Feedback linearization Feedforward control Optimal Control

Antti Yli-Krekola

16.11.2005

Feedback linearization

- Assume the output of the system can be expressed as :

$$y(t) = f(\varphi_f(t)) + g(\varphi_g(t))u(t-1)$$

where the regressors φ_f and φ_g are of the ARX form, and the latest input is $u(t-2)$.

Feedback linearization

- Assume the output of the system can be expressed as :

$$y(t) = f(\varphi_f(t)) + g(\varphi_g(t))u(t-1)$$

where the regressors φ_f and φ_g are of the ARX form, and the latest input is *u(t-2).*

- Introduce a new input signal w(t), which determines u(t) :

$$u(t-1) = \frac{w(t-1) - f[\varphi_f(t)]}{g[\varphi_g(t)]}$$

Feedback linearization

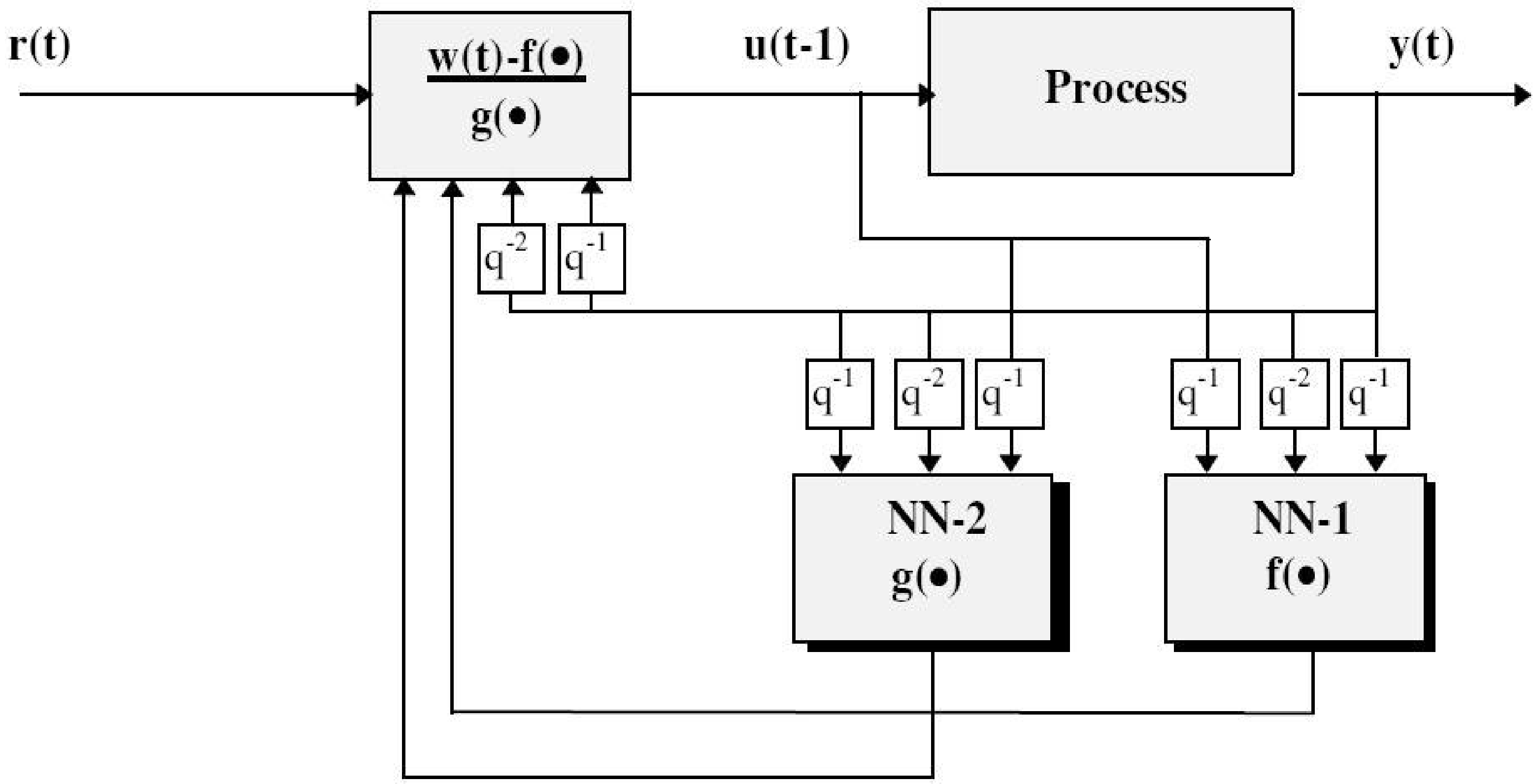
- This kind of dead-beat controller, *y(t) = w(t-1)*, is sensitive to model inaccuracies and noise.
- A more general method is to transform the nonlinear system to behave linearly as *:*

$$y(t) = H(q^{-1})w(t-1) \Leftrightarrow A(q^{-1})y(t) = B(q^{-1})w(t-1)$$

In this case:

$$u(t-1) = \frac{B(q^{-1})w(t-1) - \tilde{A}(q^{-1})y(t) - f[\varphi_f(t)]}{g[\varphi_g(t)]}$$

Feedback linearization



Feedback linearization

How to train the networks f and g ?

It is quite straightforward. For example, in the case of on-line gradient descent the gradients are formed as:

$$\frac{\partial e^2(t)}{\partial \theta} = \begin{bmatrix} \dfrac{\partial f(t \mid \theta_f)}{\partial \theta_f} \\ \dfrac{\partial g(t \mid \theta_g)}{\partial \theta_g} u(t-1) \end{bmatrix} * e(t) \qquad \theta = \begin{bmatrix} \theta_f \\ \theta_g \end{bmatrix}$$

Feedback linearization

- Advantages
  - Can make the closed loop system to follow a prescribed transfer function
  - Easy to implement

- Disadvantages
  - The class of systems obeying the rule y(t) = f() + g()*u(t-1) is restricted. It is not easy to determine whether a certain system belongs to this class.
  - Two NNs have to be trained
  - Problems with systems with unstable inverse.

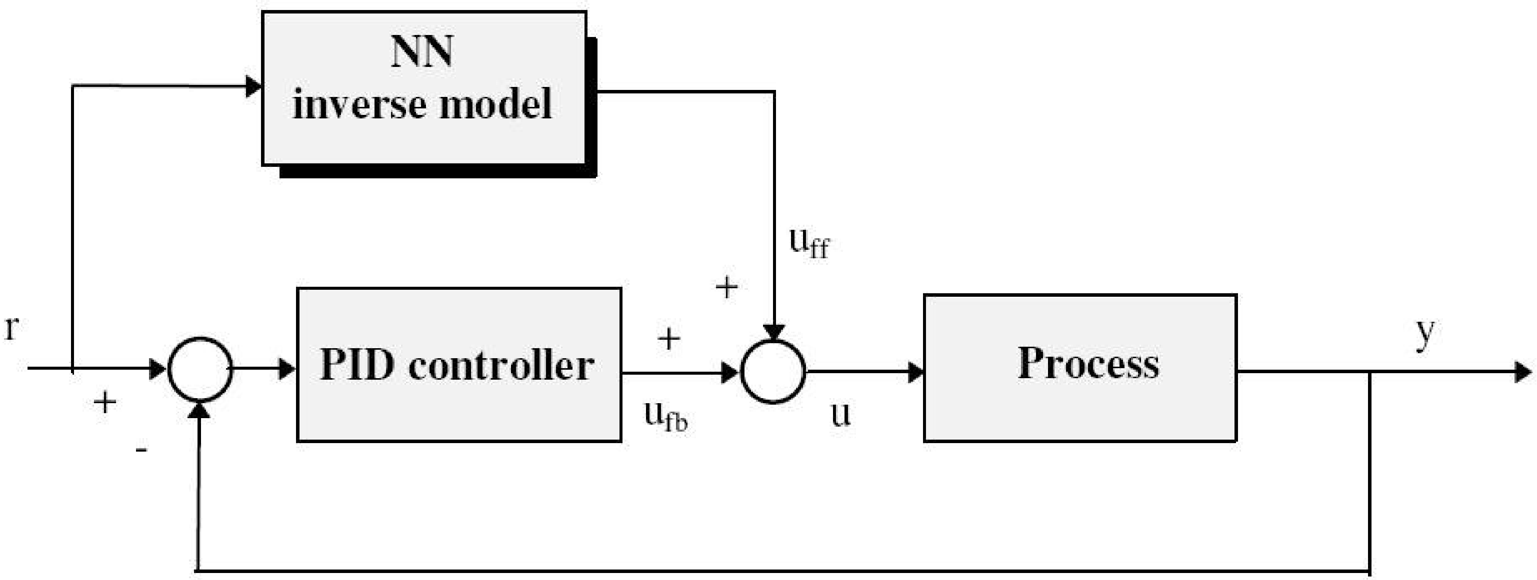
Feedforward control

- Main reasons for using feedback are:
  1. to stabilize an unstable system
  2. to compensate for disturbances and model inaccuracies
- Using feedback for rapid reference signal tracking makes the system sensitive to noise

- Feedforward control suits better for tracking rapid changes in the reference.

Feedforward control

- The feedforward part of the controller is a NN inverse model and has the reference as its only input.



Feedforward control

- Static feedforward:
  - The inverse model is NNFIR; it depends only on past values of the reference signal.
  - The feedback part (e.g. PID) of the controller compensates for model inaccuracies.
  - No instability issues.

- Dynamic feedforward:
  - The inverse model is NNOE; it is a function of past reference signal and feedforward control signal values. Feedback from the real system is avoided.
  - Difficult to resolve whether this inverse model is stable.

Feedforward control

- Steady-state feedforward
  - Inverse model is trained with different steady states.
  - Feedforward produces the steady part of the control signal, and feedback controller (e.g. PID) handles the disturbances.

=> this controller is just a scalar function of a scalar variable.

  - Cannot be used with unstable systems, because then the steady part of the control signal would be zero.

Feedforward control

- Advantages
  - Feedforward control signal can be introduced gradually to improve existing controller.
  - Improves rapid reference tracking.

- Disadvantages
  - Does not reduce the effect of disturbances acting on the system.
  - This is still a direct inverse controller. Problems with unstable inverse.

Optimal control

- Optimal in the sense, that behaviour of the control signal is also taken into account, not only of the output.
- Active control signal poses problems.

- The initial control method, which minimizes the criterion J, is turned into optimal when introducing penalty for control signal activity:

$$\mathrm{J}_{\mathrm{OPT}} = \mathrm{J} + \rho * A(u), \qquad \rho \geq 0$$

Optimal control

- If the activity A(u) is sum of squares of the input u :

$$J(\theta) = \sum_t \{[r(t) - y(t)]^2 + \rho u^2(t)\}$$

$$\frac{\partial J(\theta)}{\partial \theta} \cong \frac{\partial u(t-1)}{\partial \theta}\left[\frac{\partial \hat{y}(t)}{\partial u(t-1)} e(t) + \rho u(t-1)\right]$$

Experience has shown that it is not so crucial how the Hessian update is modified from the basic specialized training.

Optimal control

- Optimal control does not make the closed loop system to behave as a linear transfer function. In that way it is different to the earlier methods.

  Because of non-linearity, the resultant controller has to be tested on the whole output domain.

Optimal control

- Advantages
  - Reduces control signal activity.
  - Applicable to a large set of systems.
  - Can be trained for a specific reference trajectory.

- Disadvantages
  - Difficult to tune. Must be retrained each time the penalty factor is changed.
  - With specialized training, must be trained online.