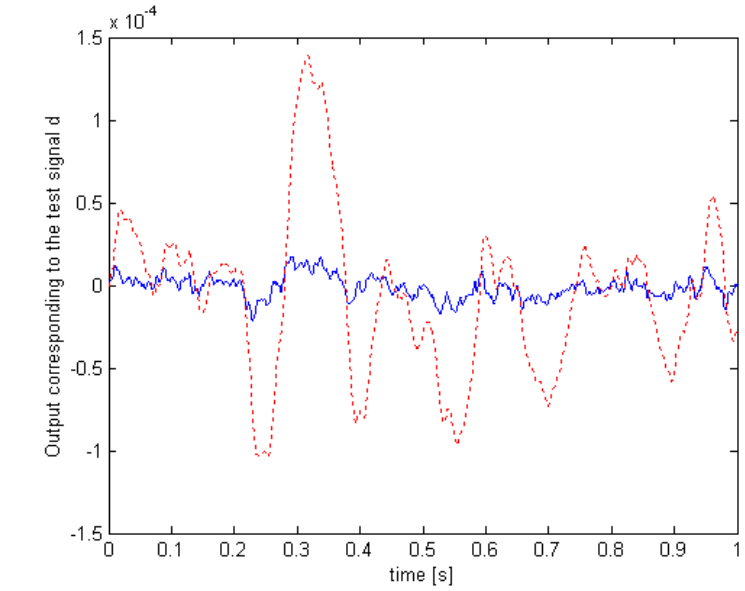



Fig. 4.21 Full: response after feedback tuning
Dotted: response with the initial controller

This figure shows that lower sensitivity is achieved via IFT. The gain and phase plots of sensitivity are shown in the figure below.

The obtained controller achieves a satisfactory property in the low-frequency domain. These figures show that the proposed IFT method works well in order to achieve low sensitivity.

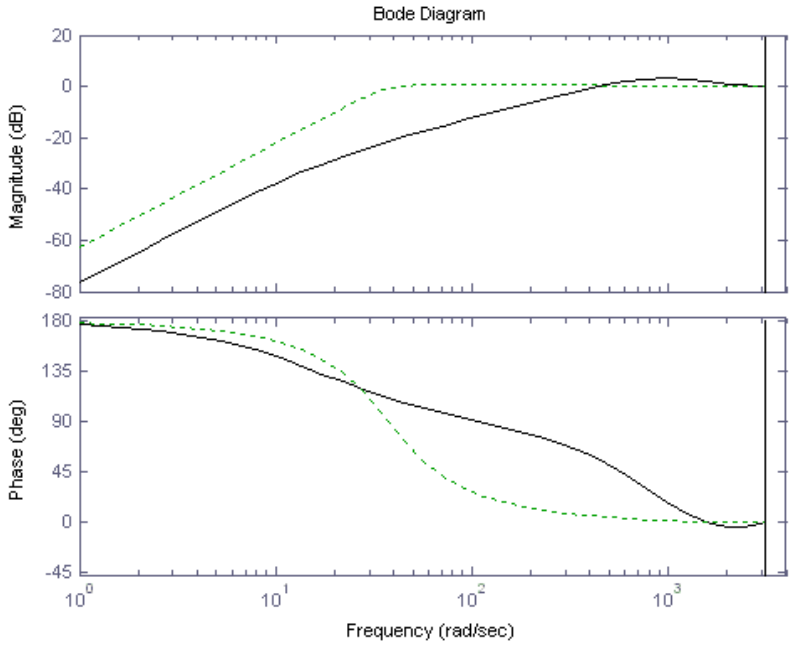



Fig. 4.22 Sensitivity function – Full: final controller after feedback tuning
Dotted: initial controller

We now fix the feedback controller as found after the previous 13 iterations and we perform 10 iterations with the purpose to tune the feedforward controller $F_f$.

By ten iterations, we pass from an initial cost $J_{ff_0} = 0.1402 \cdot 10^{-3}$ to a final cost $J_{ff_{10}} = 0.1372 \cdot 10^{-3}$.

The following figure shows the transition of the cost $J_{ff}$ as a function of iteration number.

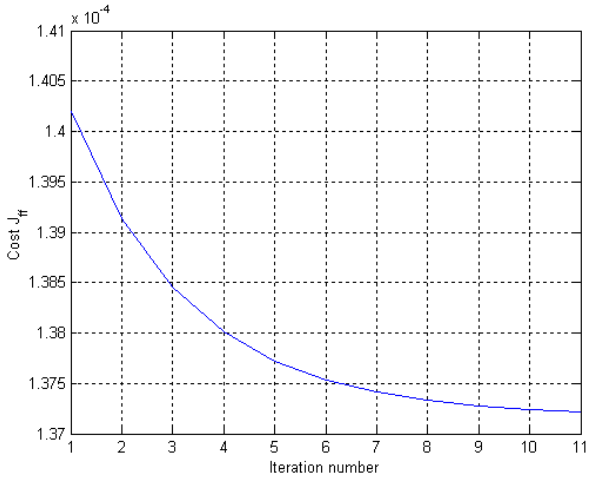



Fig. 4.23

The following figure shows the tracking error after the tuning of the feedforward controller $F_f$.

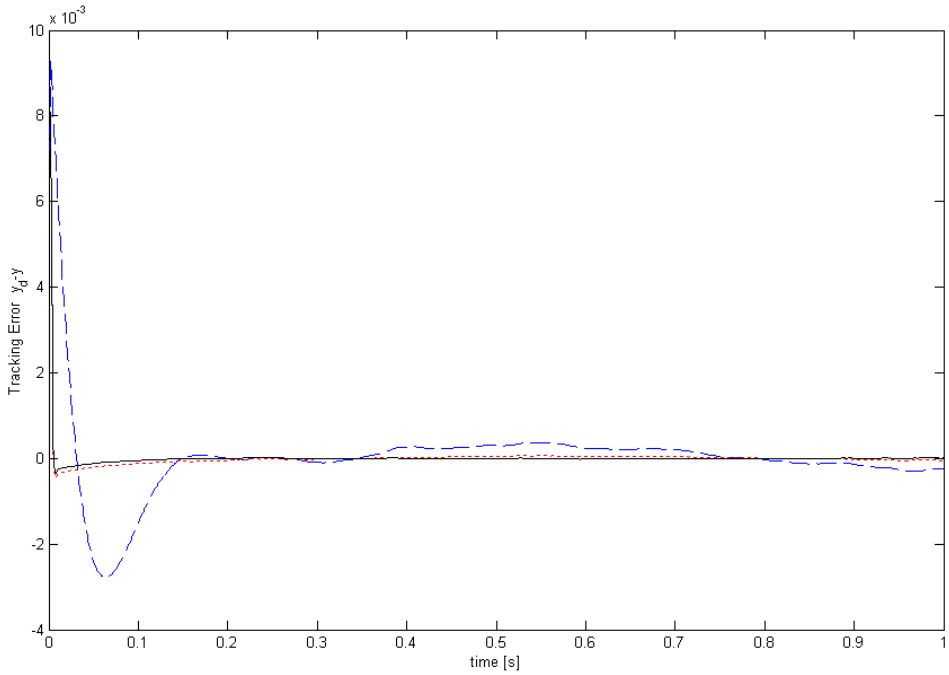



Fig. 4.24 Tracking error -Full: after feedforward tuning - Dotted: after feedback tuning - Dashed: initial

The final parameters are:

$\rho_1 = 4047.9$
$\rho_2 = -3983.7$
$\rho_3 = 9334.1$
$\rho_4 = -18669$
$\rho_5 = 9334.8$

Summarizing, the *HFS*-method use separate tuning of the feedback and feedforward controllers. As for the feedback controller, the point is focused on tuning in order to achieve low sensitivity instead of tracking property, while the feedforward controller have been tuned from the viewpoint of enhancement of tracking performance.

The experimental results show the effectiveness of the proposed IFT method.

# 5. Applications on an Industrial Robot

## 5.1 The ABB industrial robot Irb-2000 – brief description

The robot used in the experiments is an ABB Irb-2000 industrial robot.
The robot has seven links which are connected by six joints, as shown in the following figure.

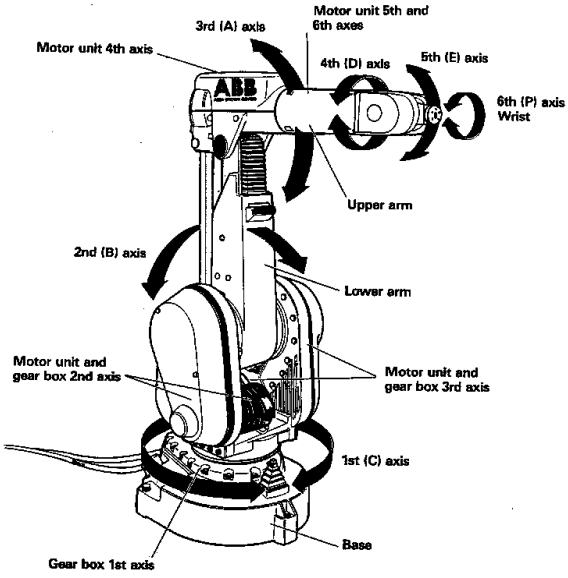



Fig. 5.1

Fig. 5.2

It is built up by two big arms and a wrist. Joint 2 (axis B in the figure is used to move the lower arm back and forth, whereas joint 3 (A) moves the upper arm up and down. Joint 4 (D) is used to turn the wrist unit and joint 5 (E) bends the wrist unit around its center. The sixth joint (F) is used to turn the robot end effector, which is mounted on the tip of the wrist (the end effector is not shown in the figure). Finally, joint 1 (C) turns the entire robot around its base.

The robot system has different built-in controllers, one for the control of each joint angle. These controllers are cascaded PID controllers.

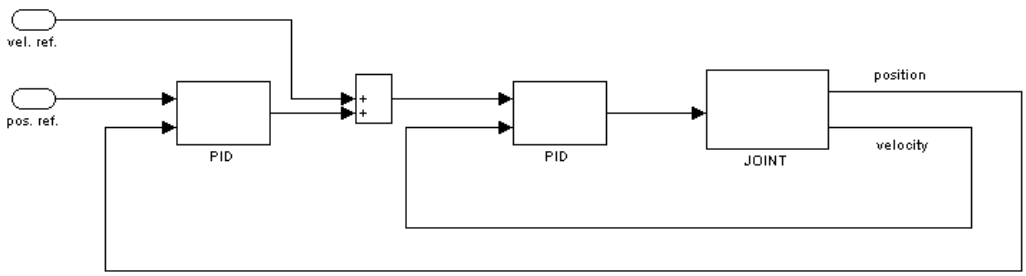



Fig. 5.3

## 5.2 The experimental platform

The experimental platform consists of:

- reconfigured Irb-2000 robot system (robot and control cabinet);
- VME based board computer system (target system);
- Host computer system consisting of Sun workstations (host system);
- Ethernet connection between host and target.

The Irb-2000 is controlled from VME-based embedded computers. Sun workstations are used for software development and control engineering, as well as for robot operator interaction.

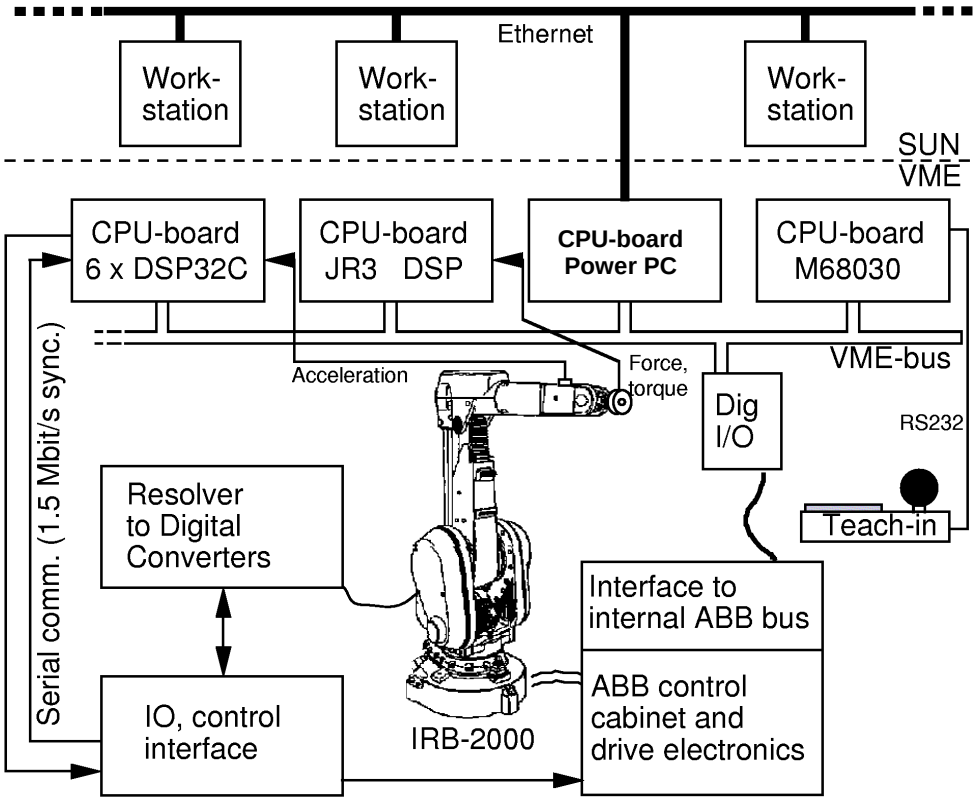



Fig. 5.4

The above figure shows the Irb-2000 part of the laboratory. Signals from internal sensors of the robot to the VME system go via the sensor interface to the DSP board connected to the VME bus.
The master computer in the VME computer is based on Power PC processor. Supervision and safety functions are implemented on a M68030 board, well separated from the rest of the system to prevent damage of the robot. Digital Signal Processors(DSP) are used for low-level control and filtering of sensors signals. Sensors requiring very high data bandwidths are connected directly to the DSP boards. An additional DSP board belongs to the force-torque sensor.

## 5.3 The Matlab-robot connection

By the Sun workstations on host-level it is possible to define and write the programs to control the robot or send it references to be tracked. That can be done inside the Matlab environment.

A Matlab program called *Exc_handler* (excitation handler) is available for simple excitation experiments on the robot. This program can be used to define velocity and position references to the robot servos. The inputs can be steps, ramps, sinusoids, noise and other arbitrary signals from the Matlab workspace.

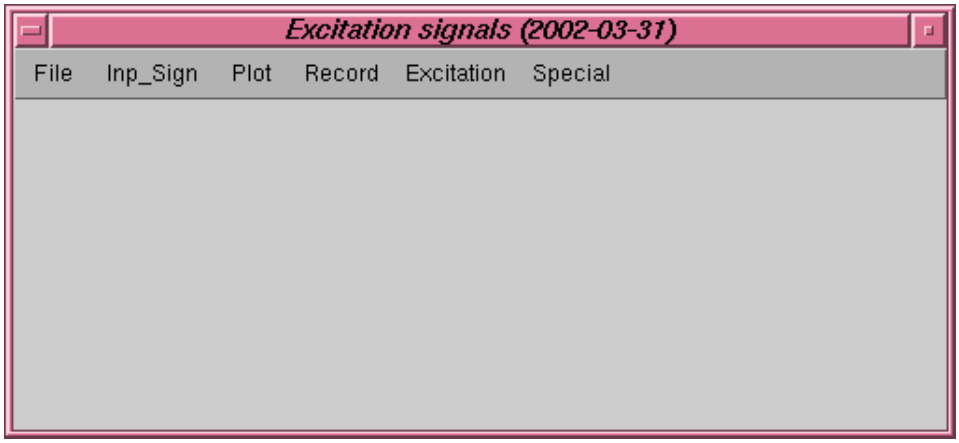



Fig. 5.5

A lot of signals can be recorded during the excitation. These include input torques, position measurements, differentiated position (velocity), and force and torque measurements from the force sensor. The recorded signals can be exported to the Matlab workspace for plotting and data processing.

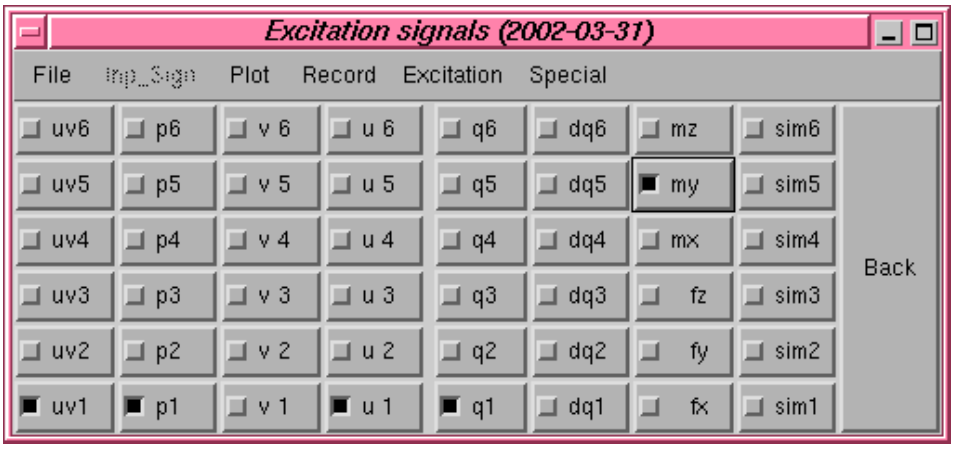



Fig. 5.6

## 5.4 The Simulink control interface

A Simulink control interface is available to handle the connection between the robot and the Matlab environment. The figure below represents one robot joint.

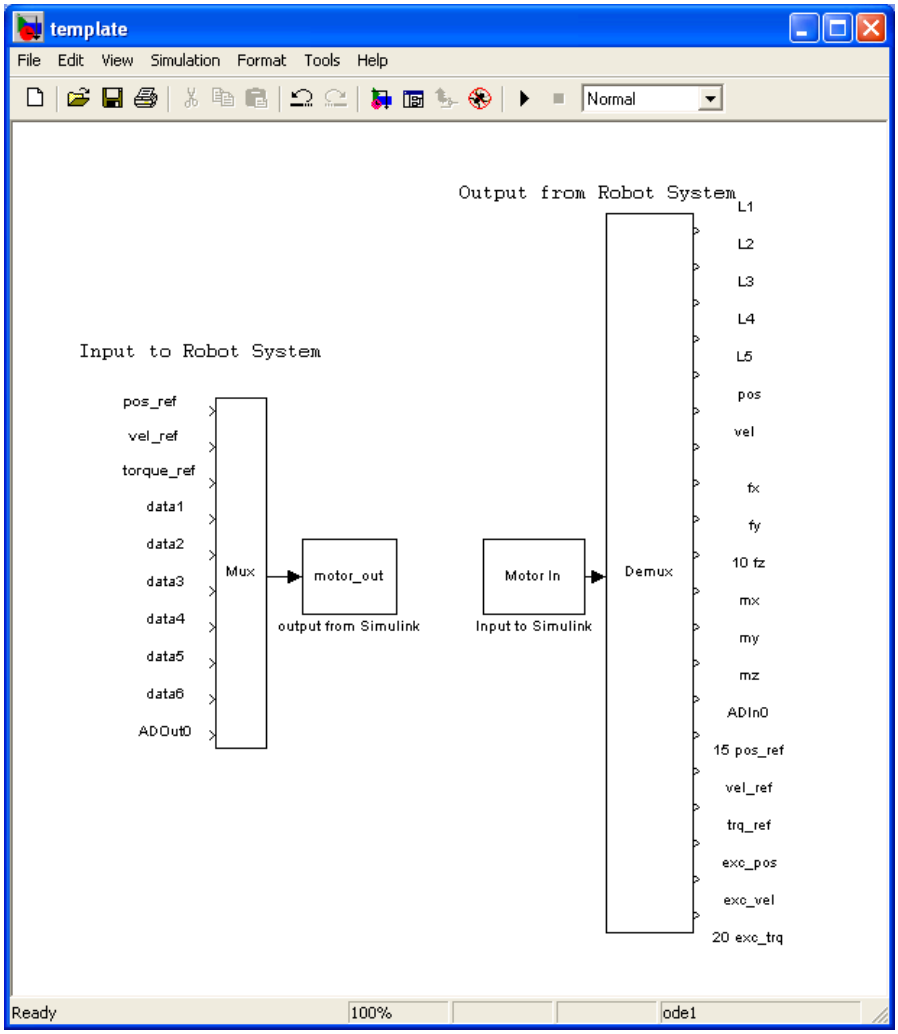



Fig. 5.7

With this interface, it is possible to connect and log signals with the Excitation handler (*Exc_handler*) and also to build our own controller that we will connect to the *torque_ref* (torque reference) signal.

After we have built our controller, the Simulink scheme can be automatically translated into C code and downloaded directly into the robot control unit.

To be able to use our own controller for joint *k* we need to activate the new controllers for the robot. This operation can be done using the matlab program *activate_simuJ*(*k*).
To reactivate the original controllers for the joint we use *activate_simuJ*(*0*).

With the matlab programs *RegOff*(*i*) and *RegOn*(*i*) we can remove and reinstall, respectively, the original controller for the specified *i*-th joint.

## 5.5 Application of IFT to a robot joint model

Before we come to the real application we will study some results of the IFT algorithm for tuning a controller for a model of the base joint (joint 1) for the robot.

The joint 1 model structure we consider is the following:

$$G(s) = \frac{1}{s}\frac{a}{s+1}\frac{s^2 + 2\omega_1\xi_1 s + \omega_1^2}{s^2 + 2\omega_2\xi_2 s + \omega_2^2}$$

The values used in the simulations are:

$a = 7$
$\omega_1 = 15$
$\omega_2 = 17$
$\xi_1 = 0.4$
$\xi_2 = 0.45$

Notice the presence of an integrator in the process model.

We control the system with a standard PID control structure:

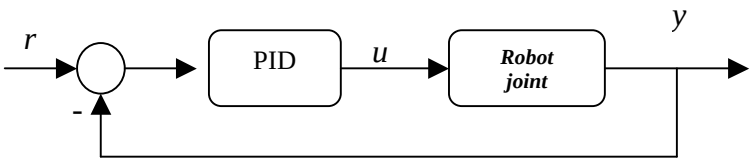



Fig. 5.8

$$G_C(s) = k_p\left[1 + \frac{1}{T_i s} + T_d s\right]$$

For the realization we choose:

$$G_C(s) = k_p\left[\frac{1 + T_d s}{1 + \tau s} + \frac{1}{T_i s}\right]$$

with $\tau = 0.01$.

### Choice of the initial PID parameters - AMIGO tuning rules

We choose to start with initial parameters $Kp$, $Ti$ and $Td$ found using the *AMIGO* conservative tuning rules [Åström & Hägglund, 2003].

The AMIGO rules, in the case when an integrator is present in the process, can be resumed in the following way.

We make an open loop step response experiment in which we measure the quantities $K_v$ and $L$ of the figure, that are the slope and the intersection with the time axis of the straight line $r$, respectively.

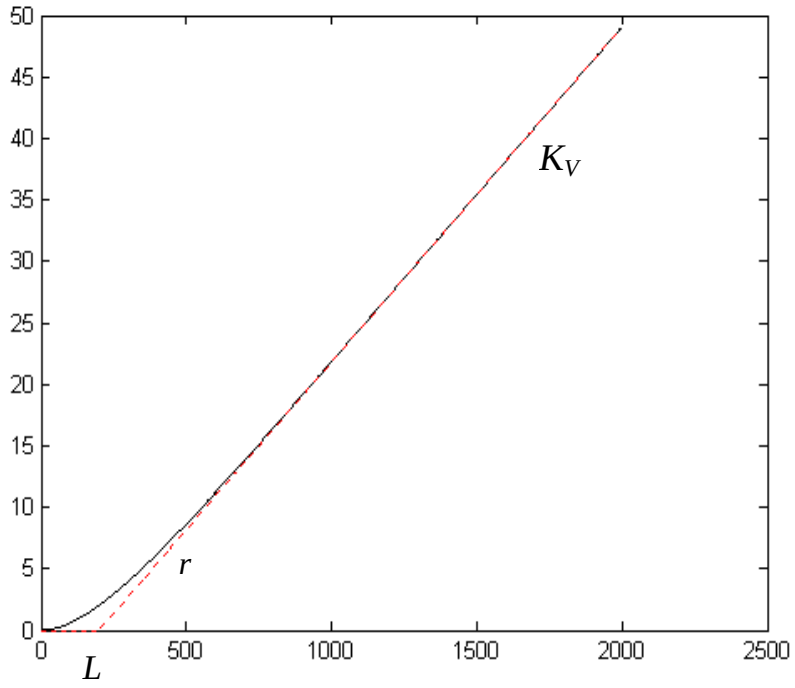



Fig. 5.9 Open loop step response of the system $G(s)$

Then the AMIGO rules give the following quantities of the PID parameters:

$$k_p = 0.45\frac{1}{K_V L}$$
$$T_i = 8L$$
$$T_d = 0.5L$$

In our specific case, we measured the quantities:

$$K_V = 5.4499$$
$$L = 1.001$$

So that we get the initial PID parameters for the robot joint 1 controller:

$$k_{p0} = 0.0825$$
$$T_{i0} = 8.008$$
$$T_{d0} = 0.5005$$

These parameters yield the following initial step response:

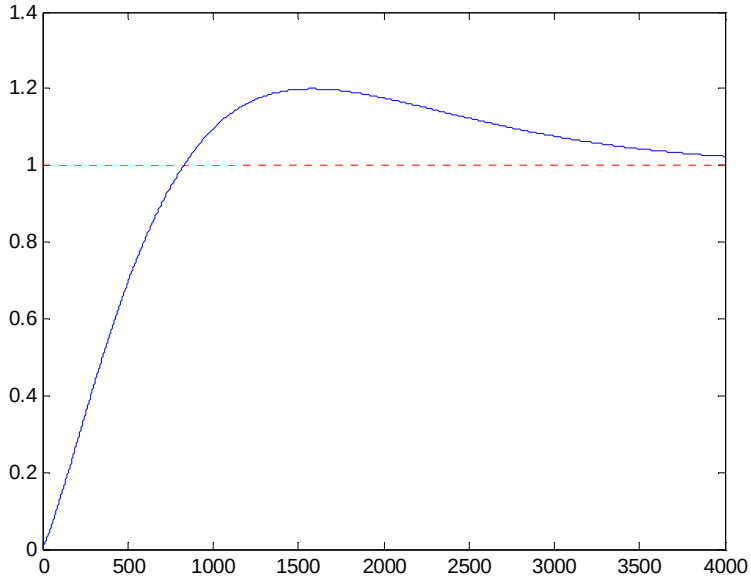



Fig. 5.10 Initial step response of the robot joint 1 model

Now we apply the standard IFT algorithm with the aim of improving the robot joint 1 response.

We condider the following desired output response to the reference signal $r$ chosen as a step:

$$y^d = T_d r = \frac{2.2}{s^2 + 3s + 2.2} r\,.$$

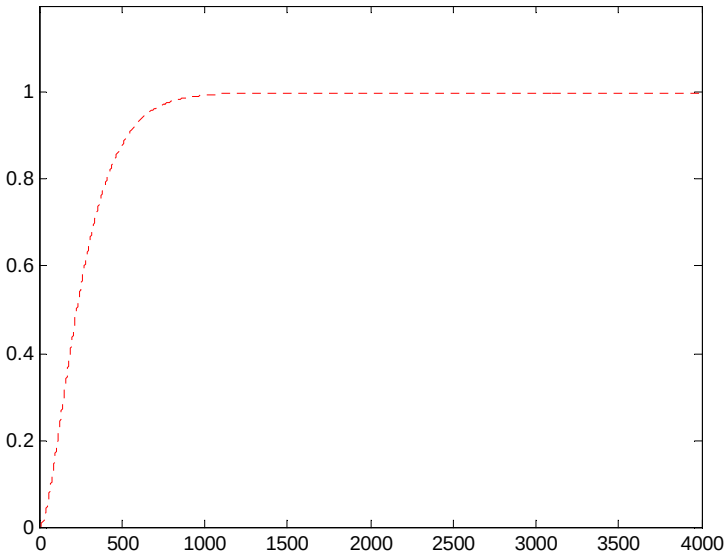



Fig. 5.11 Joint 1 - Desired output $y^d$

The application of the classical IFT criterion using the cost function

$$J(\rho) = \frac{1}{2N}\sum_{t=1}^{N}(y_t - y^d)^2$$

and considering:

- a Gauss-Newton approximation of the Hessian for the matrix $R_i$
- a step size $\gamma = 0.1$

yields, after respectively 10, 20, 30 and 40 iterations, the responses:

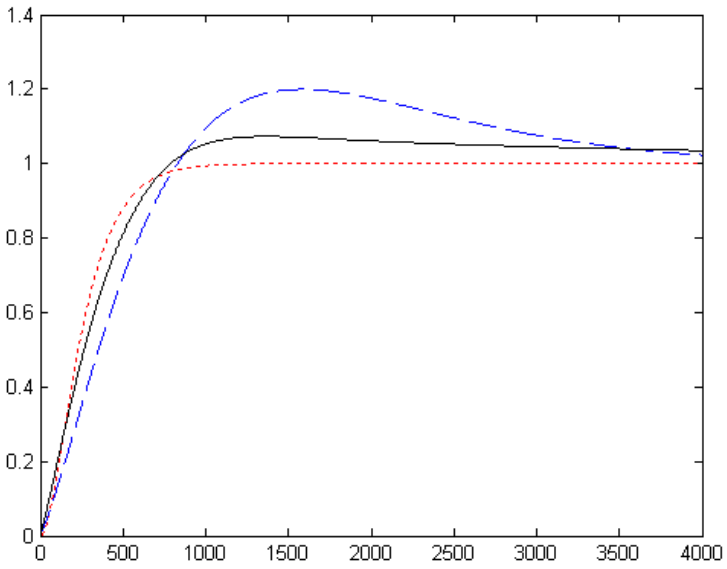



Fig. 5.12 Full line: final Joint 1 step response after 10 iterations
Dashed: initial response - Dotted: desired output

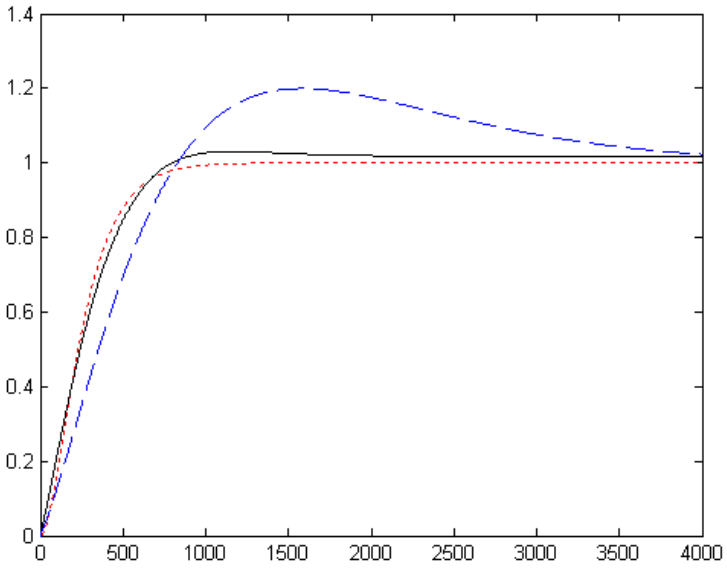



Fig. 5.13 Full line: final Joint 1 step response after 20 iterations
Dashed: initial response - Dotted: desired output

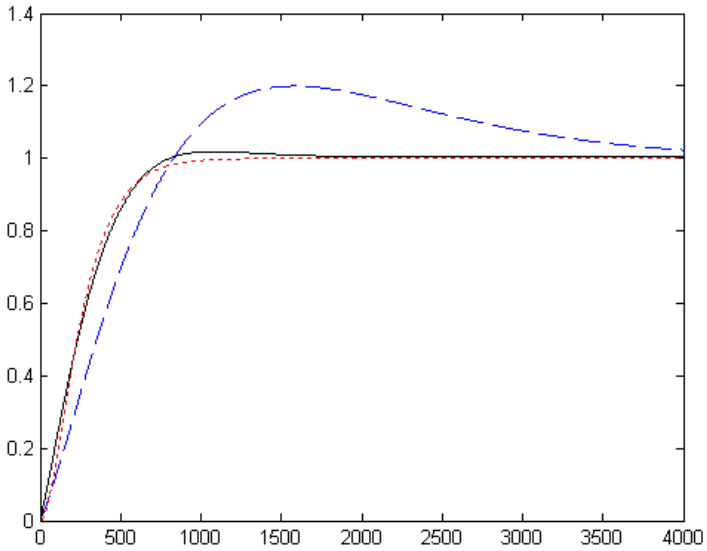



Fig. 5.14 Full line: final Joint 1 step response after 30 iterations
Dashed: initial response - Dotted: desired output

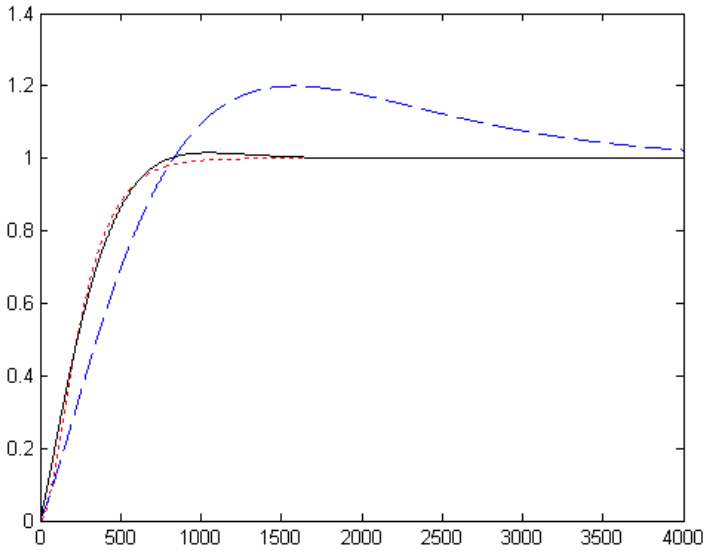



Fig. 5.15 Full line: final Joint 1 step response after 40 iterations
Dashed: initial response - Dotted: desired output

We can plot the graph of the cost function $J$ respect to the IFT iterations and summarize the values in the following table.

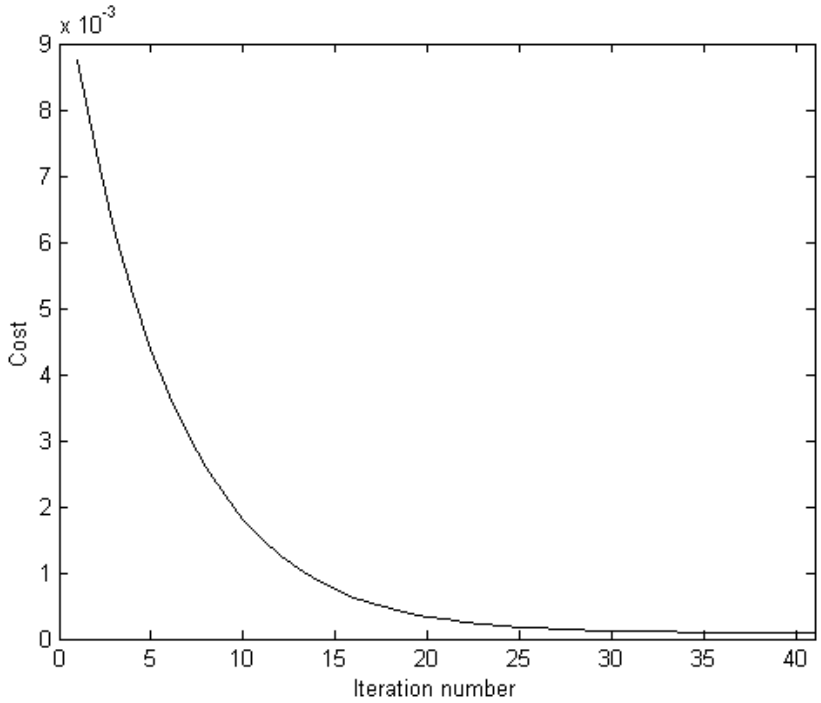



Fig. 5.16 Criterion as function of the number of IFT iterations

| **Criterion values** |
|---|
| $J_{0(initial)} = 0.008744$ |
| $J_{10} = 0.0015060$ |
| $J_{20} = 0.0002908$ |
| $J_{30} = 0.0001162$ |
| $J_{40(final)} = 0.00009158$ |

Fig. 5.17

The final values of the PID parameters are:

$$k_{p\,40} = 0.1421$$
$$T_{i\,40} = 5109.14$$
$$T_{d\,40} = 0.5651$$

We can follow the parameter update during the IFT algorithm looking at the following graphs:

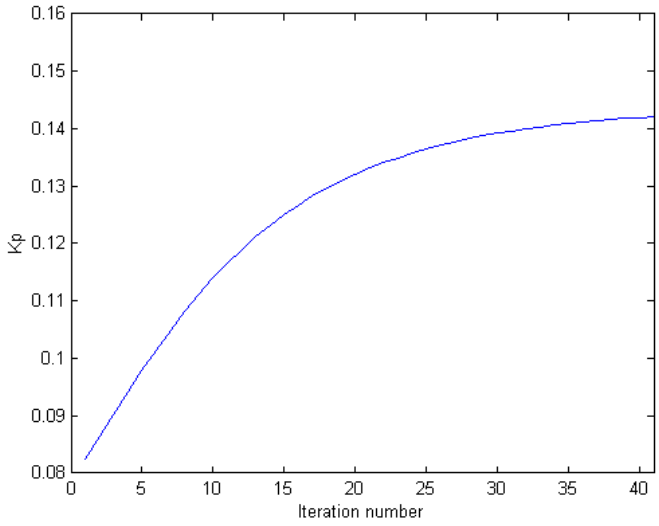



Fig. 5.18

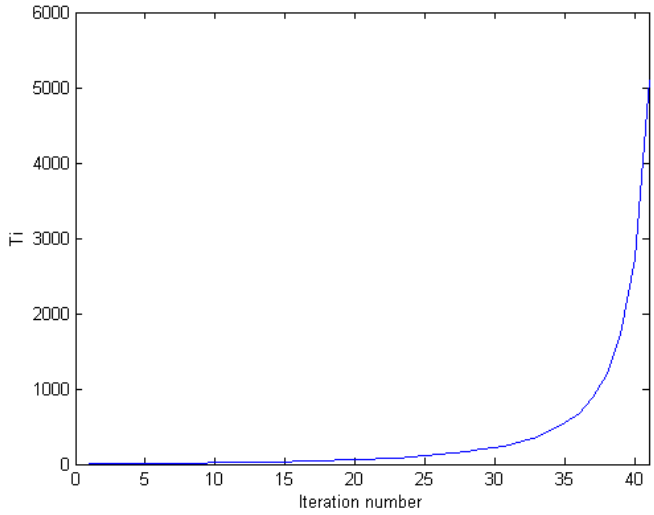



Fig. 5.19

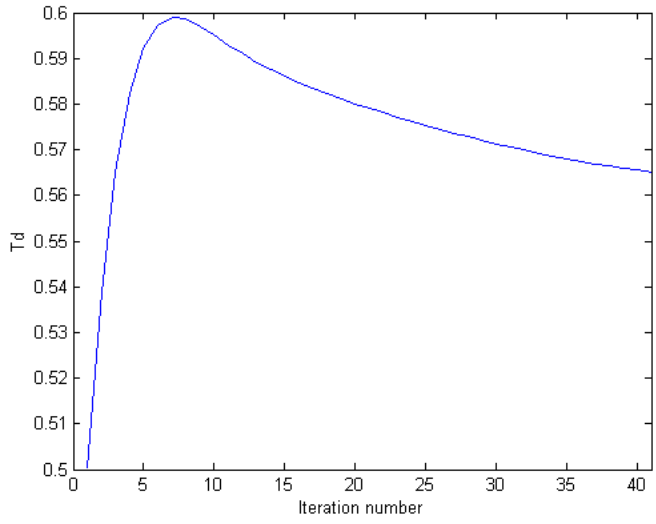



Fig. 5.20

## 5.6 APPLICATION OF IFT ON THE REAL ROBOT PROCESS

### 5.6.1 THE JOINT 1 EXPERIMENT

We implement a PID controller in the Simulink robot interface as the following (the standard controller is deactivated and the control signal is directly on the torque).

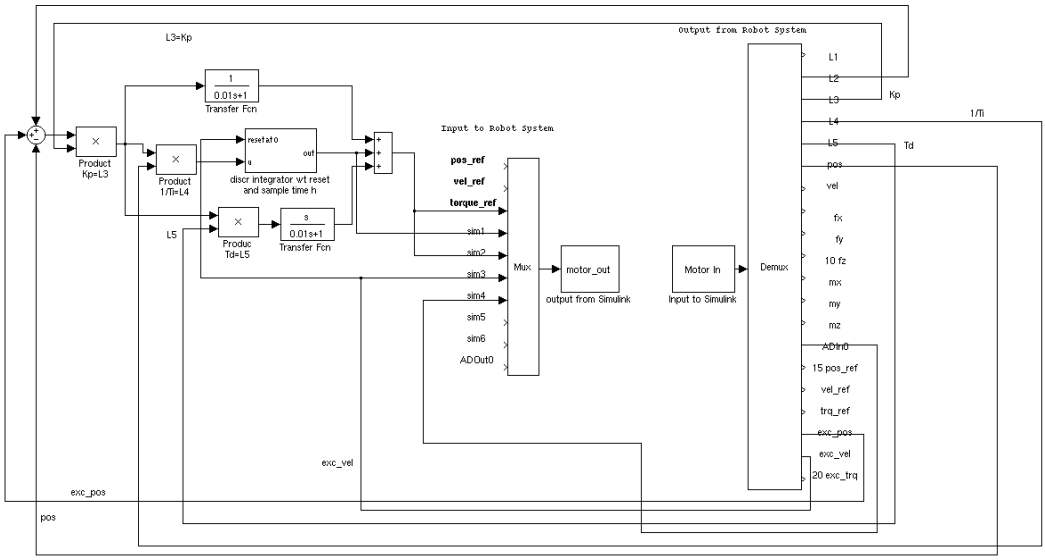



Fig. 5.21

The purpose is to improve the response of the robot joint 1.

The reference uses is $y_{ref} = \left( \frac{4/a}{s + 4/a} \right) r$ , with $a$ = 0.25, $r$ unit step.

We start with the initial parameters:

$K_p = 0.4$

$T_i = 1$

$T_d = 0.1$

We use the criterion $J(\rho) = \frac{1}{2N} \sum_{t=1}^{N} (y_t - y^d)^2$ and

- desired output as the reference signal;
- a Gauss-Newton approximation of the Hessian for the matrix $R_i$;
- a step size $\gamma = 0.1$.

We performed three different IFT schemes: standard (no weights), modified with double weighting and modified with variable mask.

The results are shown in the following sections.

**Standard IFT – no weights**

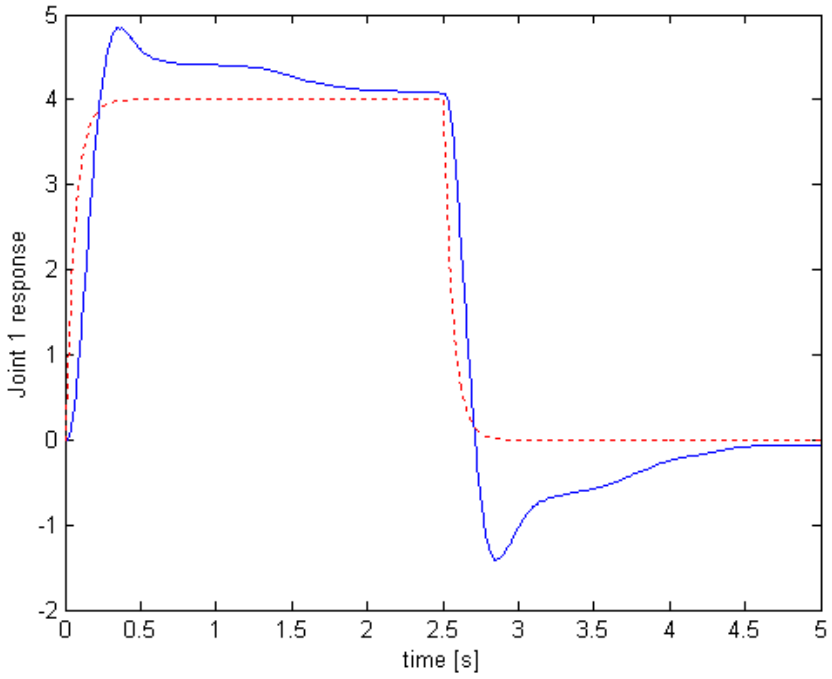



Fig. 5.22 Full: initial response - Dotted: desired output

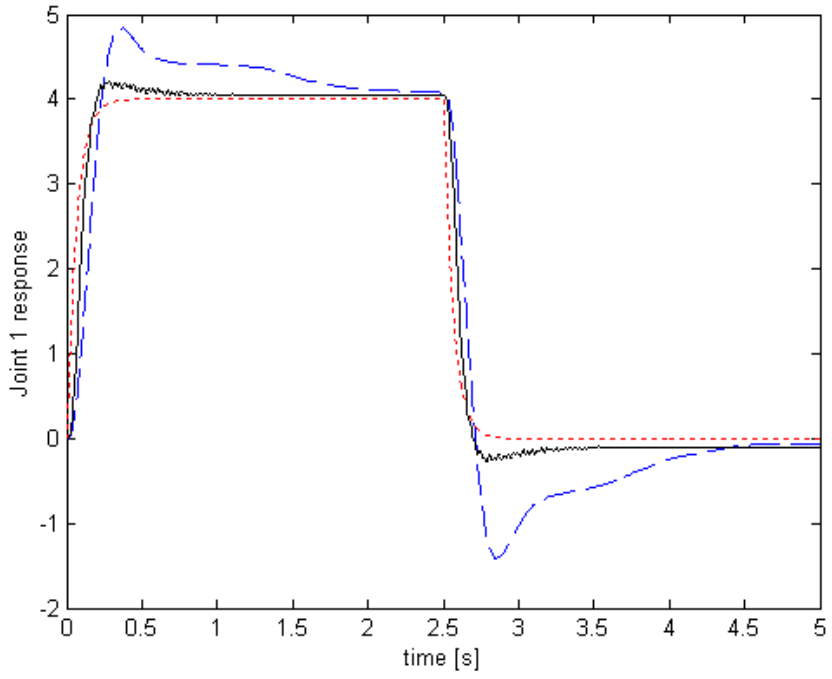



Fig. 5.23 Full: final response after 10 iterations – Dashed: initial – Dotted: desired

We can notice how the settling time is not very satisfactory.

In the following pictures the parameter updates are shown.

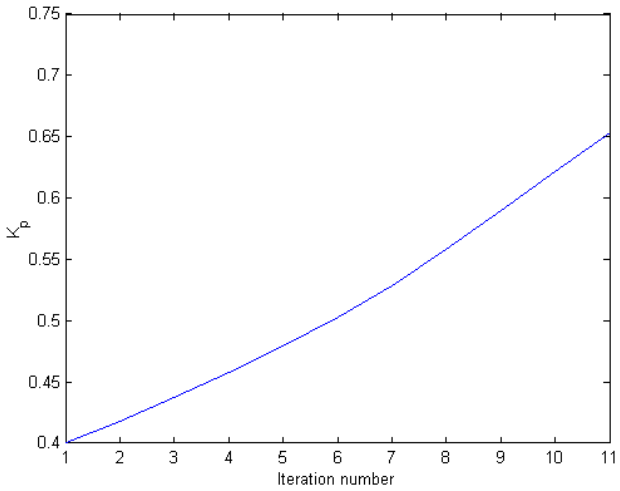



Fig. 5.24

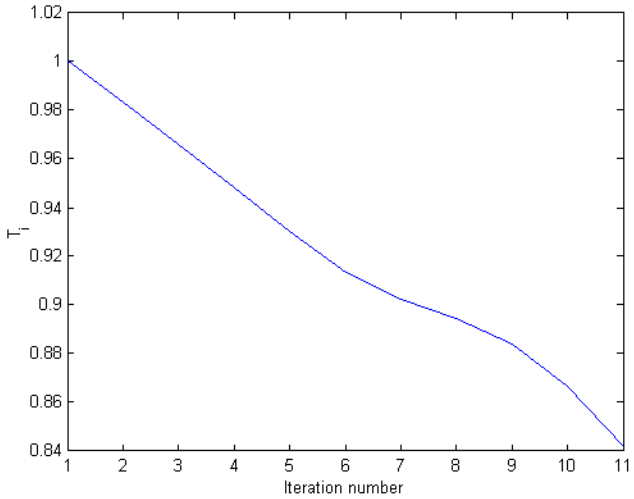



Fig. 5.25

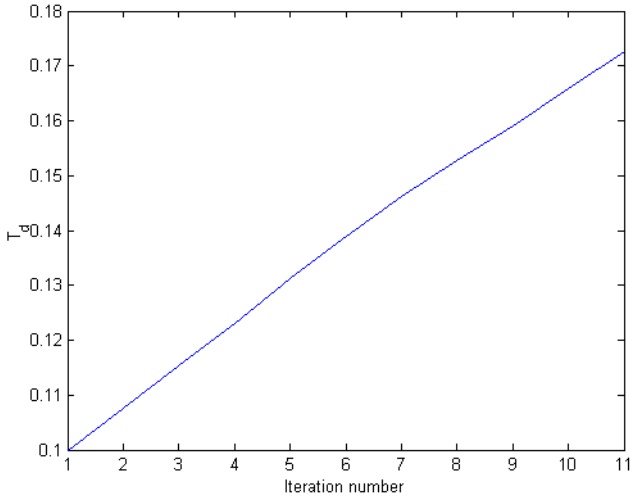



Fig. 5.26

**Modified IFT – fixed weighting**

We use now a modified criterion using time weighting as shown in the first picture.

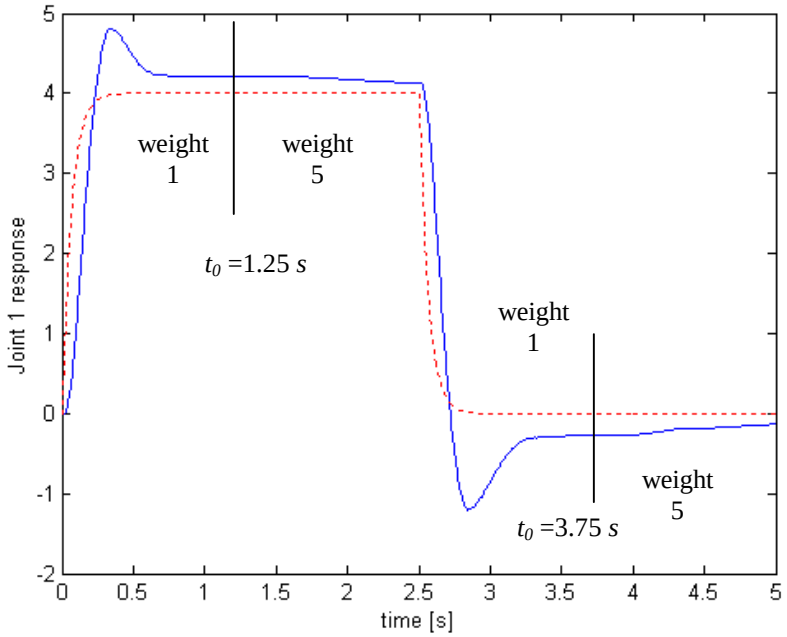



Fig. 5.27 Full: initial response - Dotted: desired output

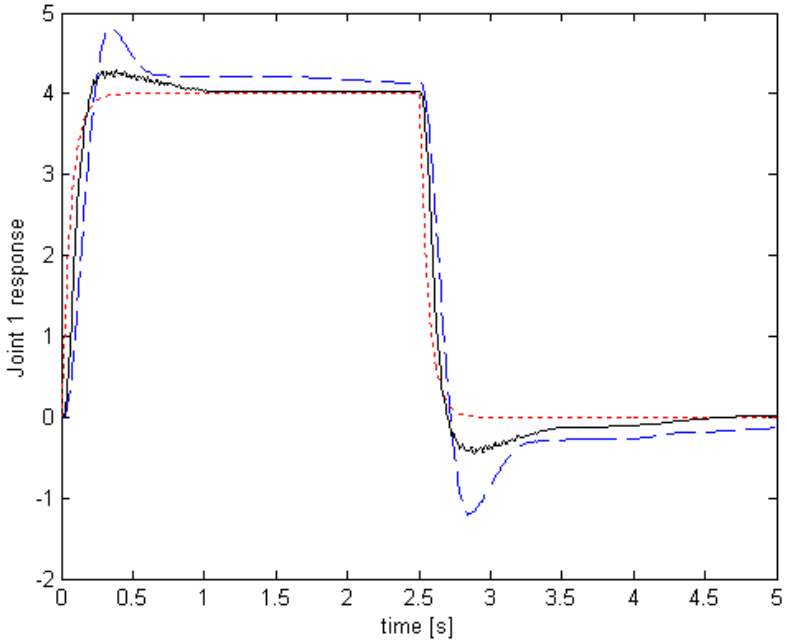



Fig. 5.28 Full: final response after 8 iterations – Dashed: initial – Dotted: desired

We can notice how this technique yields to a better response in terms of settling time. However we introduce oscillations during the overshoot and the result from iteration number 9 became critical (stability problem).

The parameter updates are shown in the pictures below.

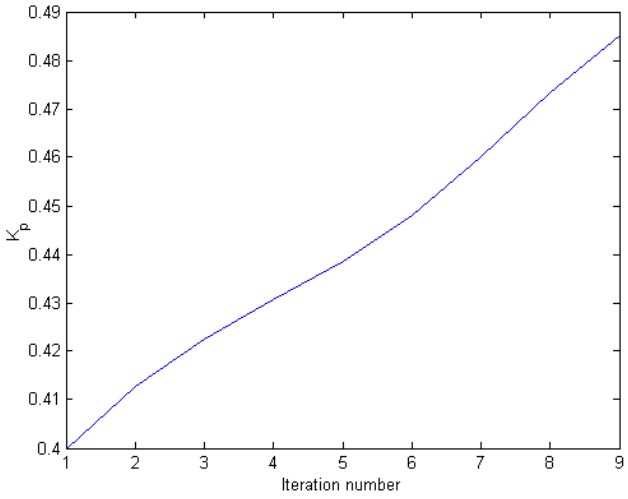



Fig. 5.29

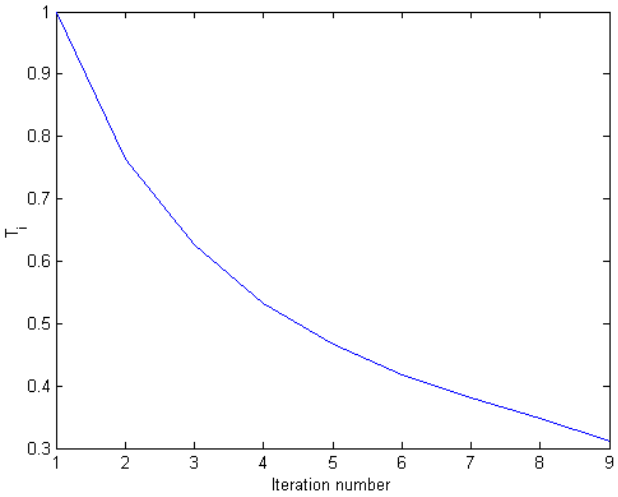



Fig. 5.30

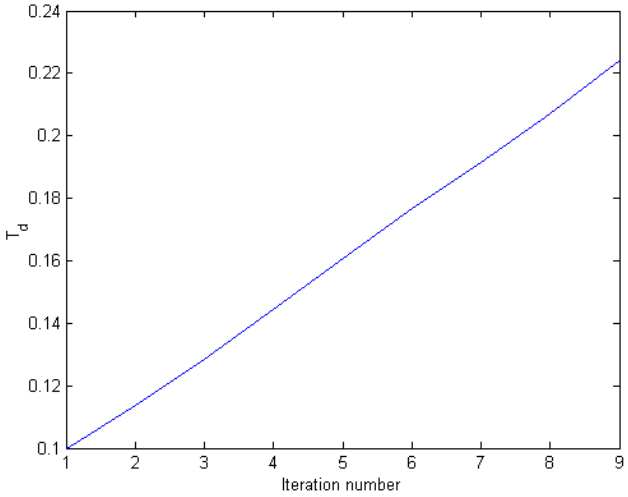



Fig. 5.31

**Modified IFT – Weighting with a variable mask**

A mask with initial lengt of 1.5 seconds is used as shown in the figure below and at every iteration it is decreased by 0.15 seconds.

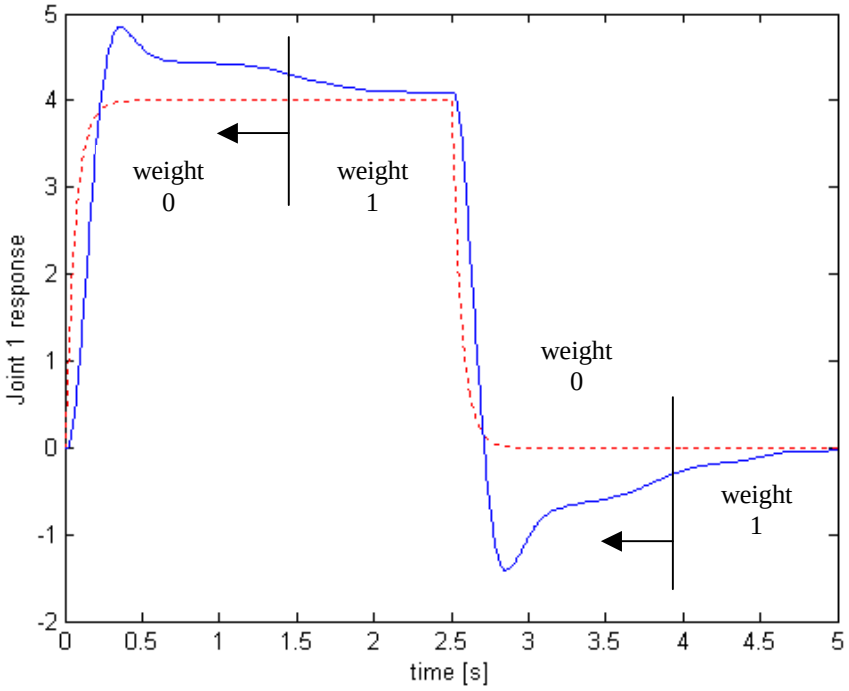



Fig. 5.32 Full: initial response - Dotted: desired output

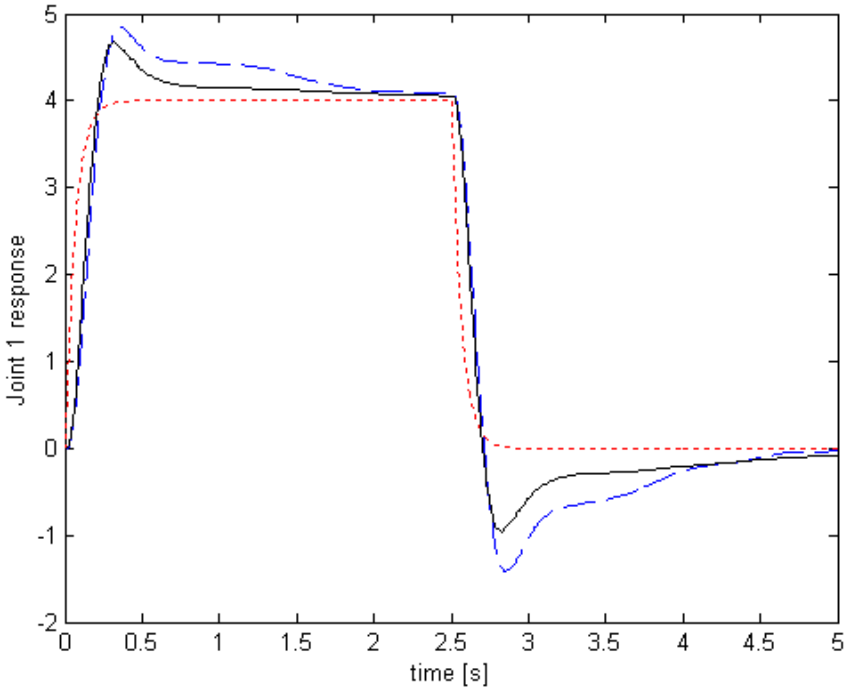



Fig. 5.33 Full: final response after 10 iterations – Dashed: initial – Dotted: desired

The parameter updates are shown in the picture below.

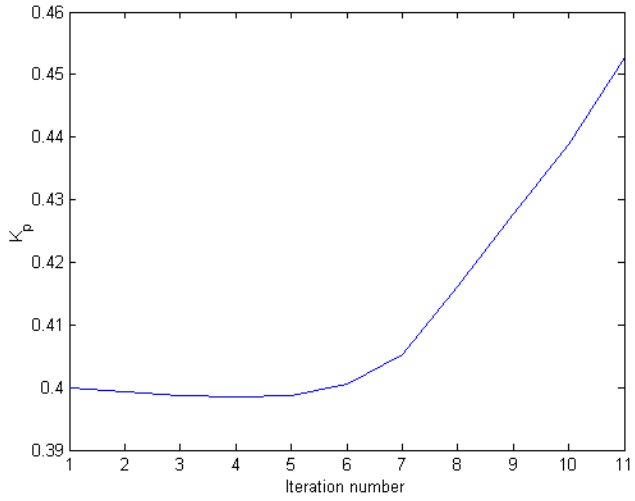



Fig. 5.34

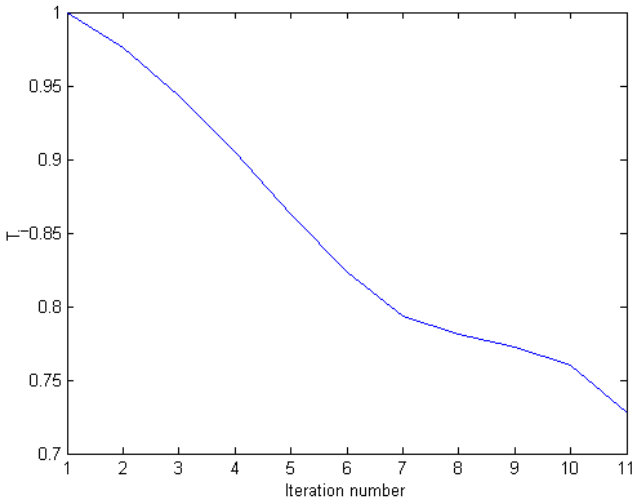



Fig. 5.35

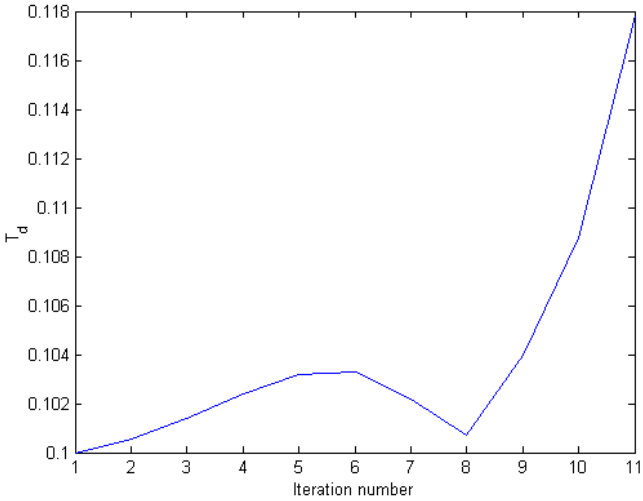



Fig. 5.36

### 5.6.2 THE FLEXIBLE BEAM EXPERIMENT

The purpose is to design a controller for the beam deflection and to improve the behaviour of the system using the IFT algorithm.
We set up the robot with a flexible beam mounted on the joint 6 of the robot (see figure below).

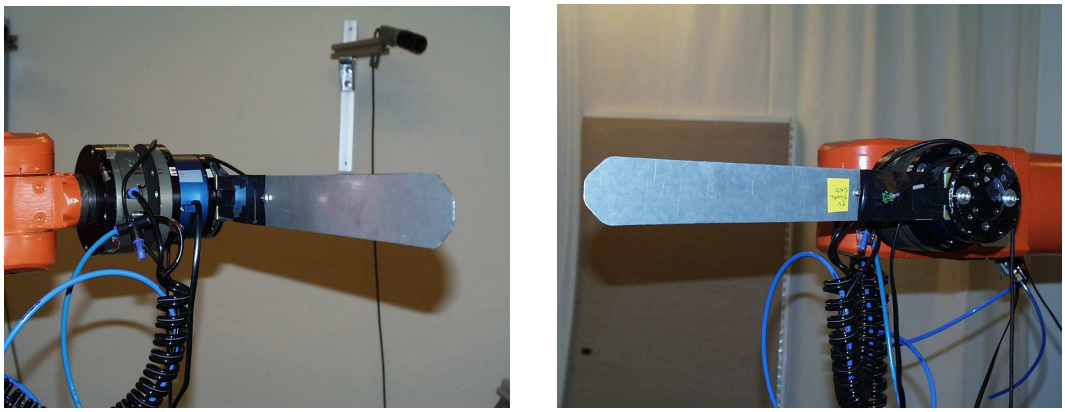


Fig. 5.37 The flexible beam

We have used a force/torque sensor (JR3) for measuring the beam deflection (torque approximately proportional to the deflection).
The beam was also equipped with a strain gauge which also could be used for estimating the beam deflection but in these experiments we only used torque measurements.

The systems is strongly related with the two-mass-process seen in the previous chapter, where we could consider the first mass as the robot mass and the second mass as the beam mass.
The process could be compared with the picture below, where $m_1 >> m_2$.

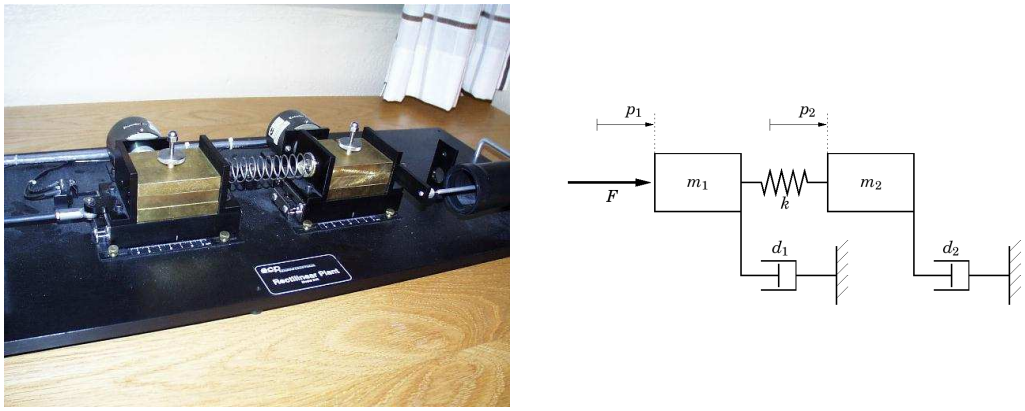



Fig. 5.38 The two-mass-process

## Identification Experiment

The first step is to identify a SIMO model (Single Input Multiple Output) from the position reference(signal $pos_{ref}$) to the robot position and beam deflection(signal $m_y$) which is to be used for control of the beam deflection.
In these experiments we have the standard position controller activated.

### Step response experiment

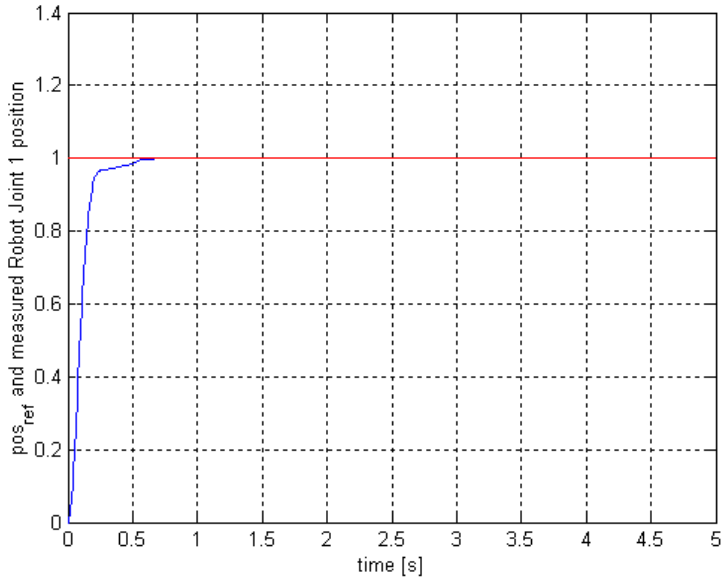



Fig. 5.39

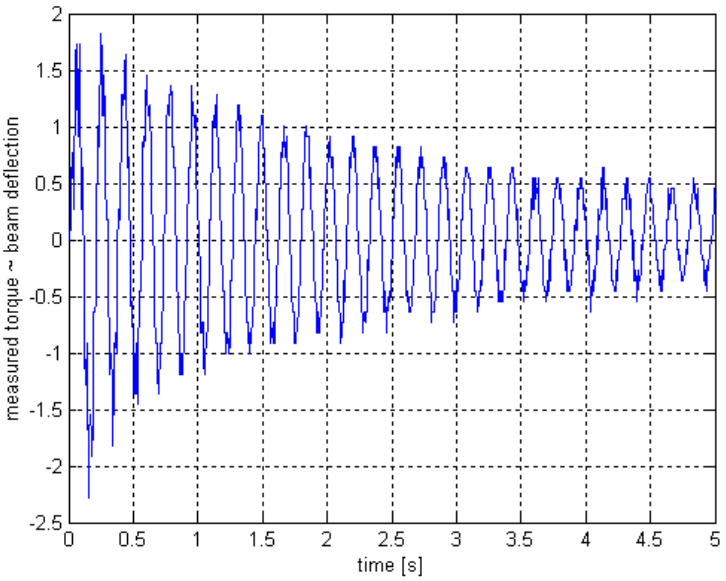



Fig. 5.40

We see that the position control of the robot joint is satisfactory but there are large poorly damped oscillations for the flexible beam (~6 Hz).

**Frequency response experiment**

To estimate a good model for the control we use a Pseudo Random Binary Sequence (PRBS) as excitation signal.

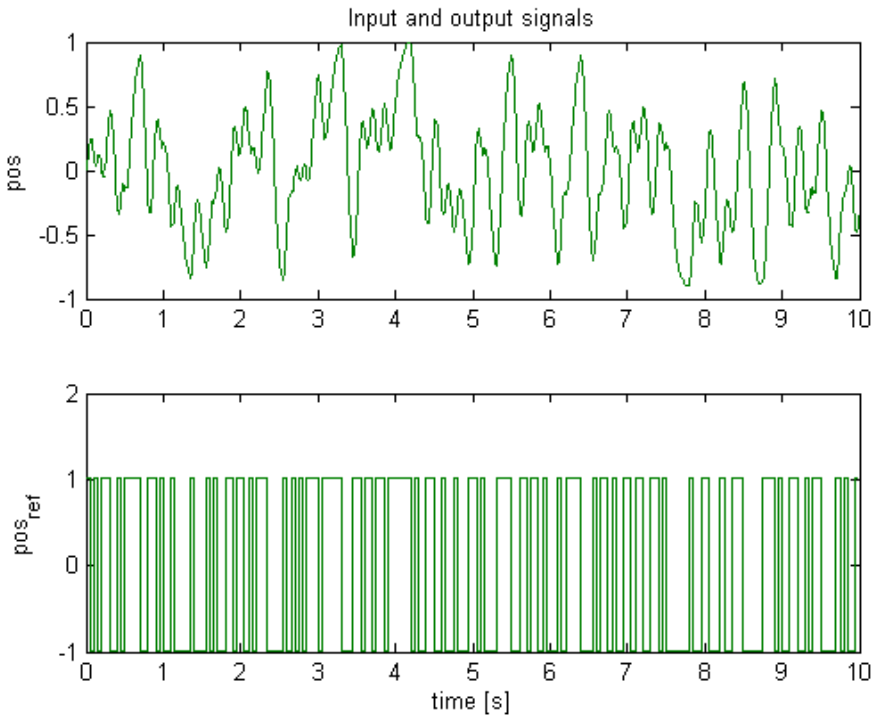



Fig. 5.41 Response of the joint 1 position with a PRBS as reference

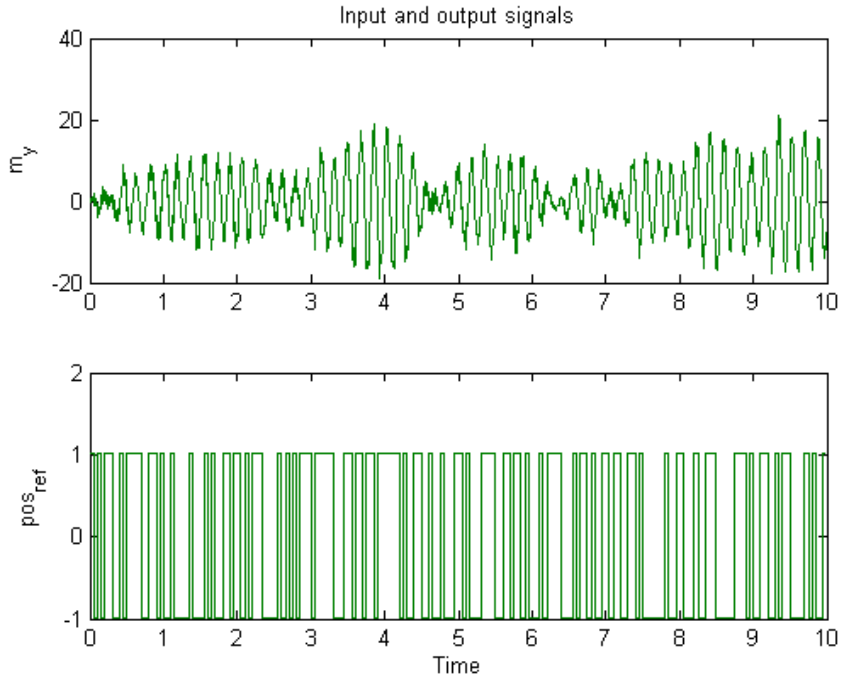



Fig. 5.42 Response of the flexible beam deviation (signal $m_y$) with a PRBS as reference

Using a sub-space estimation method (N4SID) from System Identification Toolbox (Matlab), we get good results for a state space model of order 6: loss function 0.00203914, FPE 0.00214452.

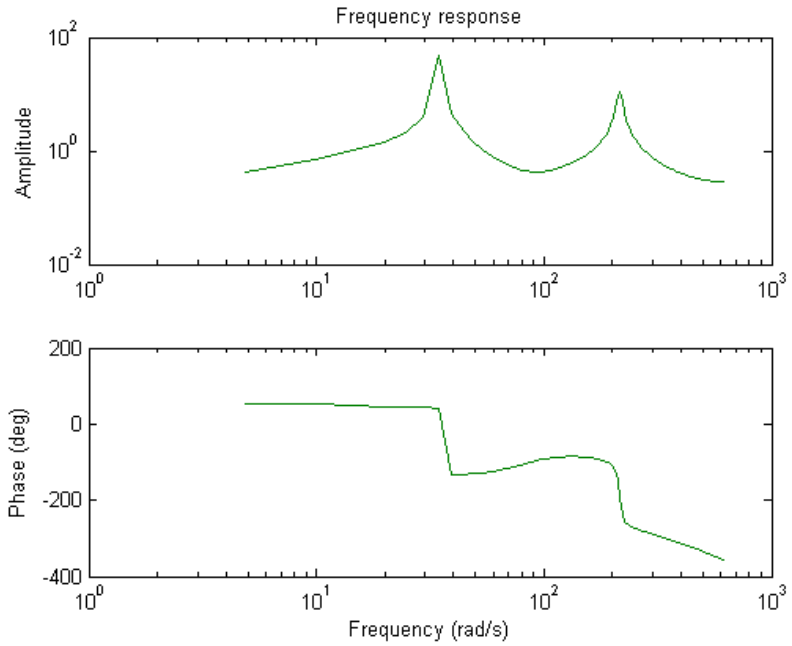



Fig. 5.43 Frequency response of the 6 order model found with the N4SID method (pos_ref to $m_y$)

We try to reduce to order 4 to capture essential dynamics but, with the original data sequence we have a problem of matching the correct first resonancy frequency and we have a lower peak for the gain. We get the following values: loss function 0.193784, FPE 0.200404.

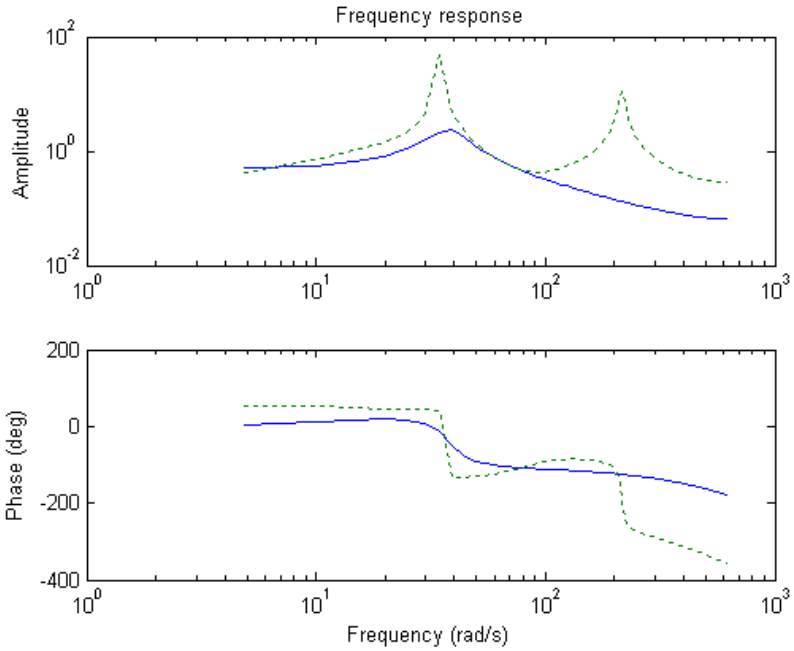



Fig. 5.44 Frequency response of the 4 order model (full line) found with the N4SID method ($pos_{ref}$ to $m_y$) and of the 6 order model (dotted)

Finally, using a low pass filter on the signals before doing the identification in order to match the first resonance frequency, we get, implementing again the N4SID method, a state space model of order 4 with loss function 1.00914e-06 and FPE 1.04361e-06 and the following frequency response.

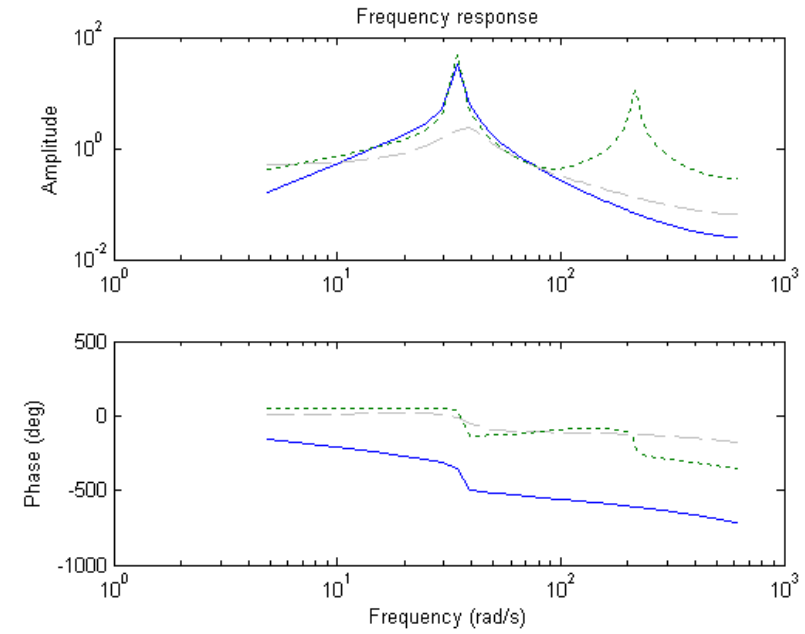



Fig. 5.45 Frequency response of the 4 order model (full line) found with the N4SID method ($pos_{ref}$ to $m_y$) using filtered data
Dotted: previous 6 order model – Dashed: previous 4 order model (no pre-filtering)

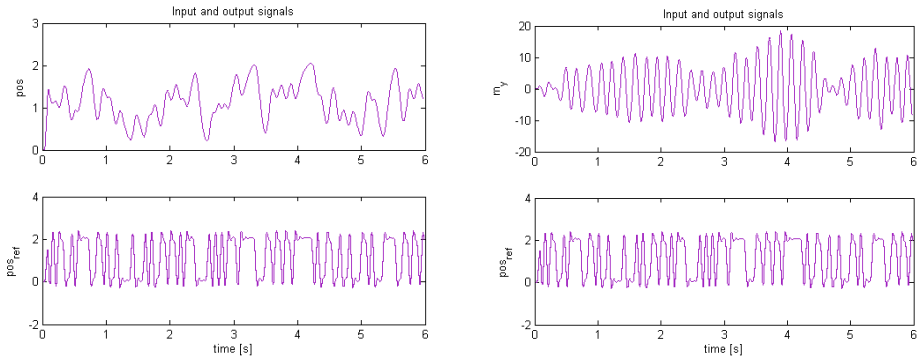



Fig. 5.46 Filtered version of the data set used for the identification of the fourth-order model with prefiltering

The final discrete state-space model identified (order 4, prefiltering) and chosen for the control implementation is:

$$x(t+t_s) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

with the sampling time $t_s = 0.005$ and the matrices:

$$A = \begin{bmatrix} 0.97659 & 0.00041152 & -0.033363 & 0.067988 \\ -0.027119 & 0.97038 & 0.008932 & -0.15777 \\ 0.14484 & 0.087583 & 0.88605 & -0.014171 \\ -0.04572 & 0.17648 & -0.025004 & 0.98178 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.00053439 \\ -0.00020808 \\ 0.012453 \\ 0.0030635 \end{bmatrix}$$

$$C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} -18.237 & -7.813 & 0.13774 & -0.0019718 \\ 108.12 & -209.04 & -3.986 & 13.831 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The poles of the identified model are:

$$p_1 = 0.9257 + j0.0547$$
$$p_2 = 0.9257 - j0.0547$$
$$p_3 = 0.9817 + j0.1736$$
$$p_4 = 0.9817 - j0.1736$$

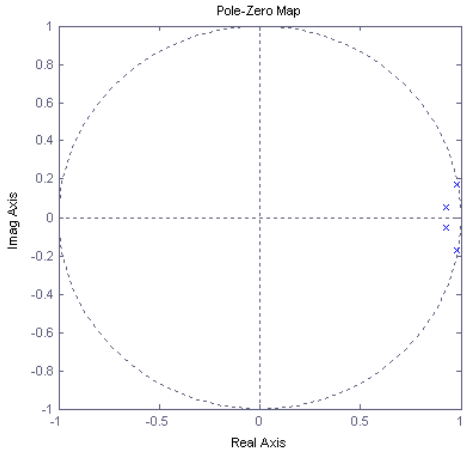



Fig. 5.47

# CONTROL DESIGN

We choose a state feedback control structure with the purpose to damp the flexible beam deflection $m_y$ and still get a good (fast) step response.

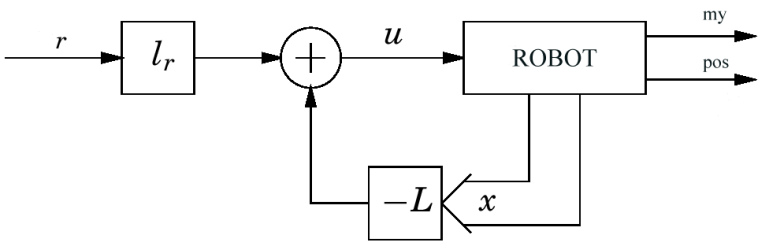



Fig. 5.48

**Pole placement**

For the closed-loop system we choose to keep the speed and the damping of the two poles corresponding to the position and *move* the two poorly damped poles related to the flexible beam keeping the same eigenfrequency but increasing the damping.
From the identified discrete time model the poles were transfered to the corresponding continuos time poles were we have an easy interpretation of the damping.

$$z^2 + a_1 z + a_2 = 0 \qquad \rightarrow \qquad s^2 + 2\omega\zeta s + \omega^2 = 0$$

Poles are mapped according to $z_i = e^{pt_s}$ ($p$ continuous time pole, $z_i$ discrete time pole) [Åström & Wittenmark].
In our case we increased the damping of the complex poles from approx 0.02 to 0.74
We make a pole placement design with the new discrete time poles.

$p_1 = 0.9296 - j0.06$
$p_2 = 0.9296 + j0.06$
$p_3 = 0.8724 - j0.1032$
$p_4 = 0.8724 + j0.1032$

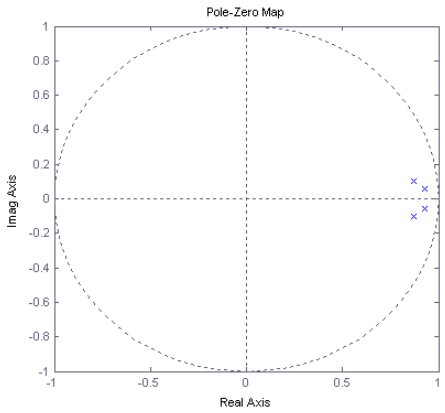



Fig. 5.49

As only the position and torque ($m_y$) are measured from the forth (or sixth order model) we also design an observer for the process from which we use the feedback.

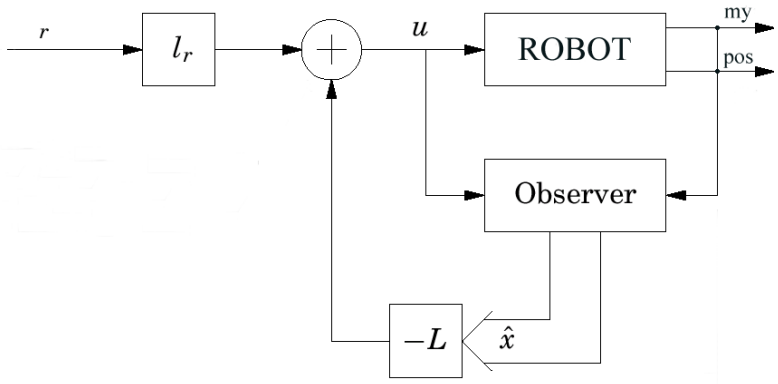



Fig. 5.50

The observer poles were chosen to be approximately 1.5 times faster than the desired closed loop poles for the system. Even though the eigenvalues of (*A-BL*) and (*A-KC*) which will be the closed loop system eigenvalues, were to be stable we also want to consider the poles of the resulting controller, namely the eigenvalues of (*A-BL-KC*), to be stable.
In the experiments when we will compare between uncontrolled and controlled system it is convenient to be able to switch on and off the use of the controller without troublesome transients. We have therefore chosen the poles so that also the transfer function of the controller is stable.
To have a stable controller may also be a safety precaution if there for instance should be a sensor failure.

The state space design using state feedback and observer gives:

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K([y_{pos}(k) \quad y_{m_y}(k)] - [\hat{y}_{pos}(k) \quad \hat{y}_{m_y}(k)])$$
$$[\hat{y}_{pos}(k) \quad \hat{y}_{m_y}(k)] = C\hat{x}(k)$$
$$u(k) = -L\hat{x}(k)$$

with

$$L = \begin{bmatrix} 73.7156 & -87.6223 & -15.0262 & 136.7882 \end{bmatrix}$$

$$K = \begin{bmatrix} -0.0089 & 0.0004 \\ 0.0007 & -0.0015 \\ 0.0124 & -0.0007 \\ 0.0029 & 0.0006 \end{bmatrix}$$

We implement the state feedback controller on the robot process using the Simulink control interface with the robot shown in below.

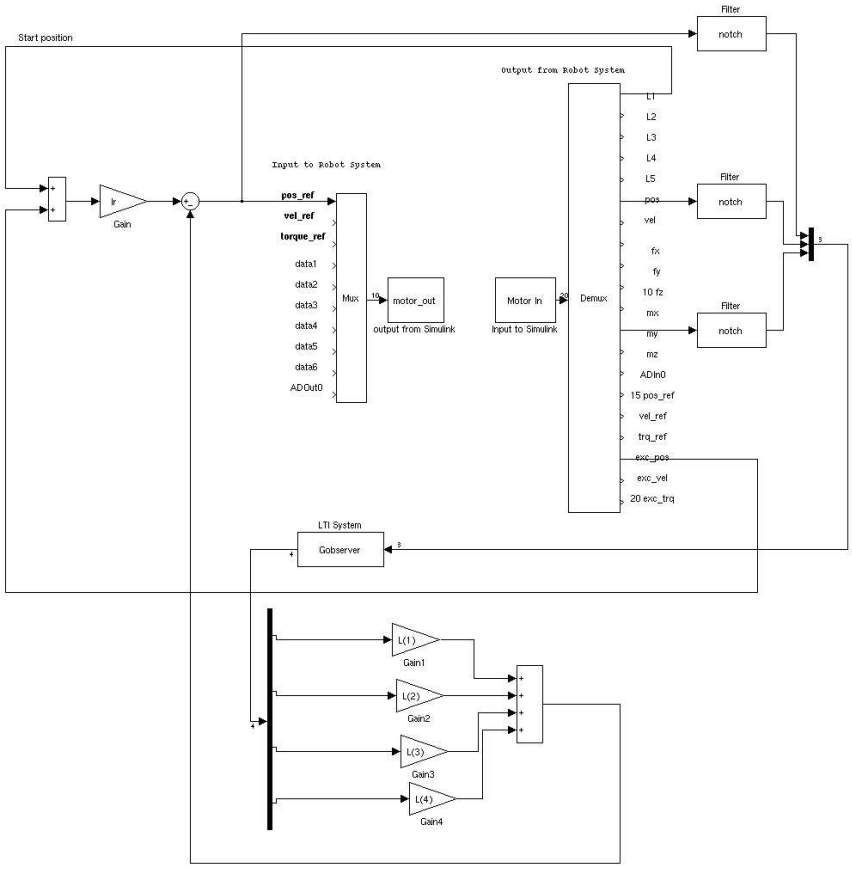



Fig. 5.51

Notice the notch filters used in the measurement of the position, beam deflection and input signal to avoid problems due to the higher resonancy frequency that is not included in the 4-th order model.

$$F_{notch}(z) = \frac{z^2 - 1.29z + 0.913}{z^2 - 1.221z + 0.844}$$

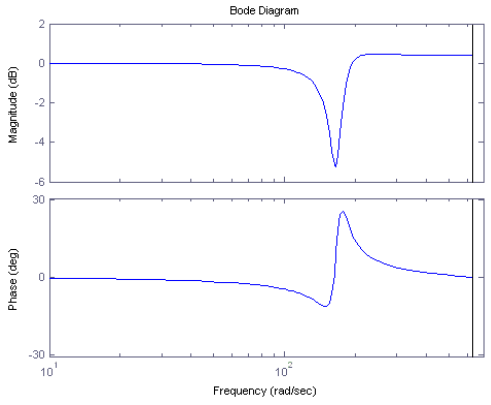



Fig. 5.52

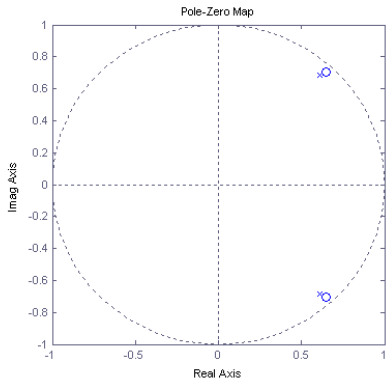



Fig.5.53

Notice also that in this case we keep the original robot joint 1 controller and in fact we enter in the *pos_ref* channel of the robot system input interface. So we use a state feedback controller to determine the new position reference to pass to the original robot controller.

The robot joint 1 position step response and the beam deflection are:

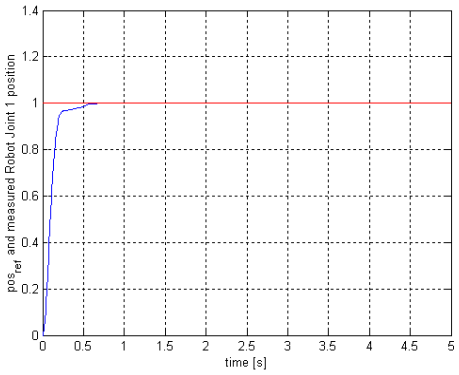



Fig. 5.54 Pos before statefeedback

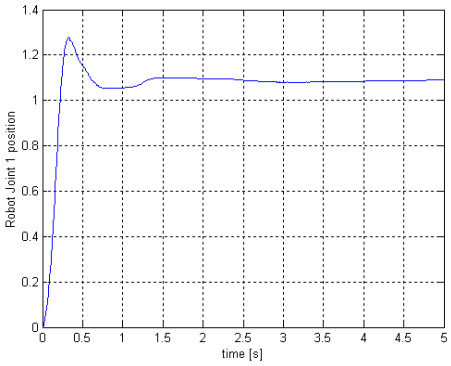



Fig. 5.55 Pos after statefeedback

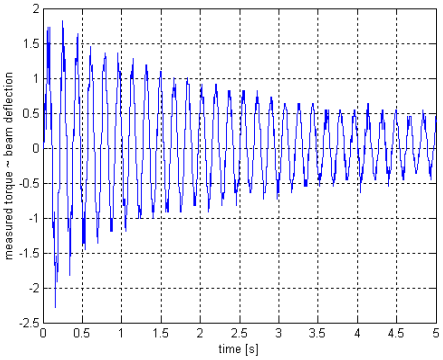



Fig. 5.56 Signal $m_y$ before statefeedback control

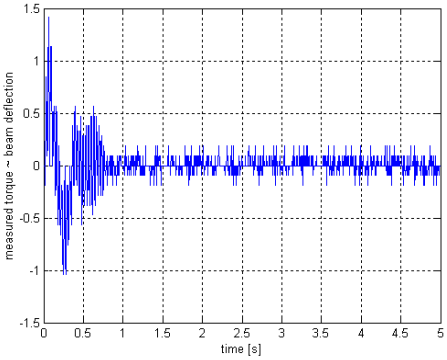



Fig. 5.57 Signal $m_y$ after statefeedback control

As we can see we get good improvements on the beam deflection but we introduce a worse behaviour in the joint 1 postion response.

In the next step, the idea is to use IFT to improve the joint 1 response still keeping a good damping of the flexible beam.

## FIRST STRATEGY: IFT and the two-gains method

We consider the controlled robot/beam system and we introduce a two-gains control modification. The role of IFT will be to tune these two gains such that we get improvements on the joint 1 step response.

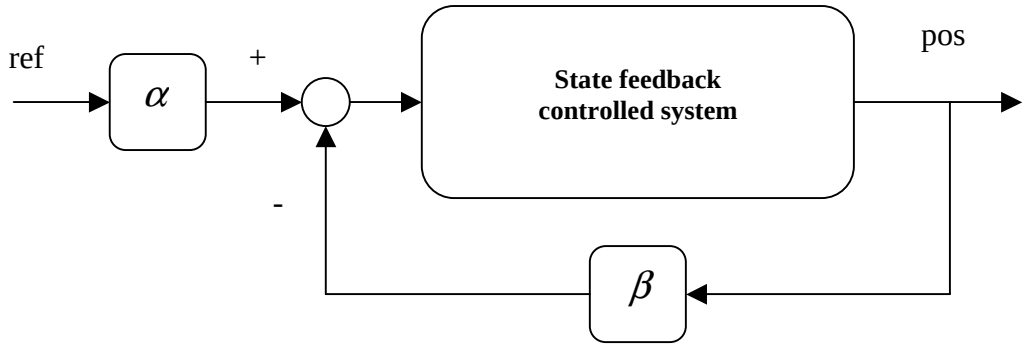



Fig. 5.58

The Simulink control interface is shown in the figure below.

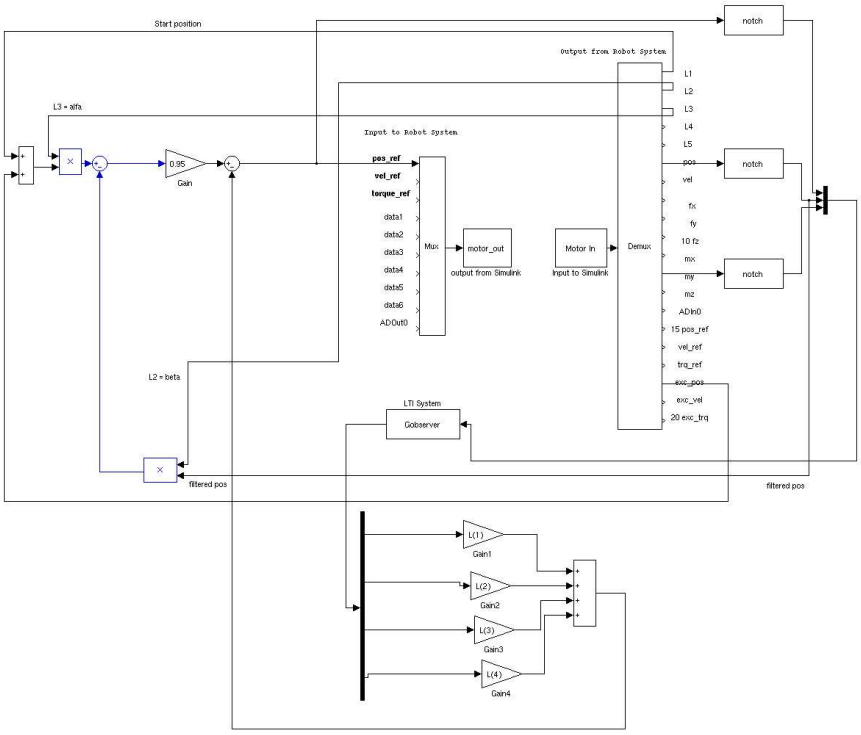



Fig. 5.59

Starting, as it is natural, with initial gains $\alpha = 1, \beta = 0$ and using in the classical IFT algorithm:

- a cost function $J(\rho) = \frac{1}{2N}\sum_{t=1}^{N}(y_{pos} - y^d)^2$
- a desired output $y^d$ as a unit step
- a Gauss-Newton approximation of the Hessian for the matrix $R_i$
- a step size $\gamma_i = 0.5\ \forall i$

we get, after 4 iterations, the following step response and corresponding flexible beam deflection.

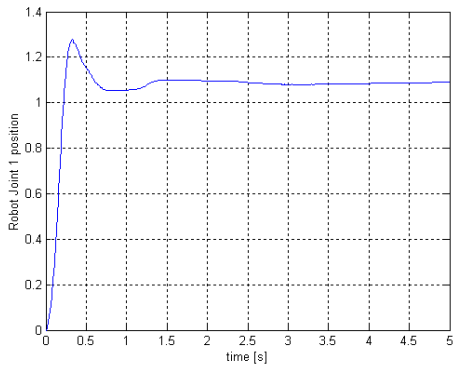



Fig. 5.60 Initial response

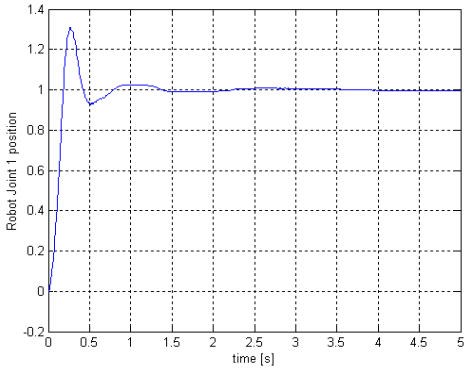



Fig. 5.61 After 5 iterations

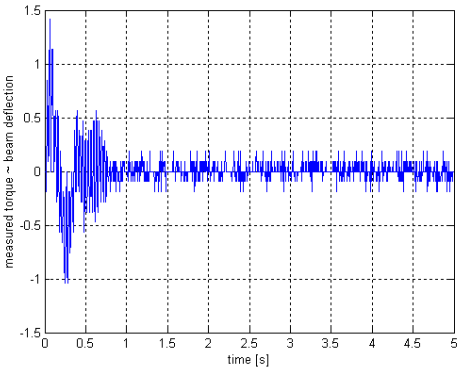



Fig. 5.62 Initial beam deflection

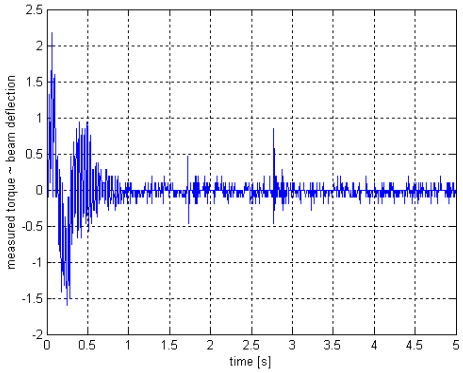



Fig. 5.63 After 5 iterations

The following figure show the cost function.

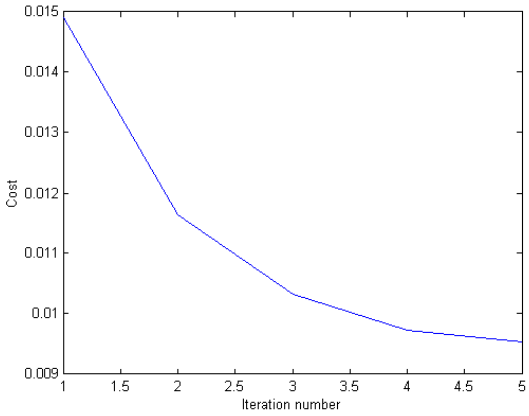



Fig. 5.64

The final parameters are:

$$\alpha = 1.3850$$

$$\beta = 0.4708$$

## SECOND STRATEGY: IFT and the three-gains method

We introduce a three-gains control modification as shown in the figure below, with $\tau = 0.01$.

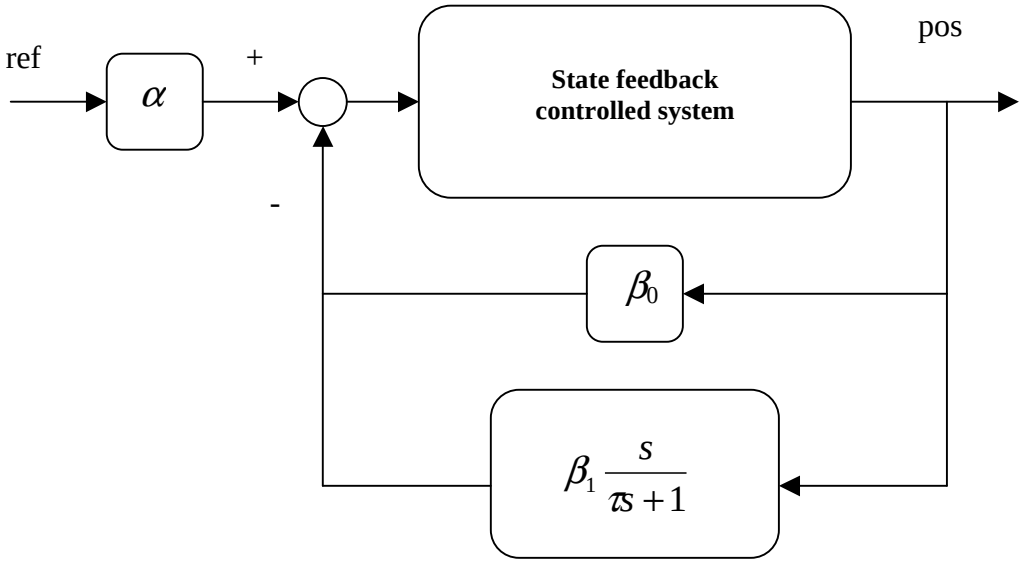



Fig. 5.65

The Simulink control interface is shown on the next page.

We start, as it is natural, with initial gains $\alpha = 1$, $\beta_0 = 0$, $\beta_1 = 0$ and we use in the classical IFT algorithm:

- a cost function $J(\rho) = \frac{1}{2N}\sum_{t=1}^{N}(y_{pos} - y^d)^2$
- a desired output $y^d$ as a unit step
- a Gauss-Newton approximation of the Hessian for the matrix $R_i$
- a step size $\gamma_i = 0.2\ \forall i$.

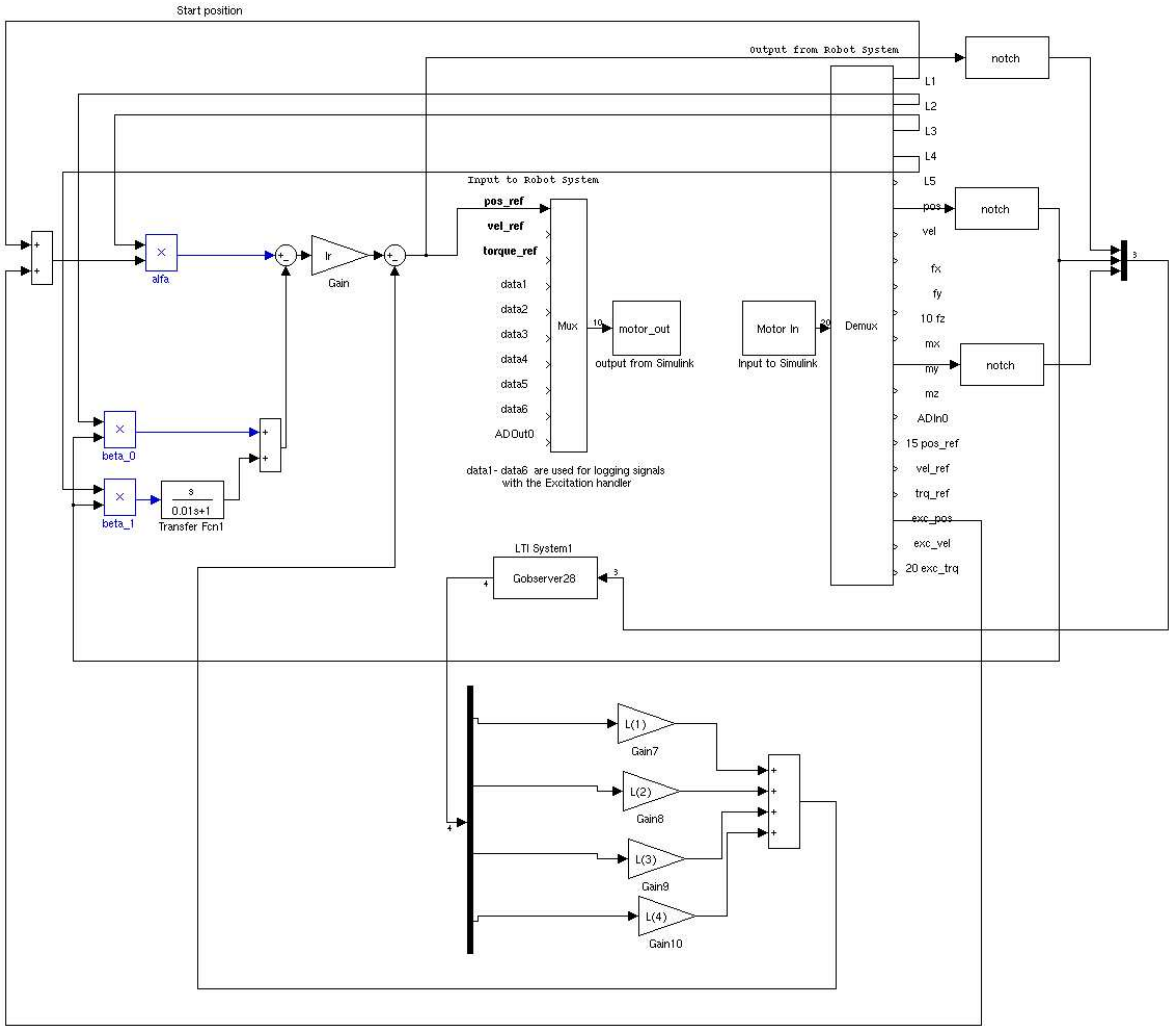



Fig. 5.66 The simulink interface with the robot used in the three-gains method

We get, after 5 iterations, the following step response and corresponding flexible beam deflection.

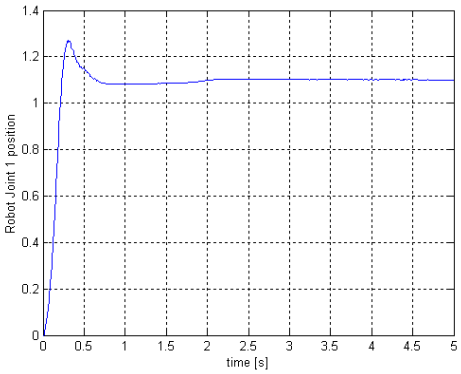



Fig. 5.67 Initial response

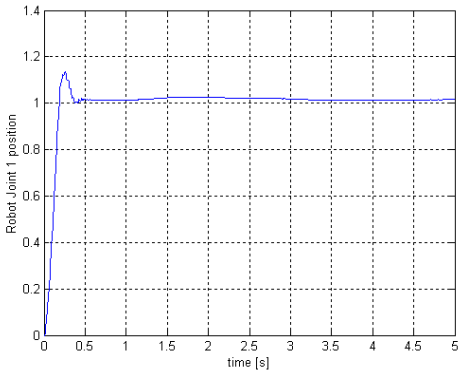



Fig. 5.68 After 5 iterations

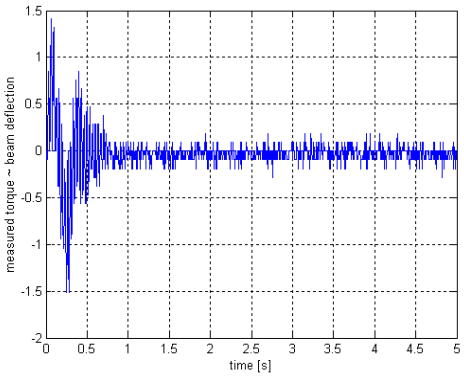



Fig. 5.69 Initial beam deflection

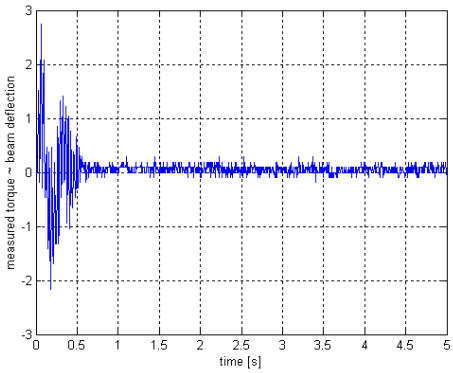



Fig. 5.70 After 5 iterations

The following figure show the cost function.

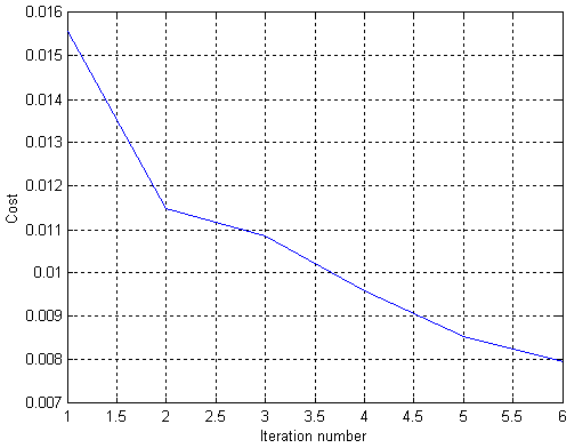



Fig. 5.71

The final parameters are:

$$\alpha = 1.6950$$
$$\beta_0 = 0.7437$$
$$\beta_1 = 0.0502$$

We can notice how the static error is reduced after some iterations but still an overshoot is present.

To reduce the overshoot we implement again the IFT, modifying the criterion introducing a time weighting as follow:

$$J(\rho) = \frac{1}{2N}\sum_{t=1}^{N}\left[w_y(t)(y_t - y^d)^2\right]$$

with the weights $w_y(t)$ shown in the graph below.

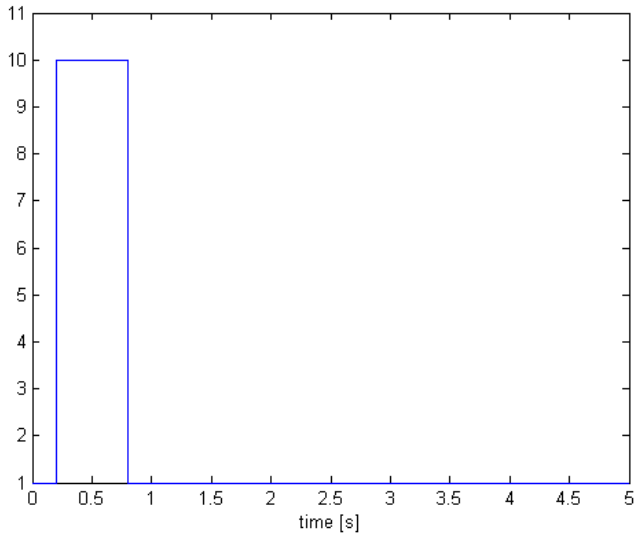



Fig. 5.72

In the IFT algorithm we use:

- a desired output $y^d$ as a unit step
- a Gauss-Newton approximation of the Hessian for the matrix $R_i$
- a step size $\gamma_i = 0.1\ \forall i$.

The initial parameters are:

$$\alpha = 1$$
$$\beta_0 = 0$$
$$\beta_1 = 0$$

Performing 4 iterations we get the following results.

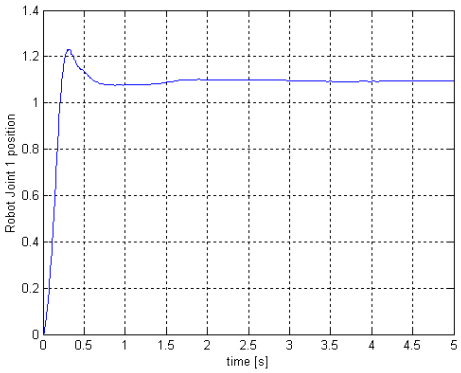



Fig. 5.73 Initial response

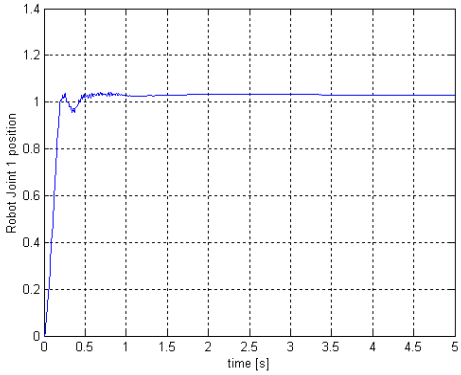



Fig. 5.74 After 4 iterations

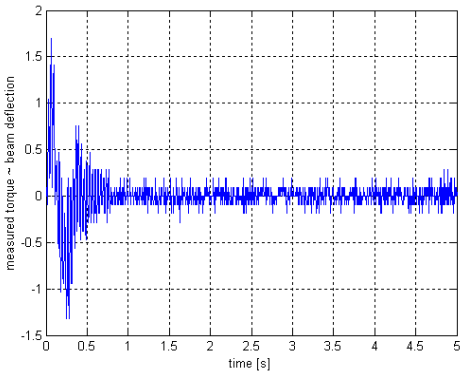



Fig. 5.75 Initial beam deflection

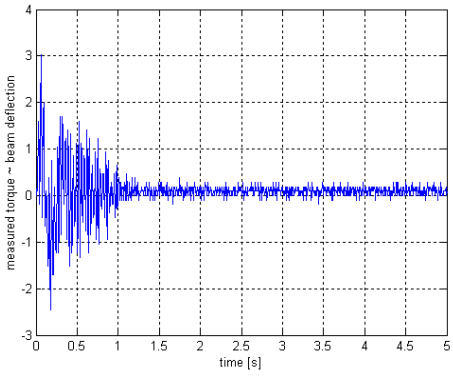



Fig. 5.76 After 4 iterations

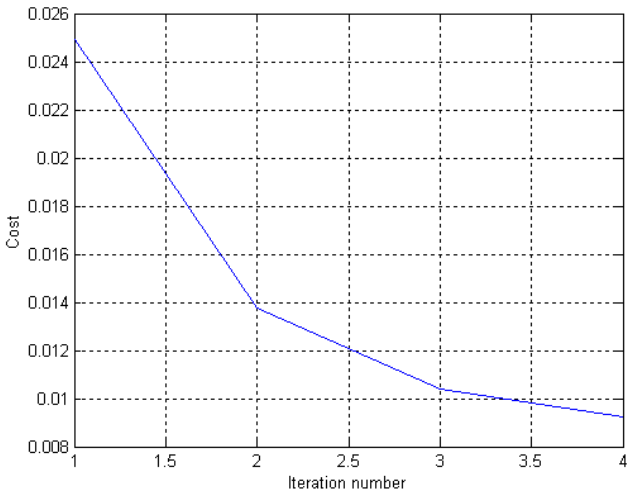



Fig. 5.77

The final parameters are:

$$\alpha = 1.9212$$
$$\beta_0 = 0.9436$$
$$\beta_1 = 0.1132$$

Using the modified criterion, even if the reduction of the static error through the iterations is slower, we can notice how the IFT improve the response in terms of overshoot.

# 6. Conclusions

In this thesis we have examined an optimization approach to iterative control design.

The important ingredient is that the gradient of the design criterion needed in the parameter update law is computed from measured closed loop data.
The approach is thus not model-based.

From a practical point of view, the scheme offers several advantages. It is straighforward to apply. It is possible to control the rate of change of the controller in each iteration. The objective can be manipulated between iterations in order to tighten or loosen performance requirements. Certain frequency regions can be emphasized if desired.

This direct optimal tuning algorithm is particularly well suited for the tuning of the basic control loops in the process industry, which are typically PID loops. These primary loops are often very badly tuned, making the application of more advanced techniques rather useless.

A first requirement in the successful application of advanced control techniques is that the primary loops be tuned properly. The IFT technique appears to be a very practical way of doing this, with an almost automatic procedure.

We showed also how IFT have high potential for the tuning of controllers applied to non-linear systems, even if attention has to be put in case of problems as high friction.

The results from the robot experiments show how in fact IFT can be used in combination with a previous controller. The experiments show that it is possible to improve the total behaviour using just few iterations of the algorithm.
The different IFT–schemes have been verified in simulations and in real experiments on an industrial robot manipulator ABB Irb-2000.
For the experiment with the flexible beam we have a case where we do not explicitly want to change the parameters of a present controller (combination of state-estimation and state-feedback). Instead we reformulate the controller system to a trivial feedback connection with two or three parameters which are used as initial parameters for the IFT algorithm.

# 7. Bibliography


De Bruyne F., B.D.O. Anderson, M. Gevers and N. Linard: "Iterative controller optimization for nonlinear systems", *Proceedings of 36th IEEE Conf. on Decision and Control*, San Diego, California (1997).

De Bruyne F. and P. Carrette: "Synthetic Generation of The Gradient for An Iterative Controller Optimization Method", *Proceedings of ECC* (1997).

Gevers M.: "A decade of progress in iterative process control design: from theory to practice", *Journal of Process Control*, 12 (2002).

Hamamoto K., T. Fukuda and T. Sugie: "Iterative feedback tuning of controllers for a two-mass-spring system with friction", *Control Engineering Practice*, 11 (2003).

Hamamoto K. and T. Sugie: "Contruction of suboptimal controllers via iterative feedback tuning", *CD-ROM of the fifth European Control Conference*, Karlsruhe (1999).

Hjalmarsson H.: "Control of nonlinear systems using iterative feedback tuning", *Proceedings of the American Control Conference*, Philadelphia, Pennsylvania, June 1998.

Hjalmarsson H.: "Iterative feedback tuning – an overview", *International Journal of Adaptive Control and Signal Processing*, 16 (2002).

Hjalmarsson H.: "Performance analysis of iterative feedback tuning", submitted to *Automatica* (1998).

Hjalmarsson H. and T. Birkeland: "Iterative feedback tuning of linear time-invariant mimo systems", *37th IEEE conference on decision and control*, Tampa (1998).

Hjalmarsson, H., M. Gevers, S. Gunnarsson and O. Lequin: "Iterative feedback tuning: theory and applications", *IEEE Control Systems Magazine*, 18 (1998).

Hjalmarsson H., S. Gunnarsson and M. Gevers: "A Convergent Iterative Restricted Complexity Control Design Scheme", *Proceedings of the 33rd CDC* (1994).

Kammer, L. C., F. De Bruyne and R. R. Bitmead: "Iterative Feedback Tuning via Minimization of the Absolute Error", *Proceedings of the 38th Conference on Decision & Control*. Phoenix, Arizona USA, December 1999.

Lequin O., M. Gevers and L. Triest: “Optimizing the settling time with Iterative Feedback Tuning”, *Proceedings of the 14th IFAC World Congress* (1999).

Lequin O., M. Gevers, M. Mossberg, E. Bosmans and L. Triest: “Iterative feedback tuning of PID parameters: comparison with classical tuning rules”, *Control Engineering Practice*, 11 (2003).

Scalamogna D.: ”Iterative Learning Control with Application to Robotics”, Master thesis, Department of Automatic Control, Lund Institute of Technology, June 2001.

Sjöberg J. and M. Agarval: “Nonlinear controller tuning based on linearized time-variant model”, *Proc. 1997 American Control Conference*, Albuquerque, New Mexico (1997).

Åström K. J. and T. Hägglund: “Revisiting the Ziegler-Nichols Step Response Method for PID Control”, Department of Automatic Control, Lund Institute of Technology, 2003 (submitted for review).

Åström K. J. and B. Wittenmark: “Computer Controlled Systems: Theory and Design”, Prentice-Hall (1984).