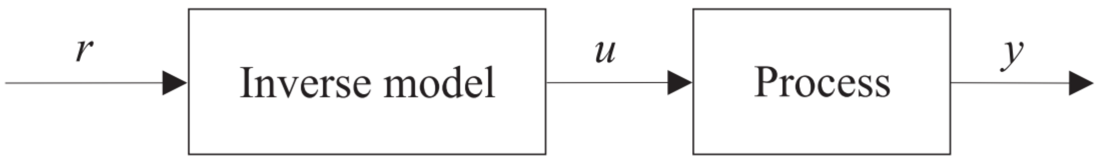
02_02_Model-Based Control

1. Inverse Control

The objective of inverse control is to compute for the current state $x(k)$ the control input $u(k)$, such that the system's output at the next sampling instant is **equal to the desired (reference) output** $r(k+1)$.

$$\text{system model: } y(k+1) = f(\mathbf{x}(k), u(k))$$
$$\text{contro signal: } u(k) = f^{-1}(\mathbf{x}(k), r(k+1))$$



Applicable condition

1. Process (model) is stable and invertible
2. The inverse model is stable

$$G(s) = \frac{B(s)}{A(s)}, \text{the right side may has root, when inversed, it becom unstable}$$

3. Process model is accurate (enough)
4. Little influence of disturbances

Two Classical Ways

Open-Loop Feedforward Control

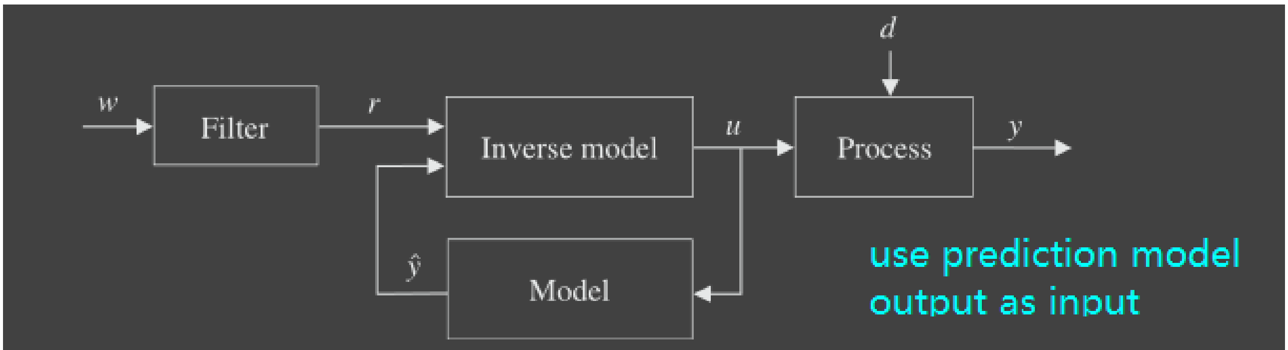


Figure 9.1.: Open-loop feedforward inverse control.

- Stable control is guaranteed for open-loop stable, minimum-phase system
- a model-plant mismatch or a disturbance d will cause a **steady-state error** at the process output

Open-Loop Feedback Control

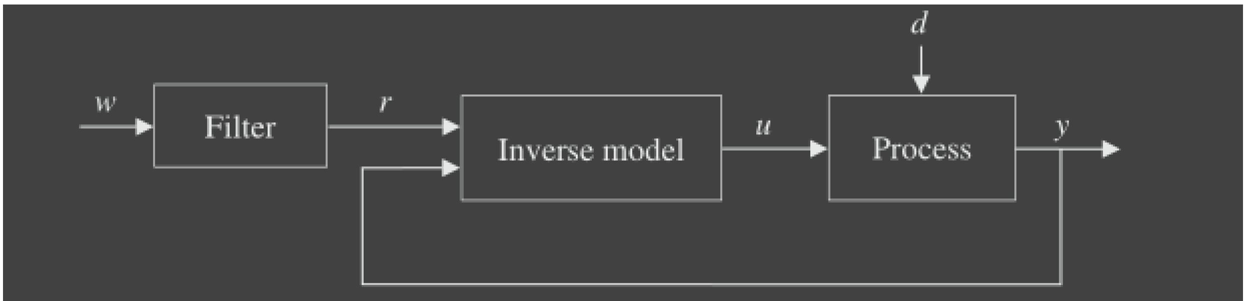


Figure 9.2.: Open-loop feedback inverse control.

- improve the prediction accuracy and eliminate offsets
- in the presence of noise or a significant model–plant mismatch, in which cases it can cause oscillations or instability.

Methods of Inverse $f(.)$

Numerically Method

$$J(u(k)) = (r(k+1) - f(\mathbf{x}(k), u(k)))^2$$

The minimization of J with respect to $u(k)$ gives the control corresponding to the inverse function

Affine TS Model

Consider the following fuzzy process model

$$\begin{aligned}
\mathcal{R}_i: \quad & \textbf{If } y(k) \text{ is } A_{i1} \textbf{ and } \ldots \textbf{ and } y(k-n_y+1) \text{ is } A_{in_y} \textbf{ and} \\
& u(k-1) \text{ is } B_{i2} \textbf{ and } \ldots \textbf{ and } u(k-n_u+1) \text{ is } B_{in_u} \textbf{ then} \\
& y_i(k+1) = \sum_{j=1}^{n_y} a_{ij} y(k-j+1) + \sum_{j=1}^{n_u} b_{ij} u(k-j+1) + c_i, \qquad (9.4)
\end{aligned}$$

So, the state space should have:

$$\mathbf{x}(k) = [y(k), y(k-1), \ldots, y(k-n_y+1), u(k-1), \ldots, u(k-n_u+1)]$$

And the output $y(k+1)$ should be:

$$y(k+1) = \frac{\sum_{i=1}^{K} \beta_i(\mathbf{x}(k)) y_i(k+1)}{\sum_{i=1}^{K} \beta_i(\mathbf{x}(k))}$$

where

$$\begin{aligned}
\beta_i(\mathbf{x}(k)) = & \mu_{A_{i1}}(y(k)) \wedge \ldots \wedge \mu_{A_{iny}}(y(k-n_y+1)) \wedge \\
& \mu_{B_{i2}}(u(k-1)) \wedge \ldots \wedge \mu_{B_{inu}}(u(k-n_u+1)).
\end{aligned}$$

We can rewrite it as:

$$\begin{aligned}
y(k+1) = & \sum_{i=1}^{K} \lambda_i(\mathbf{x}(k)) \left[\sum_{j=1}^{n_y} a_{ij} y(k-j+1) + \sum_{j=2}^{n_u} b_{ij} u(k-j+1) + c_i \right] + \\
& + \sum_{i=1}^{K} \lambda_i(\mathbf{x}(k)) b_{i1} u(k)
\end{aligned}$$

where

$$\lambda_i(\mathbf{x}(k)) = \frac{\beta_i(\mathbf{x}(k))}{\sum_{j=1}^{K} \beta_j(\mathbf{x}(k))}$$

So, now we have:

$$y(k+1) = g(\mathbf{x}(k)) + h(\mathbf{x}(k))u(k)$$

so we can calculate:

$$u(k) = \frac{r(k+1) - g(\mathbf{x}(k))}{h(\mathbf{x}(k))}$$

and it will be:

$$u(k) = \frac{r(k+1) - \sum_{i=1}^{K} \lambda_i(\mathbf{x}(k)) \left[\sum_{j=1}^{n_y} a_{ij} y(k-j+1) + \sum_{j=2}^{n_u} b_{ij} u(k-j+1) + c_i\right]}{\sum_{i=1}^{K} \lambda_i(\mathbf{x}(k)) b_{i1}}.$$

Singleton Model

$$\begin{aligned}&\textbf{If } y(k) \text{ is } A_1 \textbf{ and } y(k-1) \text{ is } A_2 \textbf{ and } \dots \textbf{ and } y(k-n_y+1) \text{ is } A_{n_y} \\ &\textbf{and } u(k) \text{ is } B_1 \textbf{ and } \dots \textbf{ and } u(k-n_u+1) \text{ is } B_{n_u} \\ &\textbf{then } y(k+1) \text{ is } c,\end{aligned} \tag{9.13}$$

First, simplify the rule base to:

$$\textbf{If } \mathbf{x}(k) \text{ is } X \textbf{ and } u(k) \text{ is } B \textbf{ then } y(k+1) \text{ is } c.$$

	$u(k)$			
$\mathbf{x}(k)$	B_1	B_2	$\dots$	B_N
X_1	c_{11}	c_{12}	$\dots$	c_{1N}
X_2	c_{21}	c_{22}	$\dots$	c_{2N}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
X_M	c_{M1}	c_{M2}	$\dots$	c_{MN}

Assume we use the product t-norm for "and":

$$\beta_{ij}(k) = \mu_{X_i}(\mathbf{x}(k)) \cdot \mu_{B_j}(u(k))$$

$$\begin{aligned} y(k+1) &= \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \beta_{ij}(k) \cdot c_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N} \beta_{ij}(k)} = \\ &= \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \mu_{X_i}(\mathbf{x}(k)) \cdot \mu_{B_j}(u(k)) \cdot c_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N} \mu_{X_i}(\mathbf{x}(k)) \cdot \mu_{B_j}(u(k))} \end{aligned}$$

Then we keep simplify the output equation

$$\begin{aligned} y(k+1) &= \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \mu_{X_i}(\mathbf{x}(k)) \cdot \mu_{B_j}(u(k)) \cdot c_{ij}}{\sum_{i=1}^{M}\sum_{j=1}^{N} \mu_{X_i}(\mathbf{x}(k))\mu_{B_j}(u(k))} \\ &= \sum_{i=1}^{M}\sum_{j=1}^{N} \lambda_i(\mathbf{x}(k)) \cdot \mu_{B_j}(u(k)) \cdot c_{ij} \\ &= \sum_{j=1}^{N} \mu_{B_j}(u(k)) \sum_{i=1}^{M} \lambda_i(\mathbf{x}(k)) \cdot c_{ij} \end{aligned}$$

where $\lambda_i(\mathbf{x}(k)) = \frac{\mu_{X_i}(\mathbf{x}(k))}{\sum_{j=1}^{K} \mu_{X_j}(\mathbf{x}(k))}$

For **invertible**, The inversion method requires that the **antecedent membership** functions $\mu_{B_j}(u(k))$ are **triangular** and form a partition $\sum_{j=1}^{N} \mu_{B_j}(u(k)) = 1$ (Invertible condition 1)

we have

$$y(k+1) = \sum_{j=1}^{N} \mu_{B_j}(u(k))c_j$$

where

$$c_j = \sum_{i=1}^{M} \lambda_i(\mathbf{x}(k)) \cdot c_{ij}$$

Then we have some rules like:

$$\textbf{If } u(k) \text{ is } B_j \quad \textbf{then } y(k+1) \text{ is } c_j(k), \qquad j = 1, \ldots, N\,.$$

Then we will invert it to:

$$\textbf{If } r(k+1) \text{ is } c_j(k) \quad \textbf{then } u(k) \text{ is } B_j \qquad j = 1, \ldots, N\,.$$

Because it is singleton, we need interpolate

$$\begin{aligned} \mu_{C_1}(r) &= \max\left(0, \min\left(1, \tfrac{c_2 - r}{c_2 - c_1}\right)\right) \\ \mu_{C_j}(r) &= \max\left(0, \min\left(\tfrac{r - c_{j-1}}{c_j - c_{j-1}}, \tfrac{c_{j+1} - r}{c_{j+1} - c_j}\right)\right), \quad 1 < j < N, \\ \mu_{C_N}(r) &= \max\left(0, \min\left(\tfrac{r - c_{N-1}}{c_N - c_{N-1}}, 1\right)\right) \end{aligned}$$

Then we can calculate the output:

$$u(k) = \sum_{j=1}^{N} \mu_{C_j}(r(k+1))b_j$$

$$y(k+1) = f_x(u(k)) = f_x\left(f_x^{-1}(r(k+1))\right) = r(k+1)$$

Notes

A invertible singleton model must with monotonous model.

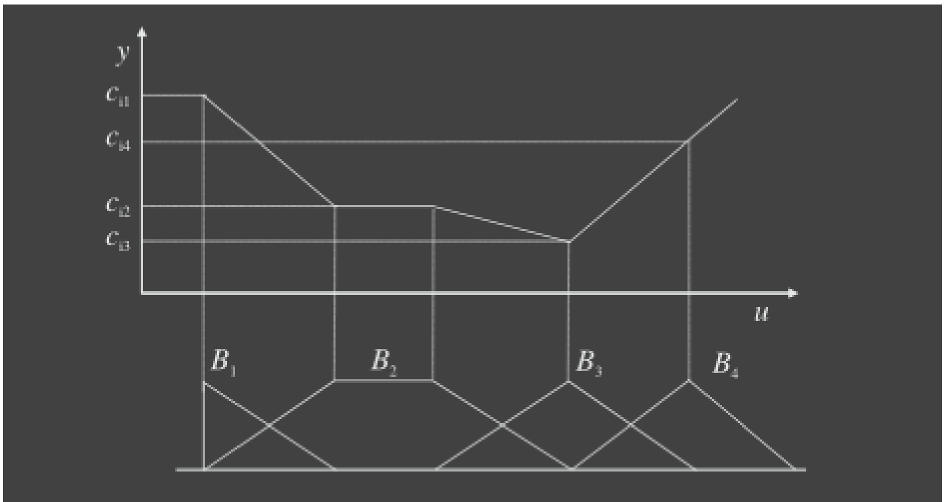


Figure 9.4.: Example of a noninvertible singleton model.

2. Internal Model Control

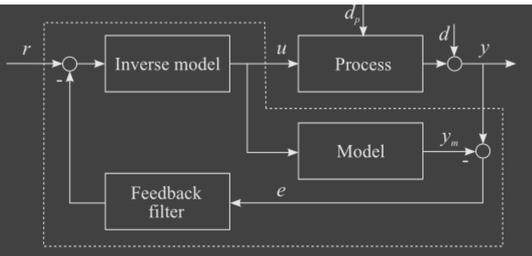


Figure 9.6.: Internal model control scheme.

The **purpose of the process model working in parallel** with the process is to **subtract the effect of the control action from the process output**.

- If the predicted and the measured process outputs **are equal**, the error e is zero and the controller works in an **open-loop configuration**.
  - If a **disturbance d** acts on the process output, the feedback signal e is equal to the influence of the disturbance and is not affected by the effects of the control action.

- This signal is subtracted from the reference. With a perfect process model, the IMC scheme
is hence able to **cancel the effect of unmeasured output-additive disturbances.**