A MRAS–BASED LEARNING FEED–FORWARD CONTROLLER

Job van Amerongen

Control Engineering, University of Twente
J.vanAmerongen@utwente.nl

Abstract: Inspired by learning feed–forward control structures, this paper considers the adaptation of the parameters of a model–reference based learning feed–forward controller that realizes an inverse model of the process. The actual process response is determined by a setpoint generator. For linear systems it can be proved that the controlled system is asymptotically stable in the sense of Liapunov. Compared with more standard model reference configurations this system has a superior performance. It is fast, robust and relatively insensitive for noisy measurements. Simulations with an arbitrary second–order process and with a model of a typical fourth–order mechatronics process demonstrate this.

Keywords: Model reference adaptive control, learning control, feed–forward control

1. INTRODUCTION

Model Reference Adaptive Controllers generally use a model in parallel with the process. The parallel model determines the desired behavior of the process (as in figure 1a) or it is used as an adjustable model for estimating the process parameters (as in figure 1b). Model Reference Adaptive Controllers have been applied, e.g. in autopilots for ships (van Amerongen, 1984), (van Amerongen *et al.*, 1990).

A setpoint generator, in mechatronic systems called a motion profile, can also act and be used as a reference model. This leads to the basic structure of figure 1c. In recent years there has been an increasing interest in feed–forward controller structures. These structures use a–priori knowledge of the process to generate proper steering signals without the need to wait for an error signal. Such structures can considerably improve the performance of the controlled system with respect to reference changes and measurable disturbances. In its most elementary form a feed–forward controller should contain the inverse model of the process. To realize such an inverse model it must be combined with a low–pass filtering structure with at least the same order as the process. Recently feed–forward structures for processes with repetitive disturbances have been developed in the form of Iterative Learning Control (ILC) (Moore, 1998), (Steinbuch and van de Molengraft, 2000), (Verwoerd, 2005) and Learning Feed–Forward Control (LFFC). A repetitive disturbance is a disturbance that comes back in almost the same form at fixed time intervals. In ILC the steering signal that is required to compensate for such a disturbance is stored in a memory and played back one cycle later. A forgetting mechanism and an update mechanism keep the contents of the memory module up to date. The information about the system needed to generate the feed–forward signal is stored in a kind of delay line.

In the Learning Feed–Forward Control setting the information about the system is stored in a function approximator, which could be realized e.g. by means of a neural network or with B-splines. This type of information representation enables that non-linear effects can easily be learned and stored. For repetitive disturbances the input for the neural network is chosen as the time (after starting a new cycle) (Otten *et al.*, 1997), (Velthuis, 2000). The Learning Feed–Forward control approach can be extended to non–repetitive, state–dependent disturbances. Instead of time, the reference signal and its derivatives are used

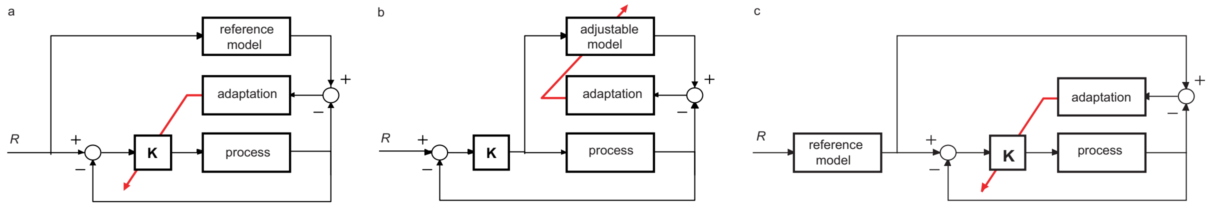

Fig. 1. a) Model Reference Adaptive System for adaptation of controller parameters, b) Model Reference Adaptive System with adjustable model for parameter identification, c) MRAS structure with Setpoint Generator

as inputs for the network. Again, non–linearities can be easily compensated, but at the price of a lot of memory and for increasing system order and more state–dependent non–linearities, long training times. To deal with this problem, alternative function approximators such as support vector machines have been applied ((Velthuis, 2000), (de Vries *et al.*, 2000), (de Kruif and de Vries, 2003), (de Kruif, 2004)).

2. MRAS-BASED FEED–FORWARD CONTROL

For systems that can be approximated by linear transfer functions the representation as a transfer function requires almost no memory and is thus very efficient. When we realize the setpoint generator by means of a so called state variable filter, the states of such a filter —which are derivatives of the output— can be used to generate an inverse model. This will be illustrated with the second–order example of (figure 2). The process is described by the transfer function H_p and the reference model by H_{ref}:

$$H_p = \frac{1}{a_p s^2 + b_p s + c_p} \tag{1}$$

$$H_{ref} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2}$$

The transfer from reference R to process output C is:

$$H_{tot} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \frac{a_m s^2 + b_m s + c_m}{a_p s^2 + b_p s + c_p} \tag{3}$$

When the parameters a_m, b_m and c_m are equal to respectively a_p, b_p and c_p, H_{tot} is equal to the desired response, given by H_{ref}.

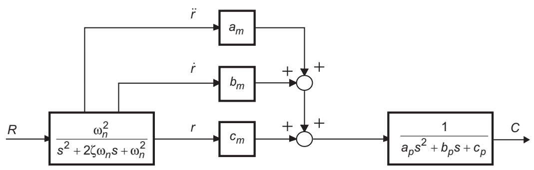

Fig. 2. Realization of an inverse model of the process

We should try to find a learning mechanism that, based on the error between the output r of the setpoint generator and the process output C, adjusts the parameters a_m, b_m and c_m such that they converge to the process parameters. When there are disturbances present it makes sense add a feedback loop. However, when we model the disturbances as a 'constant' disturbance, this disturbance could also be compensated in a feed–forward manner by adding a 'constant' input d_m to be found by the learning mechanism. This leads to the controller structure of figure 3, where the derivative–generating structure of the state variable filter is clearly visible.

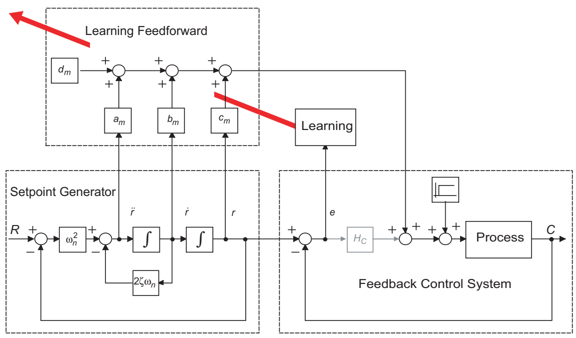

Fig. 3. Learning Feed–Forward Controller

3. ADAPTIVE LAWS

In a model-reference adaptive system the reference model can play the role of a setpoint generator (figure 1c). This suggests that we can use the well–known Liapunov approach to find stable adaptive laws for the feed–forward parameters a_m, b_m, c_m and d_m. We will continue with the second–order example, but the approach is equally well applicable to higher order systems. We assume that the process parameters are unkown and vary only slowly. At this stage we also assume that the (proportional) controller transfer $H_C = K_p$ is zero. The feedback is only used in this case to generate the error signal e for the learning mechanism. Closing the feedback loop by means of a constant gain K_p just changes the process parameters and can be compensated for by different values of the adjustable parameters. We assume that the disturbance is zero. Later on we will see that the parameter d_m can be found in a similar manner as the other parameters. The design problem is thus: *Find (stable) adjustment laws for the adjustable parameters a_m, b_m and c_m such that the error e between the setpoint generator and the process as well as the error in the feed–forward parameters asymptotically go to zero.*

We rewrite equations (1) and (2) in state space form. This yields for the reference model:

$$\dot{x}_m = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} x_m + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} R \qquad (4)$$

We define an extra state $x_{m,3}$ as:

$$x_{m,3} = \dot{x}_{m,2} = \omega_n^2 R - \omega_n^2 x_{m,1} - 2\zeta\omega_n x_{m,2} \qquad (5)$$

This yields for the process:

$$\dot{x}_p = \begin{bmatrix} 0 & 1 \\ -\frac{c_p}{a_p} & -\frac{b_p}{a_p} \end{bmatrix} x_p + \begin{bmatrix} 0 & 0 \\ \frac{c_m}{a_p} & \frac{b_m}{a_p} \end{bmatrix} x_m + \begin{bmatrix} 0 \\ \frac{a_m}{a_p} \end{bmatrix} x_{m,3} \qquad (6)$$

Because the parameters a_m, b_m and c_m should converge to a_p, b_p and c_p we define the error in the parameters as:

$$\frac{a_m - a_p}{a_p} = \frac{a_m}{a_p} - 1 = a \rightarrow \frac{a_m}{a_p} = 1 + a$$
$$\frac{b_m - b_p}{a_p} = b'_m - b'_p = b \qquad (7)$$
$$\frac{c_m - c_p}{a_p} = c'_m - c'_p = c$$

Substitution of eqns. (7) in eqn. (6) yields

$$\dot{x}_p = \begin{bmatrix} 0 & 1 \\ -c'_p & -b'_p \end{bmatrix} x_p + \begin{bmatrix} 0 & 0 \\ c'_m & b'_m \end{bmatrix} x_m + \begin{bmatrix} 0 & 0 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} x_m + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} R + \begin{bmatrix} 0 \\ a \end{bmatrix} x_{m,3} \qquad (8)$$

We define the error $e = (e_1, e_2)^T = x_m - x_p$. By subtracting eqn. (8) from eqn. (4, it follows that:

$$\dot{e} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} x_m + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} R - \begin{bmatrix} 0 & 0 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} x_m - \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} R - \begin{bmatrix} 0 & 1 \\ -c'_p & -b'_p \end{bmatrix} x_p - \begin{bmatrix} 0 & 0 \\ c'_m & b'_m \end{bmatrix} x_m - \begin{bmatrix} 0 \\ a \end{bmatrix} x_{m,3} \qquad (9)$$

This simplifies into:

$$\dot{e} = \begin{bmatrix} 0 & 1 \\ -c'_m & -b'_m \end{bmatrix} x_m - \begin{bmatrix} 0 & 1 \\ -c'_p & -b'_p \end{bmatrix} x_p - \begin{bmatrix} 0 \\ a \end{bmatrix} x_{m,3} \qquad (10)$$

This can be rewritten as

$$\dot{e} = A_m x_m - A_p x_p - \begin{bmatrix} 0 \\ a \end{bmatrix} x_{m,3} \qquad (11)$$

or

$$\dot{e} = A_m x_m \underbrace{-A_p x_m + A_p x_m}_{=0} - A_p x_p - \begin{bmatrix} 0 \\ a \end{bmatrix} x_{m,3} = A' x_m + A_p e - \begin{bmatrix} 0 \\ a \end{bmatrix} x_{m,3} \qquad (12)$$

where matrix A' contains the parameter errors b and c

$$A' = A_m - A_p = \begin{bmatrix} 0 & 0 \\ -c & -b \end{bmatrix} \qquad (13)$$

We combine the parameter errors a, b and c in the error vector ε and in the error matrix A and redefine x_m:

$$\varepsilon^T = (-c, -b, -a)$$
$$A = \begin{bmatrix} 0 & 0 & 0 \\ -c & -b & -a \end{bmatrix} \qquad (14)$$
$$x_m{}^T = (x_1, x_2, x_3)$$
$$\dot{e} = A x_m + A_p e$$

In order to proof the asymptotic stability of e we define a positive definite Liapunov function:

$$V = e^T P e + \varepsilon^T \alpha \varepsilon \qquad (15)$$

with P a positive definite symmetrical matrix and α a diagonal matrix with in principle arbitrary coefficients > 0. For a stable adaptive system the time derivative of V should be ≤ 0:

$$\frac{dV}{dt} = \dot{e}^T P e + e^T P \dot{e} + \dot{\varepsilon}^T \alpha \varepsilon + \varepsilon^T \alpha \dot{\varepsilon} \qquad (16)$$

Substituting $\dot{e}$ from (14) and rearranging the terms yields:

$$\frac{dV}{dt} = e^T A_p^T P e + e^T P A_p e + 2 e^T P A x_m + 2 \varepsilon^T \alpha \dot{\varepsilon} \qquad (17)$$

With the first two terms of (17) equal to

$$e^T \left(A_p^T P + P A_p \right) e = -e^T Q e \qquad (18)$$

it follows that:

$$\frac{dV}{dt} = -e^T Q e + 2 e^T P A x_m + 2 \varepsilon^T \alpha \dot{\varepsilon} \qquad (19)$$

When Q is a positive definite symmetrical matrix the first term in eqn (19) is always negative or zero and when the other terms are made equal to zero, $\frac{dV}{dt} \leq 0$. When A_p is stable, we can find for any positive definite symmetrical matrix Q a positive definite symmetrical matrix P by solving the Liapunov equation

$$A_p^T P + P A_p = -Q \qquad (20)$$

If the process matrix A_p is not stable, the process must be stabilized first by appropriate feedback. Apart from the stability of the adaptive system, feed–forward compensation can only be applied for stable processes.

The adjustment rules for the feed–forward gains follow from the condition that the remaining terms in eqn. (19) should be zero:

$$2e^T PAx_m + 2\varepsilon^T \alpha \dot{\varepsilon} = 0 \qquad (21)$$

After simplification (taking into account that only the last row of A contains values $\neq 0$) this yields:

$$\frac{d\varepsilon}{dt} = -\alpha^{-1}(p_{21}e + p_{22}\dot{e})x_m \qquad (22)$$

Because we assume that the adaptation is fast compared with the variations in the process parameters it follows that

$$\frac{d\varepsilon}{dt} = \frac{d}{dt}\begin{bmatrix} -(c_m - c_p)/a_p \\ -(b_m - b_p)/a_p \\ -(a_m/a_p - 1) \end{bmatrix} = -\frac{d}{dt}\begin{bmatrix} c_m/a_p \\ b_m/a_p \\ a_m/a_p \end{bmatrix} \qquad (23)$$

When the unknown parameter a_p is part of the ('arbitrary') adaptive gains α^{-1}, this yields the following adjustment rules:

$$\begin{aligned} a_m &= \frac{1}{\alpha_{33}} \int [(p_{21}e + p_{22}\dot{e})x_{m,3}]\, dt + a_m(0) \\ b_m &= \frac{1}{\alpha_{22}} \int [(p_{21}e + p_{22}\dot{e})x_{m,2}]\, dt + b_m(0) \qquad (24) \\ c_m &= \frac{1}{\alpha_{11}} \int [(p_{21}e + p_{22}\dot{e})x_{m,1}]\, dt + c_m(0) \end{aligned}$$

Like in any MRAS-based system, adaptive disturbance compensation can be added, by realizing that the parameter d_m acts on an extra input signal 1, instead on one of the state variables:

$$d_m = \frac{1}{\gamma} \int [(p_{21}e + p_{22}\dot{e})1]\, dt + d_m(0) \qquad (25)$$

$1/\alpha_{11}$, $1/\alpha_{22}$, $1/\alpha_{33}$ and $1/\gamma$ are called the adaptive gains. They determine the speed of adaptation and they can, in principle, be arbitrarily chosen.

The equations (24) can be generalized to equations for higher–order systems. For an n^{th}–order system we find for parameter $a_{m,i}$:

$$\frac{da_{m,i}}{dt} = \frac{1}{\alpha_{ii}} \left(\sum_{k=1}^{n} p_{nk}e_k \right) x_{m,i} \qquad (26)$$

In the adjustment laws the derivative of the error is needed. This derivative can be obtained by means of a (second–order) state variable filter. The bandwidth of this state variable filter has to be chosen at least ten times larger than the bandwidth of the setpoint generator in order not to endanger the stability of the system. On the other hand, the combination of a state variable filter with a not too large bandwidth and use of the model states for the adaptation as well as for the control, makes the system relatively insensitive for noisy measurements and leads to a robust system.

4. EXPERIMENTS

4.1 Second–order system

To illustrate the performance of the controller a series of experiments has been carried out with the modeling and simulation program 20-sim (*www.20sim.com*, 2006), (van Amerongen and Breedveld, 2003). In figure 4 the responses are given for the feed–forward control of a second order process, according to figure 3. Responses are plotted of the outputs of the setpoint generator and the process, of the Liapunov error ($p_{21}e + p_{22}\dot{e}$) and of the input signal for the process. The experiment consists of three phases.

$t = 0 - 20[s]$ fixed, incorrect parameters
$t = 20 - 40[s]$ parameter adaptation on
$t = 40 - 60[s]$ at $t = 40$ a disturbance of amplitude 1 is applied at the process input

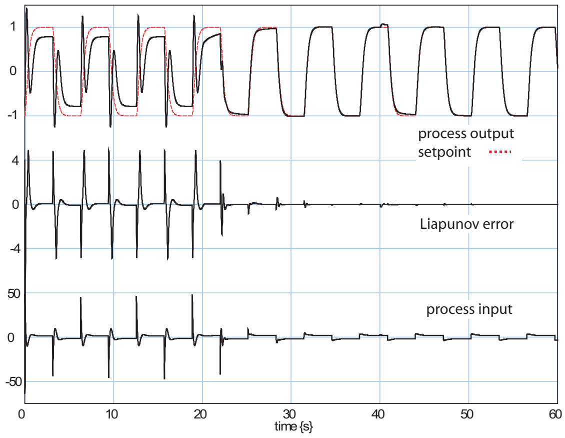

Fig. 4. Responses of feed-forward control of a second–order system

In figure 5 the parameter values are shown. The end value of the parameters is within 1% of the actual process values.

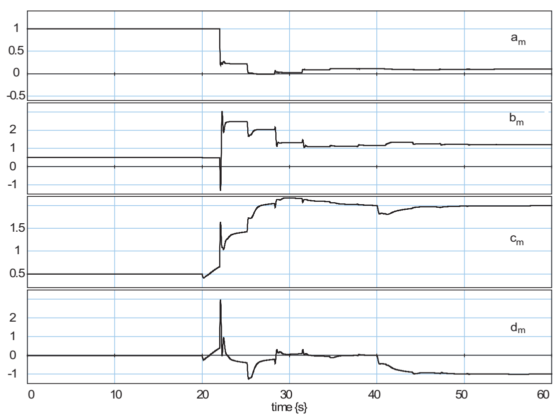

Fig. 5. Parameters of the experiment of figure 4

The proportional controller gain K_p is chosen as 5. Selecting $K_p = 0$ would give an even worse response during the phase where the adaptation is off, but hardly influences the rest of the responses, although a larger value of K_p of course helps in quicker reducing the influence of the disturbance. The process parameters are :$a_p = 0.1$, $b_p = 1.2$ and $c_p = 2$. The initial values of the feed–forward gains are $a_m = 1$, $b_m = 0.5$ and $c_m = 0.5$, and the parameters of the setpoint generator $\omega_n = 5$ and $\zeta = 1$. Q is chosen positive definite and P is solved by realizing that it can be seen as the solution of the non-linear differential equation (27). This solution can easily be obtained by simulating:

$$\frac{dP}{dt} = A_m^T P + P A_m + Q \qquad (27)$$

The adaptation is very fast and the influence of the disturbance is hardly visible. The system also performs very well when there are noisy measurements. In this case it is advantageous to switch off the feedback controller completely, or almost completely, in order to reduce the influence of the noise on the steering signal of the process. Figure 6 gives the responses of a system with noisy measurements and figure 7 the corresponding parameter values. The adaptation is switched on now at $t = 0$ and the disturbance is applied at $t = 20$. Please note that the scales along the plots differ from the first experiment.

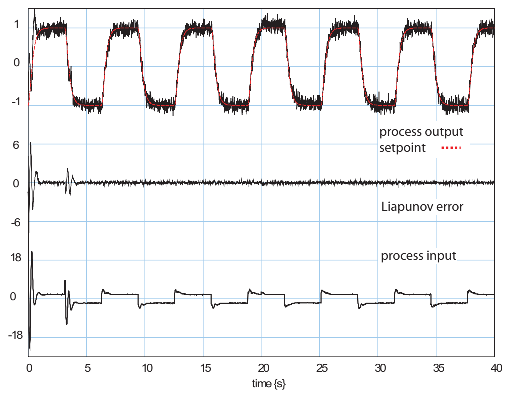

Fig. 6. Performance of the system with noisy measurements

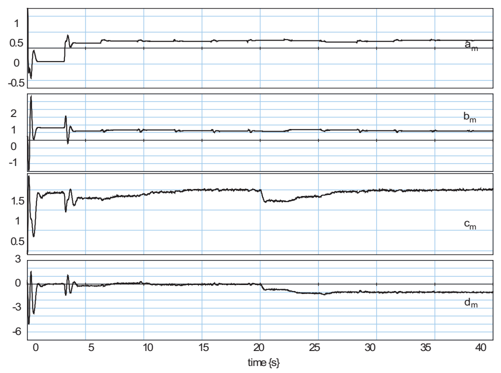

Fig. 7. Parameter adaptation in the case of noisy measurements

This structure deals much better with noisy measurements than conventional Model Reference Systems. The steering signal is very smooth and the parameter estimation is unbiased. The parameters rapidly converge to within 1% of the correct values.

4.2 Fourth–order mechatronic system

To see if the system also performs well for a more realistic mechatronic system, the process is replaced by a model of a mechatronic servo system given in figure 8.

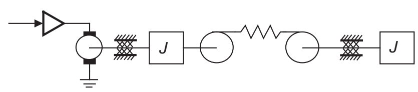

Fig. 8. Mechatronic System

This fourth–order system consists of a current amplifier with gain 1, a DC-motor (with friction $d = 0.1152$ $[Nms]$ and inertia $J = 0.00262$ $[kgms^2]$), a transmission (with compliance $k = 1.54$ $[N/m]$) and transmission ration 1:4 and a load (with friction $d = 0.0001$ $[Nms]$ and inertia $J = 0.056$ $[kgms^2]$). The second–order feed–forward controller is basically not changed, except for the fact that the parameter c_m is not adjusted, because of the pure integrator in the servo system. (If it is adjusted it finally converges to zero anyhow.) This implies that the feed–forward controller can only control the dominant second-order behavior of the process. Excitation of the resonance frequencies due to the compliance, in the order of 30 $[rad\ s^{-1}]$ should be prevented by adding an extra motion profile in front of the setpoint generator. A cycloidal motion profile with 2 seconds between the start and stop of the (desired) motion is chosen. In addition, the adaptive gains should be chosen small enough to prevent that sudden parameter changes excite the resonant frequencies. The results of the simulation are shown in figure 9. The parameters a_m and b_m converge to the correct values, corresponding to second–order model of the process (figure 10).

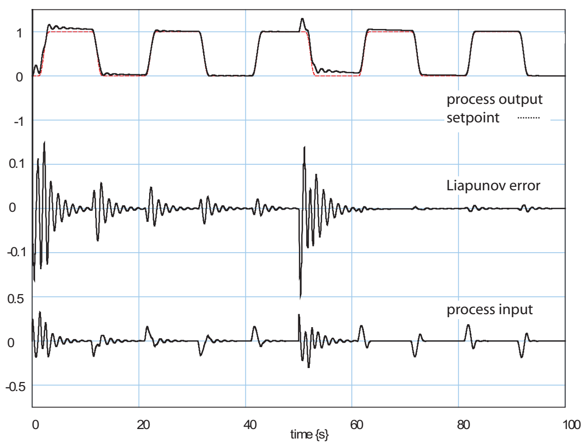

Fig. 9. Simulation results with a mechatronic system with disturbance at $t = 50$ [s]

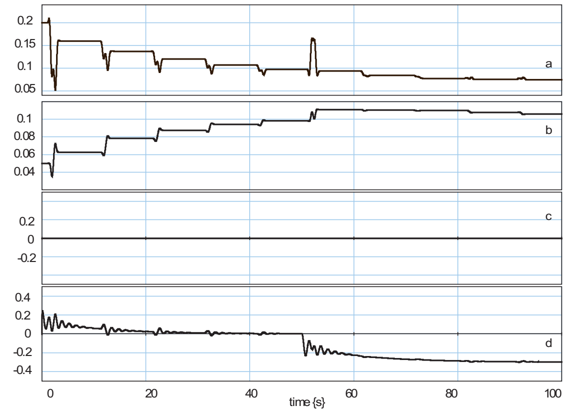

Fig. 10. Parameters of the mechatronic system

A fourth–order feed–forward controller (which implies that we estimate the resonant dynamics as well) gives the responses of figure 11.

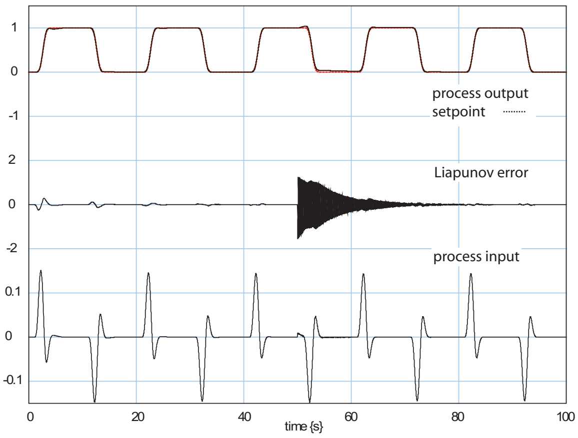

Fig. 11. Result of fourth–order feed–forward control

It can be seen that the fourth–order feed forward is able to suppress the vibrations by canceling the resonant poles with appropriate zero's. The resonances are only seen in the Liapunov error at the moment that the step–shaped disturbance is applied.

With the second-order controller similar results were obtained on the *real* system. This demonstrates once again the robustness of the proposed feed-forward controller.

5. CONCLUSIONS AND SUGGESTIONS

Inspired by feed–forward control structures like Iterative Learning Control and Learning Feed–Forward Control a new Model–Reference–based Learning Feed–Forward controller has been described, suitable for the control of linear systems, or systems that can be approximated as such. It is very robust. It is able to control a fourth–order system with resonant poles, by approximating it by a second–order model. Because for the control mainly the signals from a setpoint generator are used, noise on the measurements of the process has almost no influence on the system, leading to almost noise–free steering signals and unbiased parameter estimates. The main difference with other Model–Reference structures is, that inspired by the idea of making an inverse model, in addition to the signals x and $\dot{x}$ (or x_1 and x_2) also the signal x_3 is used in the adaptation and control of the parameters. Although this is nothing else than a combination of x_1 and x_2 and the input signal u, the resulting parametrization of the system seems to have a very beneficial effect on the performance of the adaptive system. Compared with Learning Feed–Forward Controllers the proposed method is efficient with respect to use of memory. It would be worthwhile to investigate if it can be combined with a neural–network like representation of non-linearities in the system.

REFERENCES

de Kruif, B.J. (2004). Function Approximation for Learning Control a Key Sample Based Approach. PhD thesis. University of Twente, Enschede, Netherlands.

de Kruif, B.J. and T.J.A. de Vries (2003). Pruning error minimization in least squares support vector machines. *IEEE Transactions on Neural Networks* **Vol. 14**(3), 696–702.

de Vries, T.J.A., W.J.R. Velthuis and J. van Amerongen (2000). Learning feed-forward control: A survey and historical note. In: *1st IFAC Conference on Mechatronic Systems, Darmstadt, Germany* (R. Isermann, Ed.). pp. 949–954.

Moore, K. L. (1998). Iterative learning control: An expository overview. *Applied and Computational Controls, Signal Processing, and Circuits*.

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.

Steinbuch, M. and M.J.G. van de Molengraft (2000). Iterative learning control of industrial motion systems. In: *1st IFAC Conference on Mechatronic Systems, Darmstadt, Germany* (R. Isermann, Ed.). pp. 967–972.

van Amerongen, J. (1984). Adaptive steering of ships - a model reference approach. *Automatica* **20**(1), 3–14.

van Amerongen, J. and P.C. Breedveld (2003). Modelling of physical systems for the design and control of mechatronic systems (IFAC professional brief). *Annual Reviews in Control* **27**, 87–117.

van Amerongen, J., P.G.M. van der Klugt and H.R. van Nauta Lemke (1990). Rudder roll stabilization for ships. *Automatica* **26**(4), 679–690.

Velthuis, W.J.R. (2000). Learning Feed-Forward Control - Theory, Design and Applications. PhD thesis. University of Twente, Enschede.

Verwoerd, Mark (2005). Iterative Learning Control: A Critical Review. PhD thesis. University of Twente.

www.20sim.com (2006).