Proc. 1st IFAC Conf. on Mechatronic Systems (Darmstadt, Germany, 18–20 Sept. 2000), R. Isermann (ed.), VDI/VDE Gesellschaft Mess-und Automatiserungstechnik GMA, Düsseldorf, Germany.

DESIGN PROCEDURE FOR A LEARNING FEED-FORWARD CONTROLLER

Wubbe J.R. Velthuis, Theo J.A. de Vries and J. van Amerongen

Control Laboratory, C.J. Drebbel Institute, University of Twente
PO Box 217, 7500 AE Enschede, The Netherlands.
Phone: +31 53 489 2817, Fax: +31 53 489 2223,
icontrol@el.utwente.nl, http://www.rt.el.utwente.nl/icontrol/

Abstract: When a process is subject to reproducible disturbances, Learning Feed-Forward Control (LFFC) can be used to obtain accurate tracking. Until recently, LFFCs were designed by means of trial and error. In this paper, a design procedure is formulated, according to which a LFFC can be designed in a structured way. The design is based on qualitative knowledge of the process and the disturbances. This design procedure will result in a shorter design phase and better performing LFFC. As an example, the design procedure will be used to construct a LFFC for a linear motor. Simulations show that that the resulting LFFC is able to obtain accurate control.

Keywords: Intelligent Control, Neural Control, Learning Feed-Forward Control, Adaptation, Neural Networks, Spline Networks, Design Procedure

1 Introduction

Learning Feed-Forward Control (LFFC) is a control strategy that is able to accurately control processes that are subject to reproducible disturbances [1]. The LFFC consists of a compensator (*C*) and a neural network that is incorporated in the feed-forward path (fig. 1).

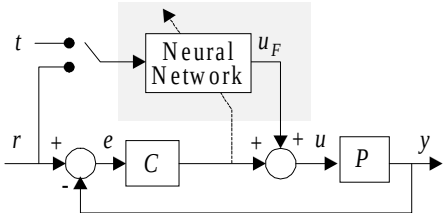

Figure1: Learning feed-forward control

During operation, the feed-forward controller is trained by the output, u_C, of the compensator, in such way that the tracking error, *e*, decreases. When fully trained, the feed-forward part generates a steering signal, u_F, that is needed to track the reference path. The feedback controller compensates all random disturbances. When the random disturbances are small compared to the reproducible disturbances, a high tracking performance will be obtained [2]. In our research, we specifically consider motions systems, i.e., systems where the task is to let an end-effector of the plant *P* move according to specifications. The input of the feed-forward controller consists either of the reference position, *r* [m], and derivatives thereof, or of the periodic motion time, *t* [s], in case of repeating motions.

The type of neural network that is used is a B-spline network (BSN) [3]. A BSN utilises piece-wise polynomial basis functions, known as B-Splines. A B-Spline of order *n* consists of piece-wise polynomial function of order *n*-1. In this research we will only consider B-Splines of order 2. To create an input-output mapping, B-Splines are distributed on the input domains of the BSN. This is done in such way that at each point of the input space, the sum of the B-Spline evaluations equals 1. The output of a BSN, u_F, is a weighted sum of the B-Spline basis functions (fig. 2).

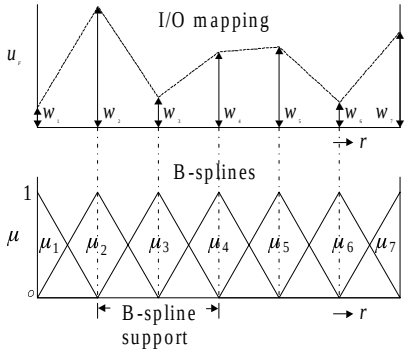

Figure 2: 2nd order B-spline network

Where μ_i is the evaluation of the i-th B-spline and w_i is its weight.

Training the BSN is done by adapting the weights of the network. This can be done either online, i.e. after each sample, or offline, i.e. after a specific motion has been completed. In this research we will only consider off-line training. The learning rule according to which the weights are adapted at the end of a motion is given in (1) [4]:

$$\Delta w_i = \gamma \frac{\sum_{k=0}^{T/h} \mu_i(kh) u_C(kh)}{\sum_{k=0}^{T/h} \mu_i(kh)} \qquad (1)$$

Where T is the motion time, h is the sample time, Δw_i is the adaptation of weight i and γ is the learning rate.

In the design of a LFFC, the following parameters have to be chosen: the compensator, the structure of the BSN, the distribution of the B-splines and the learning rate. Until recently, these parameters were chosen by rule of thumb. This may result in a long design phase and a non-optimal performing LFFC. To design a LFFC in a more structured way, a design procedure is formulated in section 2. Next, the design procedure is used in the design of a LFFC for a linear motor set-up. Finally, conclusions will be drawn.

2 Design procedure

Step 1: Design the compensator

Compensators are generally designed on the basis of a process model. Therefore, the first part of this step is to make a model the process that is suited for the design of the compensator.

As stated in the previous section, tracking performance is obtained by the feed-forward controller, while the compensator is used to compensate random disturbances. This implies that the compensator does not need to have a high performance and can be designed in such way that it features a robust stability. Note that the compensator preferably should not feature an I-like action, since the BSN already acts as one.

Step 2: Choose the structure of the BSN

Repetitive motions

In case of repetitive motions, all reproducible disturbances, either state or time dependent, are related to the periodic motion time. Therefore a 1-dimensional BSN that has the periodic time as input is sufficient.

Non-repetitive motions

To create a network structure for non-repetitive motions, qualitative knowledge of the process dynamics and the reproducible disturbances is needed. This qualitative knowledge is used to construct a model as shown in figure 2. When the process is linear, this implies that it can be written as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \qquad (2)$$

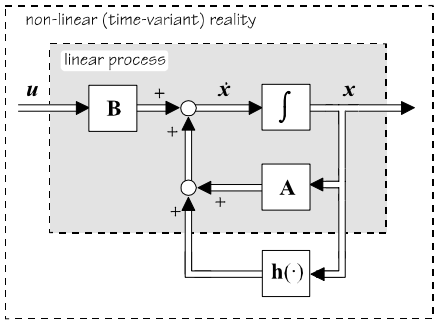

Figure 3: Non-linear system representation

The state vector is chosen such that it consists of positions and their corresponding velocities. This gives:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}, \ \mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \text{ and } \mathbf{B} = \begin{pmatrix} \mathbf{0} \\ \mathbf{B}_2 \end{pmatrix} \qquad (3)$$

In addition, we consider additive non-linear dynamics $\mathbf{h}(\cdot)$. If for example a process has both a velocity and a position dependent non-linearity, $\mathbf{h}(\cdot)$ would have the following form:

$$\mathbf{h}(\cdot) = \begin{bmatrix} 0 \\ \mathbf{h}(\mathbf{x}_1) + \mathbf{h}(\dot{\mathbf{x}}_1) \end{bmatrix} \qquad (4)$$

Note that we only use the qualitative knowledge to define the structure of each element in (3) and (4), not the precise realisation.

The desired feed-forward signal, $\mathbf{u}_d$, that makes $\mathbf{x}_1$ track the desired states, $\mathbf{x}_{1,d}$, is given by:

$$\mathbf{B}_2^{-1}\left(\ddot{\mathbf{x}}_{1,d} - \mathbf{A}_{22}\dot{\mathbf{x}}_{1,d} - \mathbf{A}_{21}\mathbf{x}_{1,d} - \mathbf{h}(\cdot)\right) = \mathbf{u}_d \qquad (5)$$

To enable the LFFC to accurately control the process for any type of motion, a BSN should be used that has each of the states given in (5) as an input. This often leads to BSNs that have multiple inputs (e.g. both the reference acceleration and the reference position).

However, in a BSN the number of network weights depends on the dimension of the input space in an exponential way. When high-dimensional BSN's are used, a large number of network weights and poor learning may result. This is known as the curse of dimensionality [5]. To prevent this, a network structure should be chosen

that consist of several low-dimensional BSN's in stead of 1 high-dimensional BSN. This is known as a parsimonious network structure. A parsimonious network structure is created by constructing one BSN for each collection of terms in (5) that depend on a common state.

Step 3: Choose the B-Spline distribution

Now that a network structure has been created the B-Spline distribution should be chosen. The width or support of the B-Splines (fig. 2) determines the accuracy with which a BSN is able to approximate an input-output relation. If the width is chosen small, the BSN is able to accurately approximate high-frequent signals. A large width will result in a more crude approximation.

Time indexed feed-forward controller

Intuitively one would choose B-Splines which have a small width. In that case the BSN is able to accurately learn any feed-forward signal. However, a stability analysis [6] showed that B-Splines that have a small width may result in unstable behaviour. The minimum width of the B-Splines for which the system remains stable, can be determined on the basis of a Bode plot of the closed loop transfer function. When $\omega_{\max}$ is the frequency for which the phase shift of the closed loop system exceeds 1.556 [rad] for the first time, the minimum width of the B-Splines is given by:

$$d_{\min} = \frac{2\pi}{\omega_{\max}} [\mathrm{s}] \qquad (6)$$

The Bode-plot can be obtained by using a model of the closed loop system, or can be determined by means of experiments. In the first case the model of the closed loop system should at least be accurate up to the frequency at which the phase shift of the negative closed loop system is equal to 1.556 rad. The higher order dynamics of the system do not need to be modelled accurately.

Now that the minimum width of the B-splines has been determined, the distribution of the B-splines can be chosen. When possible, prior knowledge of the process and the disturbances should be used to predict the shape of the desired feed-forward signal. On the basis of this signal, an initial B-Spline distribution is chosen. The width of the B-Splines should be chosen such that the BSN is just able to learn the desired feed-forward signal. This is done to keep the memory requirements of the BSN to a minimum and to guarantee fast learning. However, the initial B-Spline distribution may not result in a satisfactory tracking performance. To optimise the tracking performance, the B-Spline distribution can be adapted in an iterative way during the training phase. This is done by decreasing the width of the B-Splines at those positions where the tracking performance is poor.

The B-Spline distribution can also be obtained in an automated way using fuzzy clustering techniques [7]. These techniques analyse the signals the BSN is to learn and adapt the B-Spline distribution in such way that an optimal approximation results. The disadvantage of fuzzy clustering is that it is computational expensive.

Reference path indexed feed-forward controller

B-Splines that are defined on the reference path also cause unstable behaviour when they have a too small width. Their minimum width can be determined with the help of the minimum width of B-Splines defined on the periodic motion time, d_{min}. The main idea is that in case a B-Spline that is defined on the reference path, contributes to the output for less than d_{min} seconds, its width is too small.

For a *1-dimensional* state-indexed BSN, a stable B-Spline distribution can be chosen in 2 ways. The first method starts with defining a BSN that has the time as input. On the time domain B-Splines with an appropriate width are defined. Next, knowledge of the reference path is used to transform the positions of the B-Splines on the time domain to the corresponding position on the reference trajectory (fig. 4). What results are B-Splines which are active for at least d_{min} seconds. This method assumes that the reference trajectories are known in advance. However, this may not always be the case. The second method overcomes this problem.

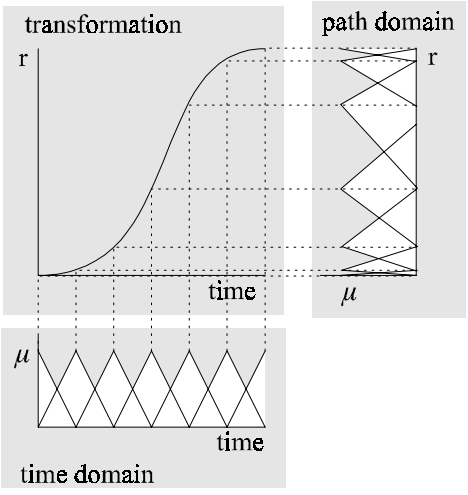

Figure 4: Conversion of B-splines

In stead of using full knowledge of the reference path, the second method uses prior knowledge of the reference trajectories, in terms of the maximum velocity, acceleration and jerk. Using the maximum reference jerk/acceleration/velocity and the value of d_{min}, we can calculate the minimum width of B-

splines defined on the reference acceleration / velocity / position respectively, such that they are active for at least d_{min} seconds for any type of motion. Consider for example B-splines that are defined on the reference position. We choose the width such that the B-Spline is active for d_{min} seconds when the process moves at maximum velocity, namely $d_{\min} v_{\max}$. This implies that such a B-Spline is active for more than d_{min} seconds for any motion slower than $v_{\max}$.

The latter method can also be applied in the *multi-dimensional* case. The resulting B-Spline distribution will be very conservative though, which will result in a loss of tracking performance.

Step 4: Choose the learning rate

Like a too small B-Spline width, also a too large learning rate can cause unstable behaviour. In [4] the maximum value of the learning rate for which the system remains stable has been determined, namely:

$$0 < \gamma < \frac{2}{\left| H(j\omega) \right|_\infty} \qquad (7)$$

Where $H(j w)$ is the closed loop transfer function. The value of the learning rate determines to what extent the weights are adapted. Setting $\gamma = 1/\left| H(j\omega) \right|_\infty$ causes instantaneous learning. All previously learned information will be forgotten. When setting $0 < \gamma < 1/\left| H(j\omega) \right|_\infty$, the weights of the BSN are not fully adapted. The information that is stored will partly consists of previously learned information and partly of the last presented learning information. When $1/\left| H(j\omega) \right|_\infty < \gamma \le 2/\left| H(j\omega) \right|_\infty$, the weights of the network are adapted too strong. The value of the weights show an oscillatory behaviour. Eventually they converge to a fixed point. When $\gamma > 2/\left| H(j\omega) \right|_\infty$ the weights also show an oscillatory behaviour, however their values diverge. We recommend to set γ close to 0.

Step 5:Training the LFFC

Time indexed feed-forward controller

In case of time indexed LFFC, training the BSN is straightforward.

Reference path indexed feed-forward controller

When the network structure consists of 1 BSN only, no special training strategy is needed. However, when additive networks are used, simultaneous training of all BSN's does guarantee that each BSN learns the feed-forward signal it is intended for. It may for example occur that a BSN that has the reference position as an input learns to compensate disturbances that depend on the velocity. To prevent this, a series of training experiments should be performed, in which one network is trained at a time. In the experiments, the reference trajectories should be chosen such that the influence of one specific disturbance is dominant. Only the BSN that is to compensate that disturbance is trained.

Step 6: Design an optional learning filter

In some situations, the minimum width determined in step 3 may not be small enough to obtain an acceptable tracking performance. To further increase the tracking performance a smaller B-Spline width would be needed. However when this is done unstable behaviour results. This problem can be overcome in 2 ways. Firstly, by improving the performance of the feedback controller. Namely, in step 3 it was stated that the frequency at which the phase shift of the closed loop system exceeds 1.556 [rad] for the first time determines the minimum B-Spline width. Improving the performance of the feedback controller causes this to happen at a higher frequency, thus allowing more narrow B-Splines. However, improving the performance of the feedback controller often decreases the robustness. Varying process parameters may easily destabilise the system.

Another approach is to add a learning filter (*L*) to the LFFC (fig. 5). The learning filter is designed in such way that it corrects the phase shift of the closed loop transfer function. This allows a smaller width of the B-splines. Since the learning filter is not contained in the closed loop system, it has no influence on the robustness.

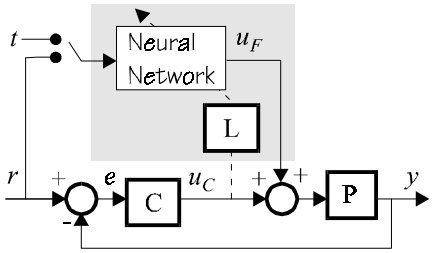

Figure 5: Learning filter

3 Simulations

In this section an LFFC will be designed on the basis of the previous guidelines. The process under control is a Linear Motion Motor System (LiMMS). The LiMMS is designed by Philips to perform linear motions for applications such as scanning and pick-and-place tasks. The motor configuration consists of a base plate on which permanent magnets are placed, and a translator, which contains the coils. The thrust force is generated by applying a 3-phase current to the coils.

During operation the translator experiences a sine-shaped disturbance force, which is caused by the strong magnetic interaction between the permanent magnets on the base plate and the iron cores that are mounted in the coils of the translator. This phenomenon is known as cogging. The cogging forces depend on the position of the translator. Furthermore, the LiMMS is subject to viscous friction. The model that is used in the following simulations is given in figure 6.

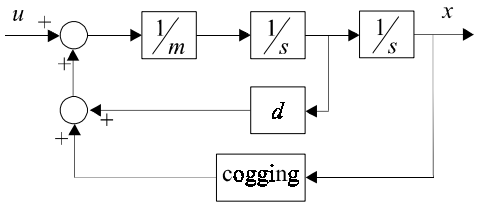

Figure 6: Model of the LiMMS

Where d=10 [Ns/m] represents the viscous friction, m=37 [kg] is the mass of the translator and x is its position. The cogging force has a magnitude of 10 [N] and has a period of 0.016 [m]:

$$F_{cogg}(x) = 10\sin\left(\frac{0.016}{2\pi}x\right)[\mathrm{N}] \qquad (8)$$

Step 1: Design the compensator

The compensator that is used is of the PD-type:

$$C(s) = K_p\left(\frac{s\tau_d + 1}{s\beta\tau_d + 1}\right) \qquad (9)$$

The values of the parameters is obtained by means of an auto-tuning mechanism, resulting in $K_p = 275280\ \ \tau_d = 0.02, \beta = 0.1$.

Step 2: Choose the structure of the BSN

From figure 6 and (8), we can derive that the desired feed-forward signal is given by:

$$37\ddot{x}_d + 10\dot{x}_d + 10\sin\left(\frac{0.016}{2\pi}x_d\right) = u_d \qquad (10)$$

We can thus conclude that the network structure should consist of three BSNs, that have the reference position, velocity and acceleration as input, respectively.

Step 3: Choose the distribution of the B-Splines

Next, a Bode-plot of the closed loop system is needed to determine the minimum width of the B-splines defined on the periodic time, see figure 7. It can be seen that a 90 degree phase shift occurs at approximately 230 rad/sec. Therefore:

$$d_{\min} = \frac{2\pi}{230}\sec = 0.0273\sec \qquad (11)$$

The BSN that has the *reference position* as input has to compensate the cogging forces. To be able to do this accurately we have chosen to define 10 B-Splines per cogging period of 0.016 [m], which results in a width of $3.5e^{-3}$ [m].

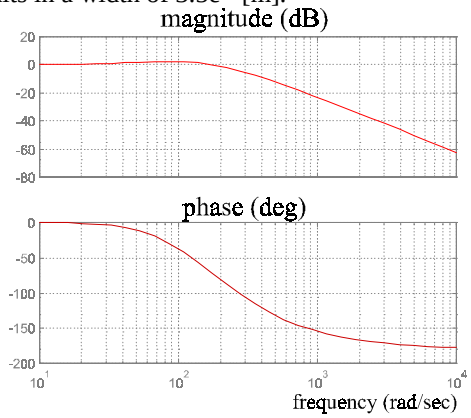

Figure 7: Bode plot of the closed loop system

To ensure that learning converges, this network must be trained at a maximum velocity of:

$$v_{\max} = \frac{3.5e^{-3}\,[m]}{0.0273\,[s]} = 0.13\,[ms^{-1}] \qquad (12)$$

The viscous friction is compensated by the BSN that has the reference velocity as input. Since this is a linear phenomenon only few B-Splines are needed. We choose 3 B-Splines that have a width of 2 [m/s]. The centres of the B-Splines are chosen at the following positions: {-2 [m/s], 0 [m/s], 2 [m/s]}. In the simulations the maximum reference acceleration will be 2 [m/s^2]. This would allow a minimum width of 2 [m/s^2] * 0.0274 [s] = 0.0548 [m/s]. We may thus conclude that for any motion this BSN will be stable.

The acceleration term is also linear. Therefore we choose 3 B-Splines that have a width of 3 m/s^2 and which are placed at {-3 [m/s^2], 0 [m/s^2], 3 [m/s^2]}. The maximum reference jerk is 10 [m/s^3]. This would allow a minimum width of 10 [m/s^3] * 0.0274 [s] = 0.274 [m/s^2]. This BSN will thus also remain stable.

Step 4: Choose the learning rate

In figure 7 it can be seen that $\left|H(j\omega)\right|_\infty = 1.3$. To ensure that learning is stable we choose $\gamma = 0.1$.

Step 5: Training the LFFC

Since the network structure consists of multiple BSNs, a special training phase will be needed. In the training phase 2 series of experiments will be performed. In the first series of experiments the BSN that has the reference position as input will be trained. To ensure that the position related phenomenon are dominant, the training motion will have a low velocity and a low acceleration. Next the

weights of this network are fixed and the BSNs that have the reference velocity and acceleration will be trained. This is done while performing high velocity / high acceleration motions. When these experiments are finished the weights of these networks are also fixed. To test its performance, the LFFC is used to track the motion shown in figure 8.

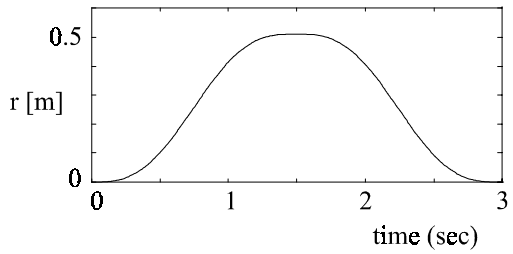

Figure 8: Reference position

In figure 9, the tracking error of LFFC before and after the training phase is shown. It can be seen that the LFFC is able to considerably improve upon the tracking performance of the feedback controller.

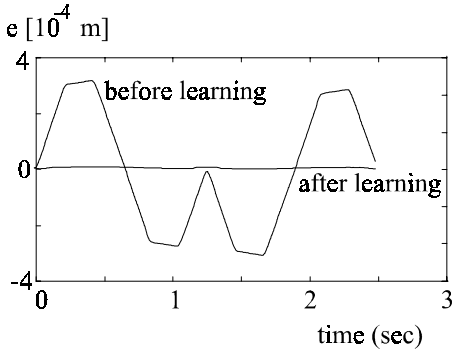

Figure 9: Reference position

4 Conclusions

In this paper, a design procedure is formulated that can be used to design a LFFC in a structured way. The procedure comprises the following steps:

1. *Design a compensator*. The compensator can be designed for a robust stability only. This can be done on the basis of a crude process model.
2. *Choose a network structure*. For repetitive motions a BSN is used that has the periodic motion time as input. For other motions a parsimonious network structure, consisting of several low-dimensional BSN's, is constructed on the basis of qualitative knowledge of the process and the disturbances.
3. *Choose the B-Spline distribution*. Firstly, the minimum width of the B-splines will be determined on the basis of a Bode plot of the closed loop system. This to guarantee that the LFFC will remain stable. Next, the B-Spline distribution is chosen on the basis of qualitative process knowledge or by means of fuzzy clustering.
4. *Choose the learning rate*.
5. *Perform training experiments*. In case the network structure consists of 1 BSN training does not need special attention. In case of several BSN a series of training experiments should be performed in which 1 BSN is trained at a time.
6. *Design an optional learning filter*. In case the minimum width of the B-Splines is not small enough to obtain the desired tracking performance, a learning filter can be added. The learning filter decreases the minimum width of the B-Splines, which results enables more accurate tracking.

This design procedure was used in the design of an LFFC for a linear motor. The LFFC consists of 3 BSNs that each compensate for one specific disturbance. This implementation gives a low number of weights, which is memory efficient and has a positive effect on the learning behaviour [3]. In simulations, the resulting LFFC showed to be able to accurately control the linear motor.

When looking at the design procedure, it can be seen that only qualitative knowledge of the process and the disturbances is needed in the design of an LFFC. Since qualitative process knowledge is often available, LFFC may be applicable to a wide range of applications.

5 References

[1] Starrenburg, J.G., W.T.C. Van Luenen, W. Oelen and J. Van Amerongen (1996), Learning feedforward controller for a mobile robot vehicle, *Control Eng. Practice*, **78** (9), 1481-1497.
[2] Otten, G., T.J.A. De Vries, J. Van Amerongen, A.M. Rankers and E.W. Gaal (1997), Linear Motor Motion Control Using a Learning Feedforward Controller, *IEEE/ASME Trans. Mechatronics*, **2** (3), 179-187.
[3] Brown, M. and C.J. Harris (1994), *Neurofuzzy adaptive modelling and control*, Prentice Hall, Hemel-Hempstead, UK.
[4] Velthuis, W.J.R., P. Schaak, T.J.A. De Vries and E. Gaal, Stability analysis of learning feed-forward control, To appear in: Automatica.
[5] Bossely, K.M., M. French, E. Rogers, C.J. Harris and M. Brown (1996), *Recent advances in neurofuzzy control system theory and design*, Proc. ViCAM, EFAM, Guimarães, Portugal, 117-133.
[6] Velthuis, W.J.R., T.J.A. De Vries and E. Gaal (1998), *Experimental verification of the stability analysis of Learning Feed-Forward Control*, Proceedings 37th IEEE Conference on Decision & Control, Tampa, FL, US, 1225-1229.
[7] Velthuis, W.J.R., T.J.A.d. Vries and J.v. Amerongen (1997), *Performance Optimisation of Learning Feed Forward Control*, Proc. IFAC Symp. AI in Real Time Control (AIRTC'97), Kuala Lumpur, Malaysia, 391-396.