A New Method for the Analysis of Neural Reference Model Control

Michael Wigbers
Neurotec Hochtechnologie
D- 88046 Friedrichshafen
mw@neurotec.de

Martin Riedmiller
University of Karlsruhe
Computer Science Dept.
D- 76128 Karlsruhe, Germany
riedml@ira.uka.de

Abstract

In recent years there has been much effort to develop the theoretical aspects of neural MRAC-Control, that is to find conditions under which an unknwon process can be identified by an input-output model and controllers can be trained by gradient descent ([3],[6]). On the other hand, the application of neural network techniques to real world control of nonlinear dynamical systems has been of substantial interest. Since the theoretical conditions that ensure controllability and the applicability of indirect adaptive control are hard to verify in practice, the success of controller training is mostly shown by testing relevant situations. We trained a controller for a subsystem of a spark ignition engine by dynamic backpropagation ([2]) and various truncated gradient algorithms. Afterwards we related the neural MRAC-approach to Pole placement and Linearization techniques in order to show the successful training by Pole analysis of the completely trained loop. This is a new method to verify the plausibility of the adaptation process and the trained regulator.

1 Introduction

The underlying idea of the neural gradient formalism for indirect adaptive control as established by Narendra and Parthasarathy in [2] is closely related to the sensitivity approach (MIT-Rule, [1]) proposed in the 1960's. Neural nets are well suited for adaptive control, since parameter adjustment of the neural model is performed totally data driven and allows comparatively precise approximation to the plant's behavior for each individual instance. In a preliminary phase of MRAC (model reference adaptive control) a neural model is trained to predict the behavior of the plant. In the main phase the parameters of the neural controller are adjusted by gradient descent, minimizing the difference between plant output and the parallel reference model. This step involves the deduction of the sensitivity functions, i. e. the Jacobian, by backpropagation applied to the model. The Jacobian can be regarded as a representation of the linearized plant dynamics and therefore can provide useful insight in the plant at equilibrium points by eigenvalue-analysis. This may facilitate the choice of a realistic control objective, specified by the reference model, that is crucial for the success of the controller training. Classical control of nonlinear systems is typically approached by linearizing the system at a nominal operating point and applying linear feedback control methods. When a system must be controlled over a wide range of operation, this process is repeated at several operating points. The 'Extended Linearization' ([5]) computes nonlinear state feedback gains, such that the eigenvalues of the linearized closed-loop system are placed at specified values that are invariant for all closed-loop operating points. In this paper we restrict to MRAC with feedforward neural networks as controllers trained by dynamic backpropagation or simplifications regarding the degree of recursion. We observed that simple training algorithms require more elaborate reference models. Narendra and Saerens have formulated conditions that can guarantee successful controller training. These are hard to verify when the plant is unknown. We focus on showing the success of the controller training by examining the poles of the completely trained closed loop.

'Extended Linearization' is meant as a controller design-procedure. We perform the same linearizations to apply an offline controller test. We evaluate the neural state feedback controller by testing whether the eigenvalues of the linearized closed loop have assumed the values that we specified by the reference model. Given a linear reference model the eigenvalues are expected to be invariant for all closed loop operating points. To demonstrate these aspects we mention the manifold subsystem of a spark-ignition engine with the manifold air pressure as output and the throttle plate angle as input. For this kind of plant Puskorius et al ([8]) of Ford research laboratory have recently demonstrated that successful indirect neural control is possible on a real world vehicle. Our approach is meant to strengthen the confidence in such controllers.

2 Neuro-MRAC (Model-Reference-Adaptive-Control)

The control objective is specified by a parallel reference model y_{ref}, that is to cancel the error at every time step k

$$E(k) = \frac{1}{2}(y(k) - y_{ref}(k))^2.$$

The feedforward net Pl of the NARMA type is assumed to approximate the plant and to be trained by a sufficient set of training samples that is generated by observing the plant's response y on suitable inputs u and initial states:

$$y(k) = Pl(y(k-1)\dots y(k-n), u(k-1)\dots u(k-n))$$

The appropriate Controller Ctl with weight-vector $\underline{w}$ and the desired plant output r is:

$$\begin{aligned} u(k-1) &= Ctl[r(k-1), y(k-1)\dots y(k-n), \\ &\quad u(k-2), \cdots, u(k-n); \underline{w}] \end{aligned}$$

The corresponding sensitivity model is given by 'dynamic backpropagation' [2]) and 'real time recurrent learning' ([7]). The summarized assumptions that have to be made ([3],[6]) to ensure the applicability of indirect adaptive control are the following: 1. the order n of the plant is known, 2. the system is observable, 3. the system is inverse stable (asymptoticaly minimum phase), 4. there exists a control sequence u that allows the desired tracking of the reference model. The recursion of dynamic backpropagation can be truncated at different levels m:

$$\frac{\partial u(k-m-1)}{\partial w} = \cdots = \frac{\partial u(0)}{\partial w} = 0$$
$$\frac{\partial y(k-m)}{\partial w} = \cdots = \frac{\partial y(0)}{\partial w} = 0$$

Static backpropagation $m = 1$ is the extreme simplification, that trains at time k only $u(k-1)$ and considers the previous control actions $u(k-2), \cdots$ and observations $y(k-1), \cdots$ as constant. This means, that assumption 4 has to be changed to: 4. there exists a control $u(k-1)$ that cancels the error $E(k) = 0$ at time k (given that $E(k-1) = 0$). In consequence a realistic reference model has to be designed carefully for the plant to allow tracking of the reference objective by adjusting only $u(k-1)$. In between these extremes there is n-step backpropagation ($m = n$) that we experienced to be more robust regarding the choice of the reference model. This learning method trains the n previous control actions and makes use of the corresponding n gradients that are directly provided by the model Pl.

3 Poleanalysis (linearization of the trained loop)

Linearizations of the plant are often used to design linear controllers that work locally around operating points (e. g. by Pole-placement). Our intention is to linearize a neural plant model and a closed-loop system after convergence of the training algorithm. We calculate the linearizations in order to apply an offline controller test. At every operating point we get a linear representation and its poles. The poles of a plant provide information regarding its dynamics and give useful hints where to specify the poles of the reference model in order to specify a realistic control objective. On the other hand the poles of a trained closed-loop system can be compared to the reference poles. If they match the reference poles the controller training reached the control objective.
The gradients of the model Pl evaluated at an operating point (y_0, u_0) and obtained by backpropagation give an ARMA-model of the plant dynamics at the operating point:

$$y(k+1) = \sum_{i=1}^{n} a_i y(k-i+1) + \sum_{i=1}^{n} b_i u(k-i+1)$$

$$a_1(Plant) = \frac{\partial Pl}{\partial y(k)} \dots a_n(Plant) = \frac{\partial Pl}{\partial y(k-n+1)}$$

$$b_1(Plant) = \frac{\partial Pl}{\partial u(k)} \ldots b_n(Plant) = \frac{\partial Pl}{\partial u(k-n+1)}$$

This linear difference equation can - in the case of observability - be transformed into the following state space representation (observability-form):

$$\underline{x}(k+1) = \underline{\Phi}\,\underline{x}(k) + \underline{h}u(k), \qquad (1)$$

$$\underline{\Phi} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 & a_n \\ 1 & 0 & \cdots & 0 & 0 & a_{n-1} \\ 0 & 1 & \cdots & 0 & 0 & a_{n-2} \\ \vdots & & \ddots & & & \vdots \\ 0 & \cdots & 0 & 1 & 0 & a_2 \\ 0 & \cdots & 0 & 0 & 1 & a_1 \end{pmatrix}, \underline{h} = \begin{pmatrix} b_{n-1} \\ b_{n-2} \\ \vdots \\ b_2 \\ b_1 \\ b_n \end{pmatrix} \qquad (2)$$

The transition to a time-continuous representation ($\underline{A}$) of the sampled system ($\underline{\Phi}$, sampling interval T) can be made by the following approximation:

$$\underline{\Phi} = e^{\underline{A}T} = \underline{I} + \underline{A}T + \frac{\underline{A}^2T^2}{2!} + \cdots$$

$$\rightarrow \underline{\Phi} \approx \underline{I} + \underline{A}T \rightarrow \underline{A} \approx \frac{1}{T}(\underline{\Phi} - \underline{I}) \qquad (3)$$

When using the coefficients $a_j(Plant)$ the eigenvalues of $\underline{A}_{Plant}$ can be regarded as the poles of the plant. In the same way we can get a time-continuous overview $\underline{A}_{Loop}$ of the completely trained closed loop (*Loop*) by applying the chain-rule to the combination of controller and plant that are connected by $u(k)$ (all gradients are obtained by backpropagation):

$$a_j(Loop) = \frac{\partial Loop}{\partial y(j)} = \frac{\partial Pl}{\partial y(j)} + \frac{\partial Pl}{\partial u(k)}\frac{\partial Ctl}{\partial y(j)}$$

4 Controlling a spark ignition engine's air flow

4.1 Manifold Model

The air flow together with the fuel flow constitute the input energy flow to a spark ignition engine. The throttle's task is to reduce the ambient air pressure (101300Pascal) and thus to regulate the amount of gas flowing into the cylinders. The block diagram of the air flow model in a throttle-manifold assembly is shown in figure 1 with the throttle angle [degrees] $u(k) = \alpha(k)$ (input), the manifold air pressure [Pascal] $y(k) = P_S(k)$ (output) and the crank shaft speed [rpm] $z(k) = n(k)$ (disturbance, here). In a first-step model we assume an ideal throttle to get a simple model of order one. The model is highly nonlinear and contains experimentally obtained data of the air mass flow past the cylinders.

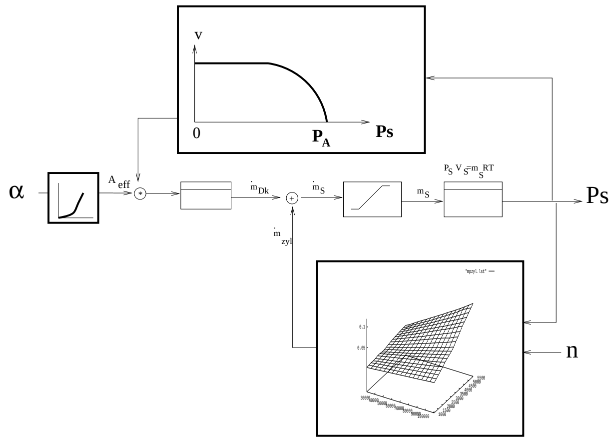


Figure 1. Blockdiagram of the first-order model

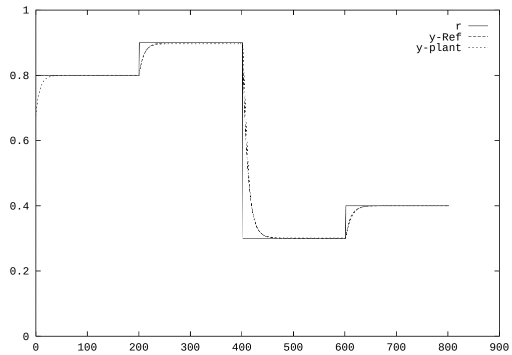
The throttle is taken as equivalent to a convergent nozzle of variable exit area A_{eff}, depending on the throttle's opening angle α ($\alpha = 0 degrees$ means closed throttle).

With this simulated engine we generated training data for the neural model Pl (training method was RPROP ([4]), a backpropagation-variant). The dynamic nonlinearity of the plant is shown in figure 3(a), that gives the plant's poles at different operating points P_{S0}. These poles (A_{Plant}) are computed by linearization of Pl ($\frac{\partial Pl}{\partial y(k)}$ evaluated at $P_{S0} = y_0$) as described earlier. At high pressures the plant becomes 'faster' as can be seen by observing the decreasing negative poles. In a second step we took into account the throttle's driving electrical actuator and the throttle's mass dynamics by modelling frictions and the influence of the air mass flow on the throttle's movement. The resulting fourth-order model was identified by a $n = 4$-model Pl.

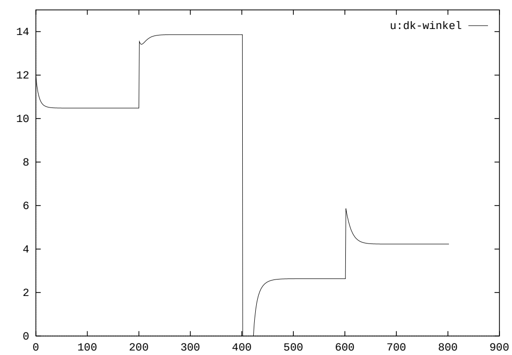
4.2 Controller-training und -test

The neural controller $\alpha(k) = Ctl_1(P_S(k), r(k))$ for the first order model is trained with a first order PT_1-reference model with pole λ . A test of the trained controller (figure 2) shows the loop's response $P_S(k)$ on changes in the desired manifold air pressure ($r(k)$ stays constant for 2 seconds at $80000, 90000, 30000, 40000$ Pascal, respectively) in comparison to the reference models ($\lambda = -10$)

behavior. The MRAC-adaptation leads to good correspondence between closed loop and reference model.



(a)



(b)

Figure 2. Test of the trained controller $\lambda = -10$ over time [1/200sec] (a) $Ps(t)[1/100.000Pa]$ (b) $u(t)[Degrees]$

The second part of figure 2 shows the throttle opening angle α over time as generated by the trained controller (ca. $10, 14, 2.5, 4$ at equilibrium points). The different necessary changes in α at high and low pressure to achieve the same change in P_{S0} ($10000 Pascal$) show the static nonlinearity as expected by every motor expert. The transition phases between equilibrium points show the dynamic nonlinearity of the plant as observed in figure 3 by examination of the plant's poles. The transition at high pressure is achieved by simply opening the throttle (plant is fast, poles are nearly the reference pole $\lambda = -10$) while at low pressure the plant has to be accelerated (plant's poles are > -10).

To obtain the same results with the fourth-order model of the plant

$$\begin{aligned} P_S(k) &= Pl(P_S(k-1), \cdots, P_S(k-4), \\ &\quad \alpha(k-1), \cdots, \alpha(k-4)) \end{aligned}$$

we trained the controller Ctl_4:

$$\begin{aligned} \alpha(k) &= Ctl_4[r(k), P_S(k), \cdots, P_S(k-3), \\ &\quad \alpha(k-1), \cdots, \alpha(k-3)]. \end{aligned}$$

We experienced that in combination with a simple PT_1 reference model static backpropagation was not able to establish a controller. In this case there exists no suitable $\alpha(k-1)$ to cancel the error at time k as required because the plant's behavior depends largely on the earlier inputs $\alpha(k-2), \cdots, \alpha(k-4)$. This causes the necessity to use either a PT_4 reference model of order four with appropriate time constants or a first order PT_1 reference model and 4- or more- step backpropagation learning algorithm. The 4-step backpropagation makes use of the sensitivity information regarding $\alpha(k-1), \cdots, \alpha(k-4)$ that is provided directly by the plant model Pl.

4.3 Poleanalysis of the closed loop

We used online tests as above in order to show that the conroller works in a satisfying fashion. For linear plants controllability is a global property. For nonlinear systems the conditions for global controllability (assumption 4) are very hard to verify and one has to resort to local concepts. The local controllability of the nonlinear system around an equilibrium point can be shown by examining its linearization. In this section we try to apply this philosophy for our offline controller test. Each trained controller is tested for the resulting closed loop poles (eigenvalues of A_{Loop}) to match the given referencepoles (λ for the first order model). This was done for the operating points that are relevant in practice (P_{S0} between 20000 and 90000Pascal). In comparison to the original plant poles (figure 3(a)), figure 3(b) shows the poles of a trained loop.

Figure 4 shows the poles of the fourth-order loop after training with a PT_1 reference model ($\lambda = -8$) and 4-step backpropagation. The marked poles around ($Re = -8, Im = 0$) are the loop's dominant poles and correspond to the reference model's pole for the operating points. We observed that the 4-step learning was able to move the dominant poles of the loop towards λ over a wide region.

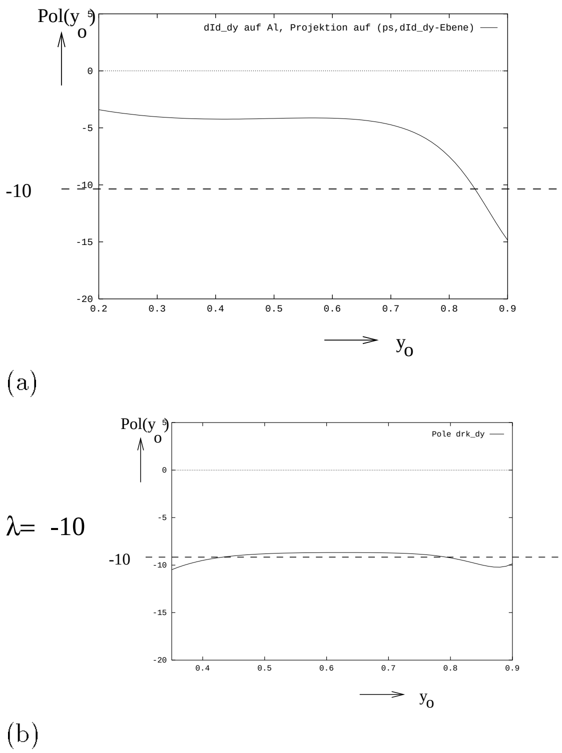


Figure 3. (a) Original plant poles on operating points $y_0 = P_{S0}$ [1/100.000Pa] (b) Poles of the loop after MRAC training, $\lambda = -10$

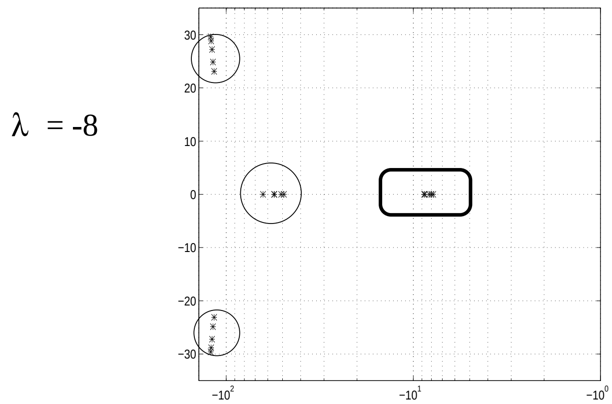


Figure 4. Poles of the 4-order loop at operating points, $\lambda = -8$

5 Conclusion

The Pole-examination ensures that the trained controller is able to work at least as good as a controller that is designed by classical methods, which place poles analytically and therefore depend on an analytic description of the plant. At all relevant operating points the loop's behavior is known to be stable and to behave like specified by the reference model. This method does not provide complete knowledge about the loop but gives another argument to demonstrate the success of controller training.

References

[1] Landau: *Adaptive Control, The Model Reference Approach*, Marcel Dekker Inc., Control and Systems Theory, Vol. 8, 1979

[2] Narendra, Parthasarathy: *Neural Networks and Dynamical Systems*, International Journal of Approximate Reasoning, 6, 1992, New York

[3] Narendra, Mukhopadhyay: *Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks*, Neural Networks, Vol.7, No. 5, 1994

[4] Riedmiller, Braun: *A direct adaptive method for faster backpropagation learning: The RPROP algorithm*, Proceedings of the IEEE International Conference on Neural Networks (ICNN), pages 586-591, San Francisco, 1993

[5] Baumann, Rugh: *Feedback Control of Nonlinear Systems by Extended Linearization*, IEEE Transactions on Automatic Control, Vol. AC-31, No. 1, January 1986

[6] Saerens, Renders, Bersini: *Neural Controllers Based on Backpropagation Algorithm*, IEEE Press Book on Intelligent Control, IEEE Press, 1994

[7] Williams and Zipser: *Experimental analysis of the real-time recurrent learning algorithm*, Connection Science, No. 1, 1989

[8] Puskorius, Feldkamp, Davis: *Dynamic Neural Network Methods Applied to On-Vehicle Idle Speed Control*, IEEE Transactions on Neural Nets, No. 5, 1996