

Model-Based Control

Model-Based Control

1. Inverse Control

1.1. Applicable condition

1.2. Two Classical Ways

Open-Loop Feedforward Control

Open-Loop Feedback Control

1.3. Methods of Inverse $f(\cdot)$

Numerically Method

Affine TS Model

Singleton Model

2. Internal Model Control

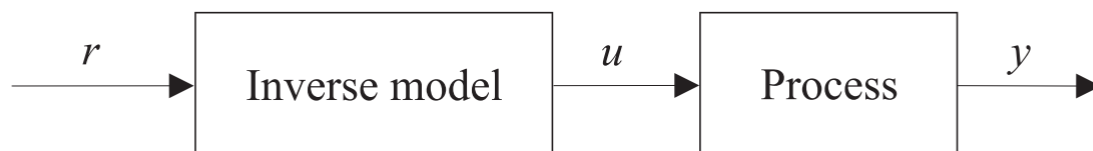
3. Model-Based Predictive Control

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** $r(k+1)$.

$$\text{system model: } y(k+1) = f(\mathbf{x}(k), u(k)) \quad (1)$$

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



1.1. Applicable condition

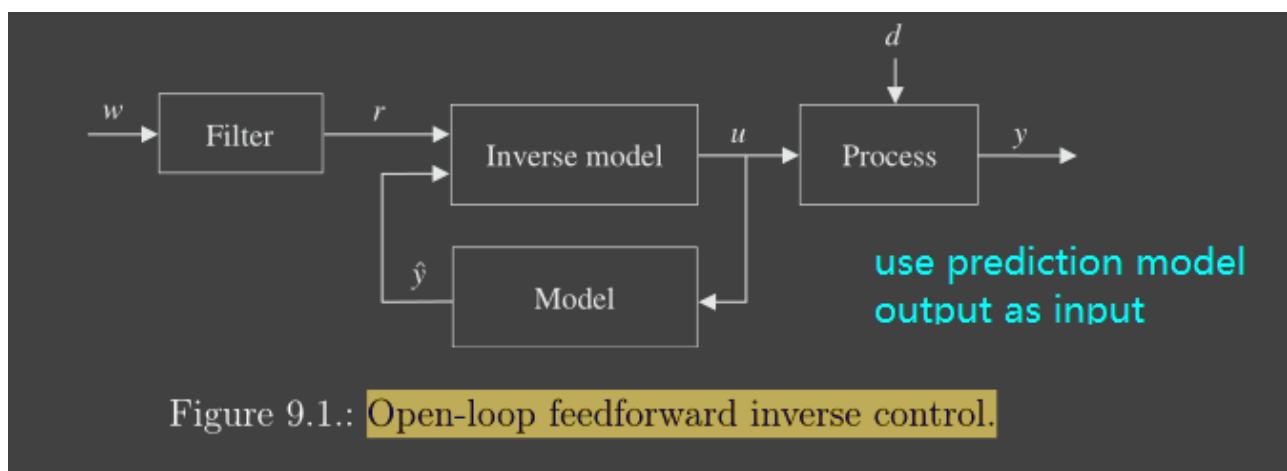
1. Process (model) is stable and invertible
2. The inverse model is stable

$$G(s) = \frac{B(s)}{A(s)}, \text{ the right side may have roots, when inverted, it becomes unstable} \quad (2)$$

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

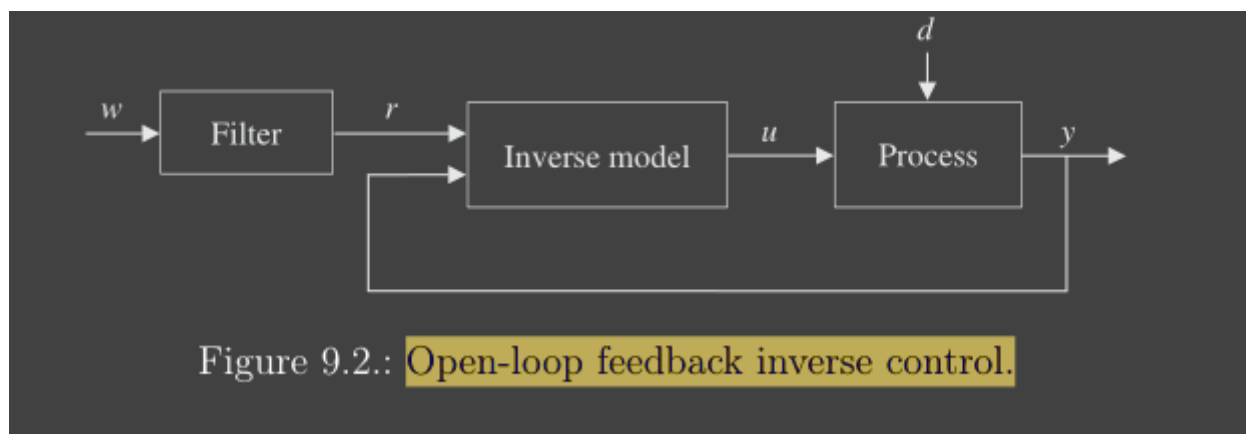
1.2. Two Classical Ways

Open-Loop Feedforward 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



- 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.

1.3. Methods of Inverse $f(\cdot)$

Numerically Method

$$J(u(k)) = (r(k+1) - f(\mathbf{x}(k), u(k)))^2 \quad (3)$$

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

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

So, the state space should have:

$$\mathbf{x}(k) = [y(k), y(k-1), \dots, y(k - n_y + 1), u(k-1), \dots, u(k - n_u + 1)] \quad (4)$$

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

$$y(k+1) = \frac{\sum_{i=1}^K \beta_i(\mathbf{x}(k)) y_i(k+1)}{\sum_{i=1}^K \beta_i(\mathbf{x}(k))} \quad (5)$$

where

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

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

$$\lambda_i(\mathbf{x}(k)) = \frac{\beta_i(\mathbf{x}(k))}{\sum_{j=1}^K \beta_j(\mathbf{x}(k))} \quad (6)$$

So, now we have:

$$y(k+1) = g(\mathbf{x}(k)) + h(\mathbf{x}(k))u(k) \quad (7)$$

so we can calculate:

$$u(k) = \frac{r(k+1) - g(\mathbf{x}(k))}{h(\mathbf{x}(k))} \quad (8)$$

and it will be:

$$u(k) = \frac{r(k+1) - \sum_{i=1}^K \lambda_i(\mathbf{x}(k)) \left[\sum_{j=1}^{n_y} a_{ij}y(k-j+1) + \sum_{j=2}^{n_u} b_{ij}u(k-j+1) + c_i \right]}{\sum_{i=1}^K \lambda_i(\mathbf{x}(k))b_{i1}}. \quad (9)$$

Singleton Model

If $y(k)$ is A_1 and $y(k-1)$ is A_2 and ... and $y(k-n_y+1)$ is A_{n_y}
 and $u(k)$ is B_1 and ... and $u(k-n_u+1)$ is B_{n_u}
 then $y(k+1)$ is c , (9.13)

First, simplify the rule base to:

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

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

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

$$\beta_{ij}(k) = \mu_{X_i}(\mathbf{x}(k)) \cdot \mu_{B_j}(u(k)) \quad (10)$$

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

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

$$y(k+1) = \sum_{j=1}^N \mu_{B_j}(u(k)) c_j \quad (11)$$

where

$$c_j = \sum_{i=1}^M \lambda_i(\mathbf{x}(k)) \cdot c_{ij} \quad (12)$$

Then we have some rules like:

If $u(k)$ is B_j then $y(k+1)$ is $c_j(k)$, $j = 1, \dots, N$.

Then we will invert it to:

If $r(k+1)$ is $c_j(k)$ then $u(k)$ is B_j $j = 1, \dots, N$.

Because it is singleton, we need interpolate

$$\begin{aligned}
\mu_{C_1}(r) &= \max\left(0, \min\left(1, \frac{c_2 - r}{c_2 - c_1}\right)\right) \\
\mu_{C_j}(r) &= \max\left(0, \min\left(\frac{r - c_{j-1}}{c_j - c_{j-1}}, \frac{c_{j+1} - r}{c_{j+1} - c_j}\right)\right), \quad 1 < j < N, \\
\mu_{C_N}(r) &= \max\left(0, \min\left(\frac{r - c_{N-1}}{c_N - c_{N-1}}, 1\right)\right)
\end{aligned} \quad (13)$$

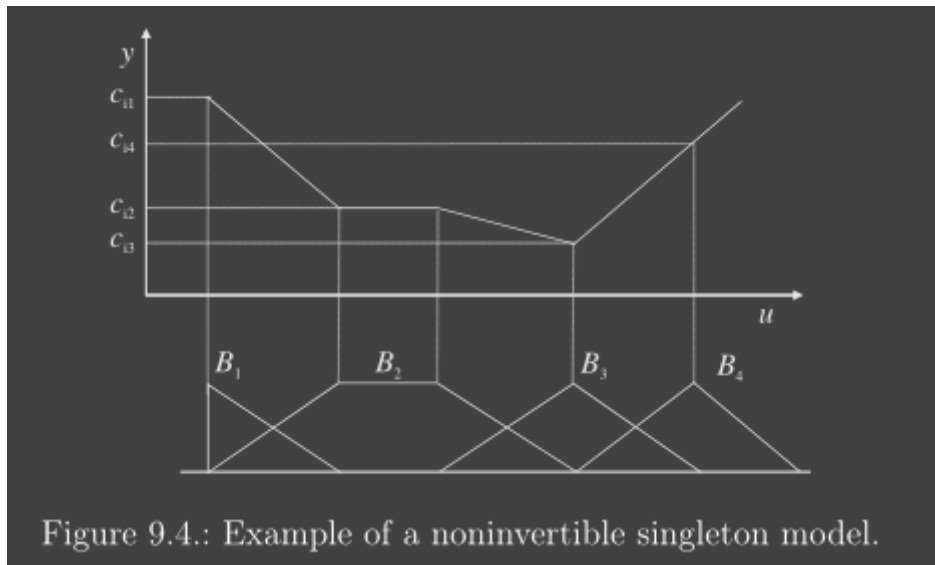
Then we can calculate the output:

$$u(k) = \sum_{j=1}^N \mu_{C_j}(r(k+1))b_j \quad (14)$$

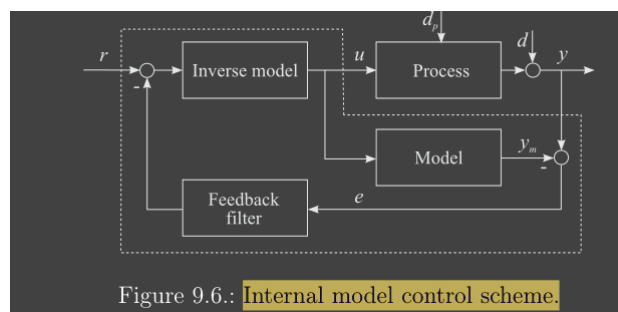
$$y(k+1) = f_x(u(k)) = f_x(f_x^{-1}(r(k+1))) = r(k+1) \quad (15)$$

Notes

A invertible singleton model must with monotonous model.



2. Internal Model Control



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.**

3. Model-Based Predictive Control
