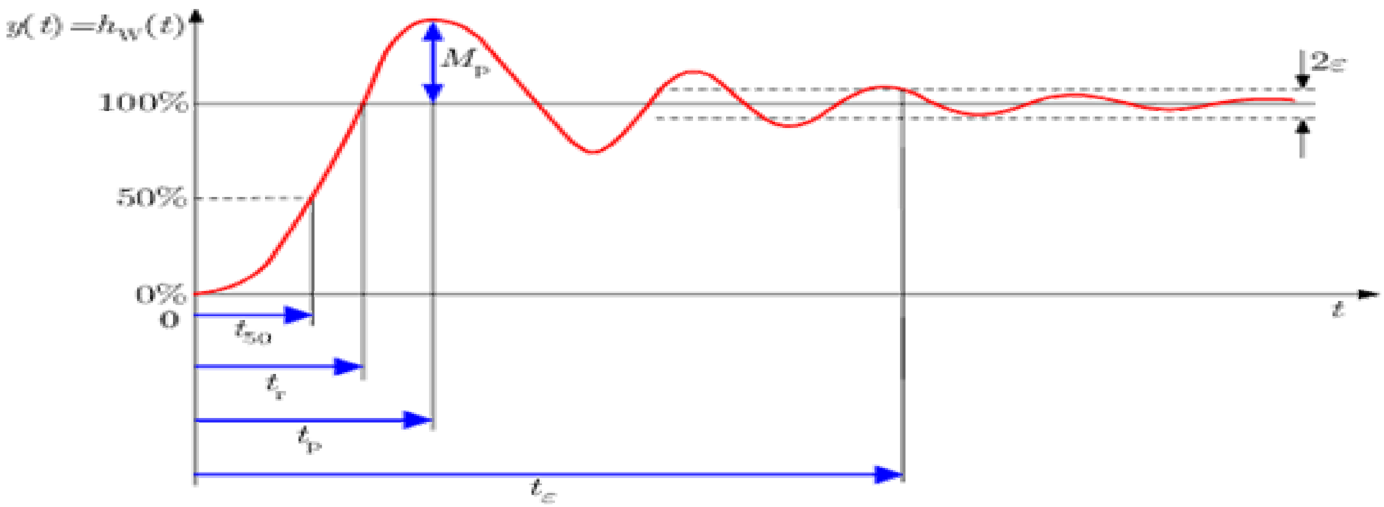
Introduction to Control
Direct Inverse Control

Mikael Pohjola

Introduction to Control

- Make the system behave in a desired manner
- Control the response
  - Speed
  - Overshoot, rise-time...
  - Disturbance rejection
- Depends on the system: actuators, process, measurements
- Result should be stable



Design

- Regulator problem
  - Keep output constant
    - E.g. room temperature
  - Disturbance rejection
- Servo problem
  - Output follows a trajectory
    - E.g. controlling a robot
  - Close tracking

Design goals

- Closed loop follow a prescribed transfer function
  - Cancellation of non-linearities
  - Resulting closed-loop transfer function linear
    - Pole placement
    - Model Reference Controller
- Minimize quadratic cost function
  - Closed loop non-linear
  - Intuitive
    - Minimum variance
    - Predictive control
    - Optimal control

Control Strategy Design

- Direct design
  - NN the controller
  - Implementation simple
  - Tuning difficult
  - Usually model based

    - Direct inverse control
    - Internal model control
    - Feedback linearization
    - Feedforward with inverse models
    - Optimal control

- Indirect design
  - NN model of process
  - Always model based
  - Controller designed online
  - Flexible

    - Approximate pole placement
    - Minimum variance
    - Predictive control

Neural Networks in control

- Non-linear systems
  - When linear can not be used
  - More difficult
  - Adaptive control, online tuning
- No notion of zeros or poles
  - Stability not guaranteed
    - Limited class of systems, accurate mode, known inaccuracies...
  - Can be locally stable/unstable
- NN too flexible in some cases
  - E.g. adaptive control of time-variant systems

Direct inverse control

- System

$$y(t+1) = g\big(y(t),\ldots,y(t-n+1),u(t),\ldots,u(t-m)\big)$$

- Controller: inverse of the system

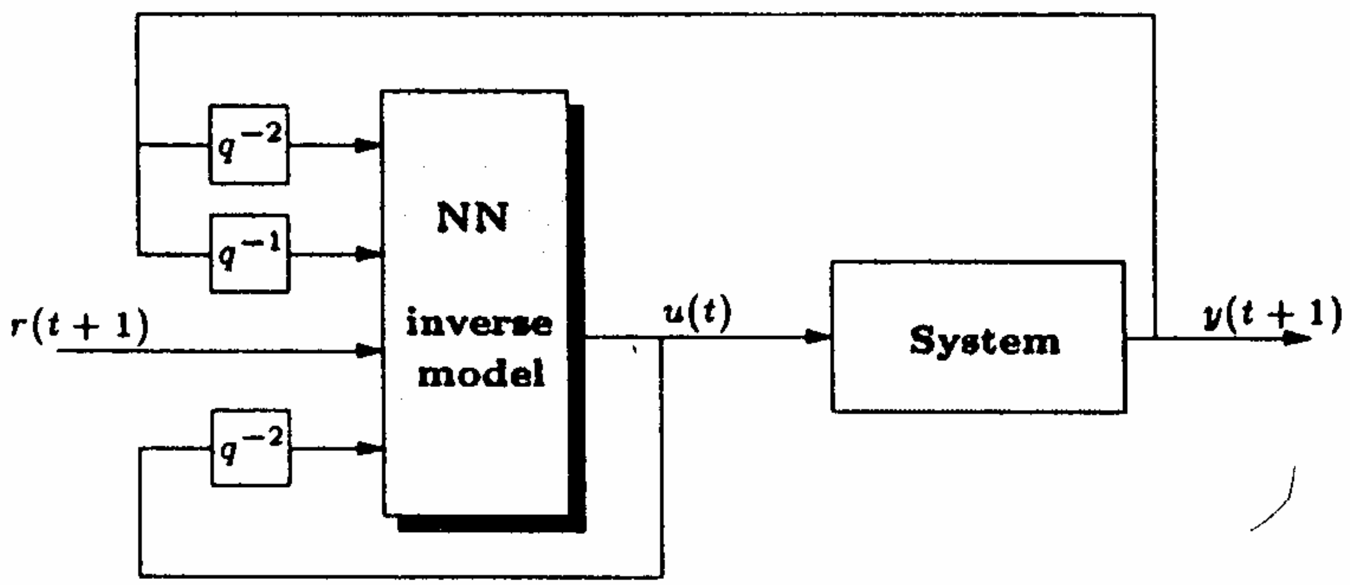
$$\hat{u}(t) = \hat{g}^{-1}\big(y(t+1), y(t),\ldots,y(t-n+1),u(t-1),\ldots,u(t-m)\big)$$

  - Isolates the most recent control input
  - Replace y(t+1) by the desired output r(t+1)

$$u(t) = \hat{g}^{-1}\big(r(t+1), y(t),\ldots, y(t-n+1), u(t-1),\ldots,u(t-m)\big)$$

- If inverse exact, the output is the reference
  - dead-beat controller

Direct inverse control



Process with delay

- Delay, d

$$y(t+d) = g\big(y(t+d-1),\ldots,y(t+d-n),u(t),\ldots,u(t-m)\big)$$

- Controller

$$\hat{u}(t) = \hat{g}^{-1}\big(r(t+d),y(t+d-1)\ldots,y(t+d-n),u(t-1),\ldots,u(t-m)\big)$$

- Unknown

$$y(t+1)\ldots y(t+d-1)$$

- Solution: prediction

The inverse model

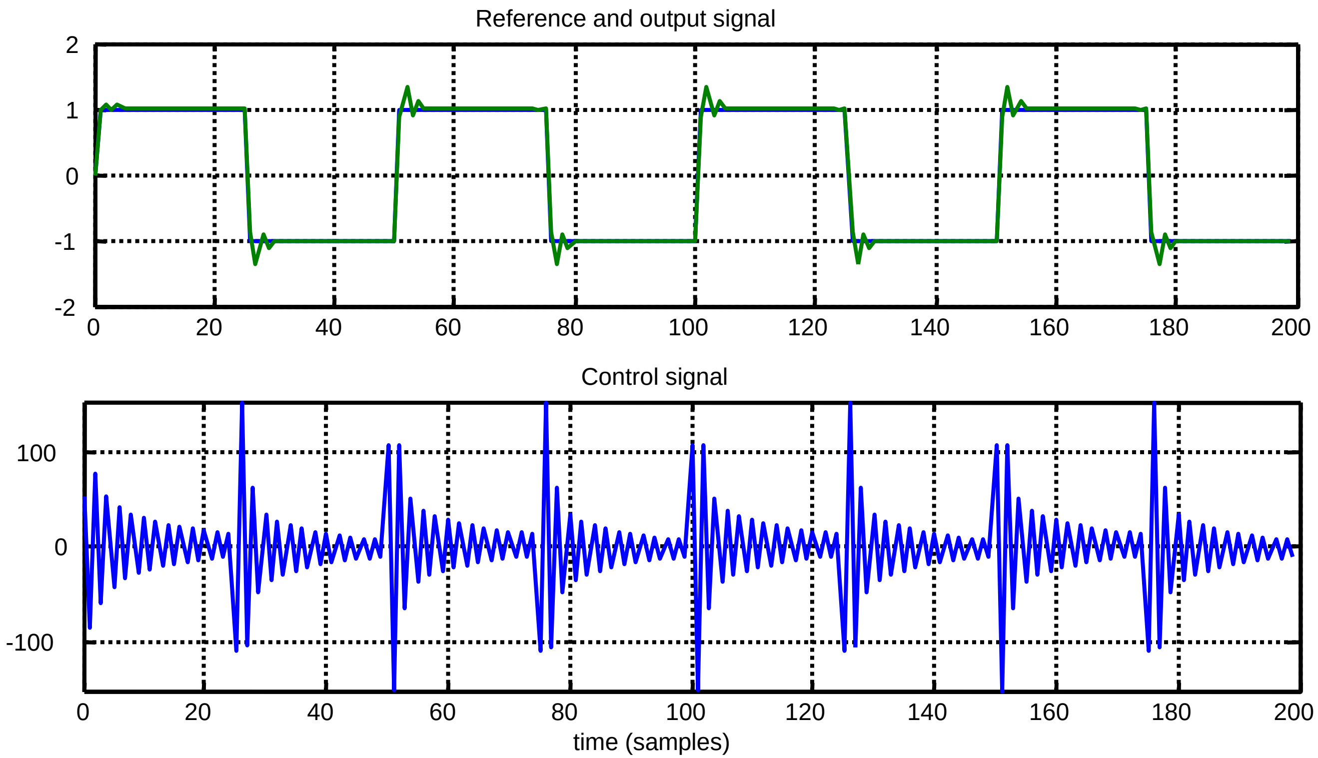
- System identification in the inverse direction: y→u
- Offline: general training
- Online: special training
- Training inverse model with realistic signals
  - Iterative offline training

General training

- Minimize: $J(\theta, Z^N) = \frac{1}{2N}\sum_{t=1}^{N}\left(u(t) - \hat{u}(t|\theta)\right)^2$
- Back propagation, Levenberg-Marquardt
- Earlier lectures: tools to do non-linear control

Benchmark system

$$\ddot{y}(t) + \dot{y}(t) + y(t) + y^3(t) = u(t)$$



Issues

- Sensitive to noise
- Poor robustness
- Large control signals
- Inverse model unstable or does not exist

Conclusion

- Introduction to control
  - Regulator/Servo-problem
  - Direct/Indirect control
  - Design goals
- Direct Inverse Control
  - NN controller inverse of system
  - General training