Proceedings of the 1996 IEEE International
Symposium on Intelligent Control
Dearborn, MI • September 15-18, 1996

Learning feed forward control of a flexible beam

W.J.R. Velthuis, T.J.A. de Vries, J. van Amerongen

EL–BSC–RT, University of Twente, P.O.Box 217, 7500 AE Enschede, The Netherlands
e-mail: icontrol@el.utwente.nl; web: http://www.rt.el.utwente.nl/icontrol/

Abstract

Servo control is usually done by means of model-based feedback controllers, which has two difficulties. Firstly, the design of a well performing feedback controller requires extensive and time consuming modelling of the process. Secondly, by applying feedback control a compromise has to be made between performance and robust stability. The learning feed forward controller (LFFC) may help to overcome these difficulties. The LFFC consists of a feedback and a feed forward controller. The feedback controller is designed such that robust stability is guaranteed, while the performance is obtained by the feed forward controller. The feed forward controller is a function approximator that is adapted on the basis of the feedback signal. The LFFC is applied to a flexible robot arm, which has complex dynamics and unknown properties, such as friction. A stability analysis of the (idealised) LFFC is presented. Simulation experiments (with a non-idealised LFFC) confirm the results of this analysis and show that without extensive modelling a good performance can be obtained.

Keywords: intelligent control, neural control, adaptation, spline networks, flexible beams, stability analysis

1. Introduction

The principle of feedback control is used often to improve the dynamic behaviour of physical systems. Feedback controllers are generally model based: they are designed on basis of a mathematical model of the system under control. Application of such controllers has two difficulties.

Firstly, an accurate model of the physical system is needed in order to obtain a well–performing controller. This implies that quite some modelling and identification is needed. These processes are time consuming, complicated and sometimes even impossible, due to:
- process uncertainties
- varying process parameters.

The *second* drawback is that a compromise has to be made between performance and robustness. A high performance feedback controller generally does not feature a robust stability and/or performance. Small variations in process conditions deteriorate the performance and may even destabilise the system. On the other hand, a feedback controller with a large robustness often has sub-optimal performance.

These problems might be overcome by using a learning feed forward controller (LFFC), see figure 1.1 [4]. By adding a (learning) feed forward component to the feedback controller, an extra degree of freedom in the design of the controller is created. The feed forward part is intended to generate steering signals that make the output of the process, $y(t)$, follow the reference signal, $r(t)$.

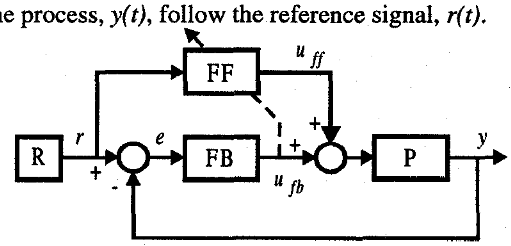

Figure 1.1 Learning feed forward control

Ideally, the output of the feed forward part, $u_{ff}(t)$, is equal to the control signal needed to let $y(t)$ match $r(t)$. In this case the error signal, and therefore the feedback signal, will equal 0. If the feed forward steering is not perfect, an error in y will occur. The output of the feedback part, $u_{fb}(t)$, will then try to reduce the error caused by the non ideal $u_{ff}(t)$. Thus $u_{fb}(t)$ can be interpreted as a measure for the error in $u_{ff}(t)$. This implies that if the feed forward part is adapted such that $u_{fb}(t)+u_{ff}(t)$ is applied in stead of $u_{ff}(t)$ when the same reference path is tracked again, this would result in a smaller error in $y(t)$.

A feed forward controller that can be adapted in this way can be implemented by a function approximator. The function approximator defines a mapping form the reference signal $r(t)$ to the steering signal $u_{ff}(t)$. During control, the mapping is adapted according to:

$$u_{ff}^{n+1}(t) = u_{ff}^{n}(t) + \gamma u_{fb}^{n}(t) \qquad (1.1)$$

In which n is an index denoting the number of times the path has been tracked and γ is the learning rate, $0 < \gamma \leq 1$.

In case $\gamma = 1$, the function approximator will perfectly fit the last presented example, but forgets previous presented examples. If $\gamma < 1$, the resulting mapping is some combination of all presented examples. Adaptation of the function approximator is continued until the error in $y(t)$ equals 0. In this situation the process tracks the reference signal perfectly.

Using a LFFC has two advantages. *Firstly*, separate means are created for obtaining good performance and robustness. In feedback control, both objectives have to be obtained by one controller. In the LFFC, the feedback controller guarantees the robustness, while the feed forward controller obtains a good performance of the control configuration. *Secondly*, the LFFC simplifies and shortens the design process. Since a good performance is obtained by the feed forward controller, the feedback controller merely has to be designed such that it has robust stability.

In previous research, the LFFC has been used to control a mobile robot [4]. This application concerned a process that had relatively simple dynamics and unmodelled properties, e.g. friction. The LFFC learned to deal with the unmodelled properties and was able to accurately control the mobile robot. In the research described in this paper, the LFFC will be applied to a flexible beam. The dynamics and the unmodelled properties of this process are more complex than those of the mobile robot. To prove that by learning, a well performing controller with a robust stability is obtained, a stability analysis will be presented. This analysis gives us insight in the properties of the learning mechanism as well. In section 2, the dynamics of the flexible beam are described. The theoretical background of the LFFC is presented in section 3. The results of simulation experiments are given in section 4.

2. Flexible beam

In a large variety of fields, robots are used to manipulate objects. To obtain accurate object manipulation, rigid robot arms are used currently. However, rigid robot arms are heavy and therefore need powerful, energy consuming actuators. In some fields of application, like space aviation, light manipulators and actuators having a low energy consumption are wanted. When reducing its weight, the robot arm becomes flexible and can no longer be controlled accurately by conventional controllers [3]. One method to solve the problem of vibration phenomena is to improve the stiffness of the arm by changing its structure. This method has limited, and therefore not always satisfying, results. Another method, pursued in this paper, is to use advanced control strategies. For the research on the control of vibrations, an extremely flexible robot arm, an aluminium beam, is used (figure 2.1).

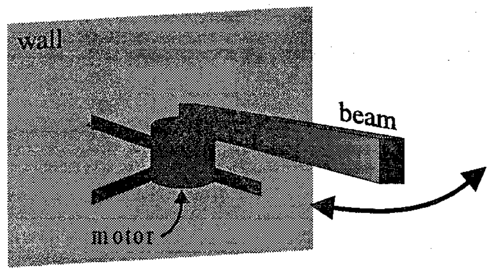

Figure 2.1 Flexible beam

The deflections that occur in a flexible beam can be modelled as the superposition of particular wave forms, the so called *modes*. Modes are identified by a mode number, where a higher mode number implies a shorter wave length. In figure 2.2 the first 3 modes of vibration are shown.

Figure 2.2 First 3 modes of flexibility

In previous work [2] a state-feedback controller has been used to control a flexible beam. To obtain a good performance a relatively accurate model of the process, considering 3 modes of vibration, was needed. This required extensive, time consuming modelling of the process and identification of the process parameters. To measure the state of the process, strain gauches were placed on the beam. Using the output of the strain gauches and the process model, the state of the system could be estimated.

Contrary to the state feedback controller, the feedback component used in the LFFC does not need an accurate process model. The model of the flexible beam that is used to design the feedback controller is a rigid, massless beam with a mass-spring combination at the tip (figure 2.3).

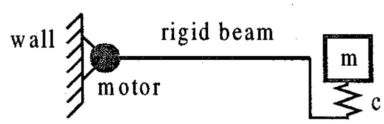

Figure 2.3 Simplified model of the flexible beam

3. LFFC

Stability analysis
For a stability analysis of the LFFC configuration (figure 1.1 and equation (1.1)) , three assumptions are made:

1. The process P is assumed to be linear.

2. The feed forward controller is an ideal function approximator:
   a) it is able to reproduce any mapping without distortion
   b) the mapping can be adapted locally. That is, a particular input-output pair of the mapping can be adapted without influencing the remainder.
3. The desired path, that is used as an input for the feed forward controller, is unique. This means that,

$$\forall t \neq t' : r(t) \neq r(t') \qquad (3.1)$$

The feed forward component is a mapping of the input *r(t)* to the output $u_{ff}(t)$. Because *r(t)* is unique, the output of the feed forward controller can be regarded as the result of a mapping from *t* to $u_{ff}(t)$. Hence, the feed forward steering can be seen as an additional reference signal, that is adapted each time the path has been traversed.

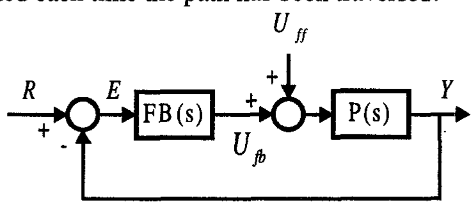

Figure 3.1 Feed forward controller interpreted as an additional reference signal

The LFFC is assumed to be stable if *y*(*t*) is bounded, after presenting the any number of times:

$$\{\forall n \in \mathbb{N} \mid \exists\, \varepsilon \in \mathbb{R} : y(t) < \varepsilon\} \qquad (3.2)$$

Using superposition, Y and U_{fb} can be written as:

$$Y^n = \frac{FB(s)P(s)}{\chi_c} R + \frac{P(s)}{\chi_c} U_{ff}^n \qquad (3.3)$$

$$U_{fb}^n = \frac{FB(s)}{\chi_c} R + \frac{-FB(s)P(s)}{\chi_c} U_{ff}^n \qquad (3.4)$$

In which,

$$\chi_c = 1 + FB(s)P(s) \qquad (3.5)$$

The feed forward controller is updated according to (1.1):

$$U_{ff}^{n+1} = U_{ff}^n + \gamma\, U_{fb}^n \qquad (3.6)$$

Substituting (3.4) in (3.6) results in

$$U_{ff}^n = \psi U_{ff}^{n-1} + \gamma \frac{FB(s)}{\chi_c} R \qquad (3.7)$$

In which,

$$\psi = 1 + \gamma \frac{-FB(s)P(s)}{\chi_c} \qquad (3.8)$$

Using (3.7) U_{ff}^n can be written as function of R and U_{ff}^0 :

$$U_{ff}^n = (\psi)^n U_{ff}^0 + \gamma \left(\frac{FB(s)}{\chi_c} \right) \sum_{i=0}^{n-1} (\psi)^i R \qquad (3.9)$$

In which,

$$U_{ff}^0 = 0 \qquad (3.10)$$

because in the first run the feed forward controller does not contain any information about the process control. Substituting (3.9) in (3.3) results in,

$$Y^n = \frac{FB(s)P(s)\left(\chi_c + \gamma \sum_{i=0}^{n-1} (\psi)^i \right)}{{\chi_c}^2} R \qquad (3.11)$$

The condition for (3.11) to converge is,

$$|\psi| = \left| 1 + \gamma \frac{-FB(s)P(s)}{\chi_c} \right| < 1 \qquad (3.12)$$

Therefore it is necessary that,

$$-\frac{\pi}{2} < \arg\left(\gamma \frac{FB(s)P(s)}{\chi_c} \right) < \frac{\pi}{2} \qquad (3.13)$$

and

$$\gamma \left| \frac{FB(s)P(s)}{\chi_c} \right| < 2\cos\left(\arg\left(\frac{FB(s)P(s)}{\chi_c} \right) \right) \qquad (3.14)$$

Neither (3.14) nor (3.13) can be satisfied for all s by choosing an appropriate learning rate γ. Stable control using LFFC therefore requires (3.13) and (3.14) to hold for the given process and feedback controller.

An interesting observation now is that (3.13) will typically be violated for high frequencies only. This gives rise to the hypothesis that *if the feed forward controller is chosen to not be a perfect function approximator, but one that will approximate the smooth (low frequency) part of a function only, LFFC will not become unstable.* In that case namely, learning will not occur in the high frequency range and only the stable standard feedback controller remains in that range. It is now possible to choose a γ such that also (3.14) is satisfied for the low frequency range. This hypothesis will be evaluated in simulation experiments.

Implementation

In the research described in this paper, the LFFC is used in a real-time environment. The function approximator should therefore perform the following actions within one sample interval:

- calculate the control signal
- learn the desired control action

Most neural networks are computationally expensive and cannot be used for real-time applications without powerful computers. In this research a B-spline network (BSN) [1] which is computationally attractive and learns fast, is used to implement the LFFC.

In general terms, a BSN defines a mapping from input x to output y using B-spline basis functions. A B-spline of order n is a piece-wise polynomial of order n-1 (see figure 3.2). The interval over which a B-spline does not evaluate to 0 is called its support here.

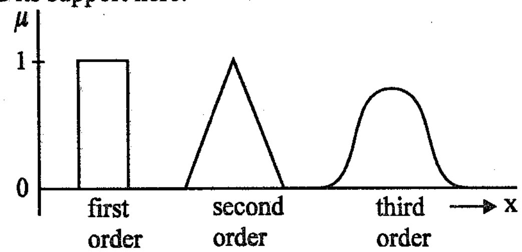

Figure 3.2 B-splines

Hence, to realise a mapping from t to u_{ff} with a BSN, m B-splines are placed on the domain of t, $\mathbb{T}$. The B-splines are placed such, that the sum of the evaluations of each of the B-splines, $\mu_i(t)$, is equal to 1 for all $t \in \mathbb{T}$(figure 3.3).

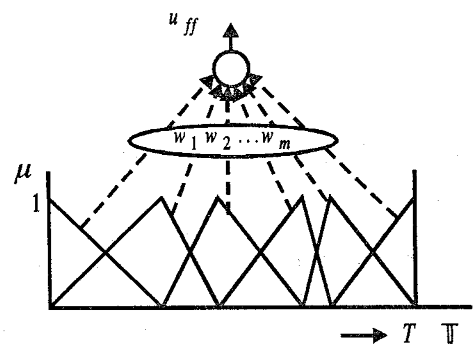

Figure 3.3 B-spline network (BSN)

The output of the BSN at input t is calculated as a weighted sum of the B-spline evaluations :

$$u_{ff}(t) = \sum_{i=1}^{m} \mu_i(t) w_i \tag{3.15}$$

Note that (3.15) is how the feed forward controller is actually implemented, and hence is the time domain representation of the realised U_{ff}. As this is a piece-wise linear signal, it violates assumption 2a (see stability analysis). In fact, our choice of implementation by means of a BSN implies that only mappings with a certain 'smoothness' can be approximated accurately. What this smoothness is depends on the width of the B-spline basis functions. To accurately fit a mapping that is not so smooth (has high frequency components), B-splines that have a small support are required. In case the B-splines have a wide support, rapid changes in the desired mapping will be averaged by the BSN (see figure 3.4). Hence, the BSN is fit to test the hypothesis formulated at the end of the stability analysis.

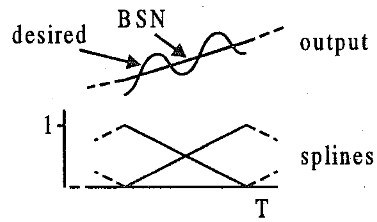

Figure 3.4 Function approximation using a BSN

As stated, the feedback signal is a measure for the error in the output of the LFFC. The error in the output of the BSN is defined as:

$$E(t) = \frac{1}{2}\left(u_{fb}(t)\right)^2 \tag{3.16}$$

The goal of learning is to minimise E for all t. This can be realised in the BSN by adapting the weights, w_i, according to the following learning rule [1]:

$$\Delta w_i = \gamma\, u_{fb}\, \mu_i(t) \tag{3.17}$$

In which γ , $0 < \gamma \leq 1$, is the learning rate. Learning rule (3.17) has about the same effect as (1.1). However, an important difference is that if we apply (3.17) at a certain time instant, we modify the mapping over the complete support of basis function i, and not just the mapping at that particular time instant as in (1.1). Hence, our choice of implementation with this learning rule implies that also assumption 2b (see stability analysis) is violated. The expectation is that the somewhat differing learning behaviour will not influence the stability, however.

Application to the flexible beam

The design of a LFFC can be split up in the following 3 steps:

1. definition of the reference path.
2. design of the feedback controller.
3. design of the feed forward controller, which consists of:
   - choice of inputs of the controller
   - definition of splines on the inputs
   - choice of the learning rate

The reference path is chosen to be a cycloid function, defined by:

$$\theta_r(t) = \begin{cases} \frac{A}{t_r}t - \frac{A}{2\pi}\sin\left(\frac{2\pi}{t_r}t\right) & t \le t_r \\ A & t < t_r \end{cases} \quad (3.18)$$

Where A is the final tip angle and t_r is the rise time. In this research A=1 rad and t_r=2 sec. As this is a smooth reference path, the controlled flexible beam should be able to track this path accurately.

Since the feedback controller only has to guarantee robust stability, a relative simple controller can be used. In this research the feedback controller will be implemented as a PD controller. On the basis of the model discussed in section 2, the following controller is obtained, using root-locus techniques.

$$FB(s) = s + 1 \quad (3.19)$$

The reference path is unique, and hence the LFFC can be realised as a mapping from t to u_{ff}. Next, the distribution of B-spline basis functions on the domain of t has to be defined. This is done on the basis of the stability analysis of the LFFC. In figure 3.5 the Bode-plot of stability criterion (3.14) and (3.13) for the flexible beam is given.

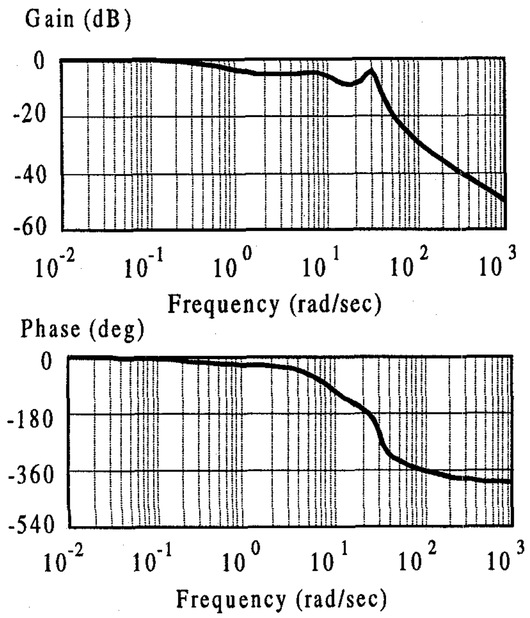

Figure 3.5 Stability analysis of the system

It can be seen that for $\omega > 8$ rad/s the phase shift of the criterion exceeds 90°. So (3.13) is violated above this frequency, and a LFFC controlled beam with an ideal function approximator will be unstable in this frequency range. To verify this, the two sets of B-splines that are depicted in figure 3.6 will be used in simulation experiments. The B-splines in figure 3.6a are too wide to learn frequency components of 8 rad/s in a signal, so the LFFC will be stable. The B-splines in figure 3.6b have a small width, which enables them to learn such frequency components, and hence are expected to cause an unstable LFFC controlled beam.

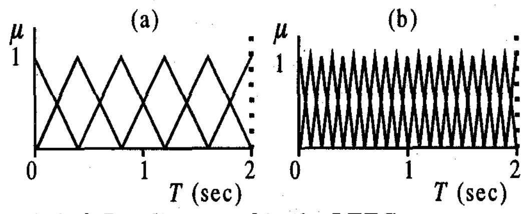

Figure 3.6a,b B-splines used in the LFFC

The learning rate is chosen by the rule of the thumb. To enable fast learning and some averaging, the learning rate is chosen to be equal to 0.5. With this value, (3.14) is satisfied.

4. Simulation experiments

The simulation experiments presented in this section consist of a number of learning cycles, denoted as 'runs'. In each run, the LFFC is intended to make the tip of the flexible beam track the reference path and learn, such that in the next run the tracking is more accurate. The model of the flexible beam that is used in the simulations considers the first 2 modes of flexibility only.

In the first series of experiments a LFFC using the B-spline distribution given in figure 3.6a, is applied to the flexible beam. Figure 4.1 shows the reference path and the tip angle.

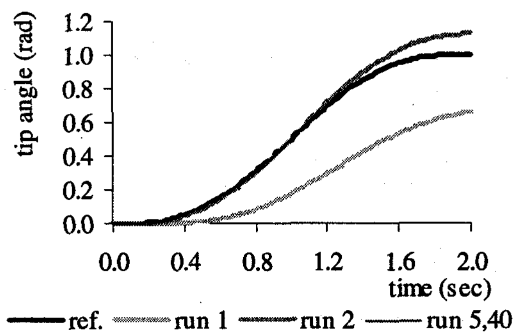

Figure 4.1 Performance of a LFFC that cannot learn high frequencies

In the first run the feedback controller, which is designed for robust stability, is not able to make the process follow the reference path. By learning, the performance of the LFFC is improved, such that after 4 runs, the reference signal is tracked well. As predicted, the LFFC does not become unstable in case the learning is continued.

Next the B-spline distribution that is depicted in figure 3.6b will be used to control the flexible beam. These B-splines are able to accurately learn functions that have frequency components higher than 8 rad/sec. In figure 4.2 the results of this experiment are shown.

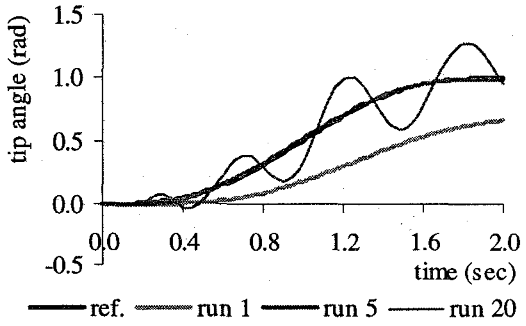

Figure 4.2 Performance of a LFFC that can learn high frequencies

Although the LFFC seems to have learned well after 5 runs, the system becomes unstable if the learning is continued. Apparently, the LFFC first learns the low frequency components, at which the LFFC is stable, and then the higher, unstable frequencies.

Finally, the robust stability of the LFFC is researched. After a LFFC has learned to accurately control an unloaded flexible beam, a payload of 0.25 kg will be 'attached' to the tip. The performance of the LFFC is shown in figure 4.3. The performance of the controlled beam deteriorates initially, but the system remains stable. After 4 runs the LFFC has adapted itself such that the optimal performance is regained. The system stays stable as learning is continued.

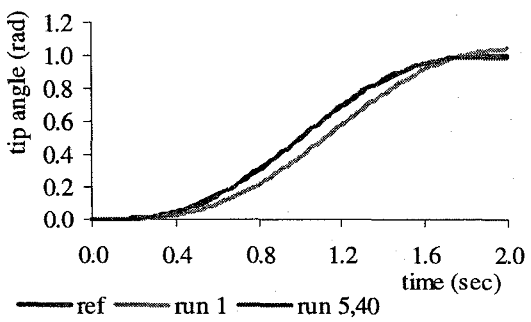

Figure 4.3 Performance of a LFFC on a flexible beam with payload (mass=0.25 kg)

5. Conclusions

A stability analysis revealed that a linear process $P(s)$ controlled by a LFFC with an ideal function approximator as feed forward controller and a feedback controller $FB(s)$ remains stable iff

$$\left|1+\gamma\frac{-FB(s)P(s)}{1+FB(s)P(s)}\right|<1$$

This condition will typically be violated for high frequencies only. Hence, one may expect that *if the feed forward controller is chosen to be a function approximator that will approximate the smooth (low frequency) part of a function only, a LFFC controlled system will not become unstable.*

Simulation experiments of a LFFC controlled flexible beam with a B-spline network as feed forward controller and a PD feedback controller confirmed this hypothesis. If the B-spline basis functions are given a wide support, the network can only learn a smooth mapping, and the controlled system remains stable. If the basis functions are given a small support, also a less smooth mapping can be learned and the controlled system becomes unstable. This indicates that a stability analysis may help to find an appropriate basis function distribution when designing such a LFFC. Further research will be done to find quantitative rules for this.

The obtained LFFC combines good performance with robust stability. The LFFC was able learn to accurately control the flexible beam within a small number of learning cycles. Varying process parameters did not destabilise the system. The LFFC was able to adapt to them in a fast way, such that optimal performance was retained quickly. This result suggests that a LFFC with a B-spline network as feed forward controller can control processes that have difficult dynamics, such as flexible beams, without extensive and time-consuming modelling. Since the feedback controller used in the LFFC merely has to guarantee robust stability, a relatively simple model of the process is sufficient for controller design.

6. References

[1] Brown, M., and Harris, C., (1994), *Neurofuzzy adaptive modelling and control*, ISBN 0-13-134453-6.

[2] Krijger, M.J. de, (1995), *Bond graph modelling and control of flexible robot arms*, PhD Thesis, Univ. of Twente, Enschede, Netherlands, ISBN 90-9008645-4.

[3] Kruise, I. (1990), *Modelling and control of a flexible beam and robot arm*, PhD Thesis, Univ. of Twente, Enschede, Netherlands, ISBN 90-90003666-0

[4] Starrenburg, J.G., W.T.C. van Luenen, W. Oelen and J. van Amerongen (1996), *Learning feed forward controller for a mobile robot, Control Eng. Practise*, submitted.