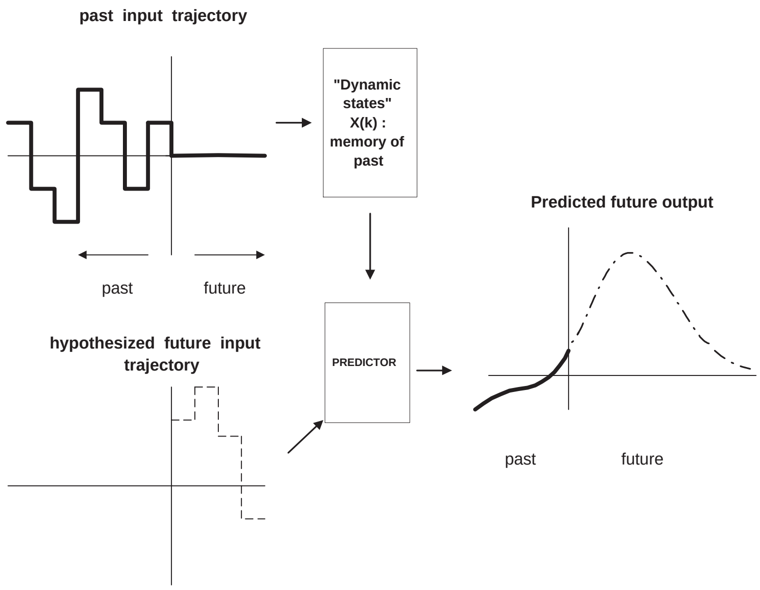
2.2 MULTI-STEP PREDICTION

2.2.1 OVERVIEW

- In control, we are often interested in describing the *future* output behavior with a model.
- For a dynamic system, future output behavior depends on both *past* and future inputs.



Hence, past inputs must be remembered in some form for prediction. *Dynamic states* (in an input / output description) are defined as

> *memory* about the past inputs needed for prediction of the future output behavior

For a same system, states can be defined in many different ways, i.e., there are many ways to remember the past for the purpose of future prediction).

- For instance, states can consist of the entire past input trajectory:

$$x(k) = [v(k-1),\ v(k-2),\ \ldots\ldots,\ v(0)]^T$$

This choice is not practical since the memory size keeps growing with time.

- For an FIR system, one only has to keep n past inputs (Why?) :

$$x(k) = [v(k-1),\ v(k-2),\ \ldots\ldots,\ v(k-n)]^T$$

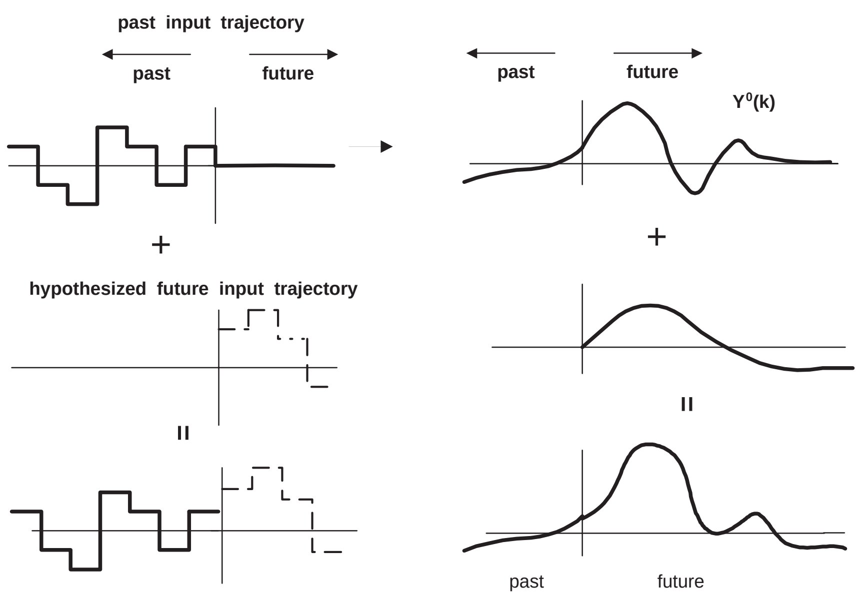
With this choice of $x(k)$, we can certainly build the prediction of the future output behavior.

- Since the ultimate purpose of the memory is to predict future output, the *past* may be more conveniently tracked in terms of its effect on the future rather than the past itself. This is discussed next.

2.2.2 RECURSIVE MULTI-STEP PREDICTION FOR AN FIR SYSTEM

- **Separating Past and Future Input Effects**

  For linear systems, due to the separation principle, the effect of past and (hypothesized) future inputs can be computed separately and added:



- **Past Effects As Memory**

  Define $Y^0(k)$ as *future* output deviation due to *past* input deviation:

$$Y^0(k) = [y^0(k/k),\ y^0(k+1/k),\ \ldots\ldots, y^0(\infty/k)]^T$$

where

$$y^0(i/k) \stackrel{\Delta}{=} y(i) \text{ assuming } v(k+j) = 0 \text{ for } j \geq 0$$

Note that

$$y^0(k/k) = y(k)$$

since the assumption of $v(k+j) = 0, j \geq 0$ does not affect the output at time k.

Although $Y^0(k)$ is infinite dimensional, for FIR system, we only have to keep n terms (why?):

$$Y^0(k) = [y^0(k/k),\ y^0(k+1/k),\ \ldots\ldots, y^0(n/k)]^T$$

This vector can be chosen as *states* since it describes the effect of *past* input deviation on future output deviation.

Future output can be written as

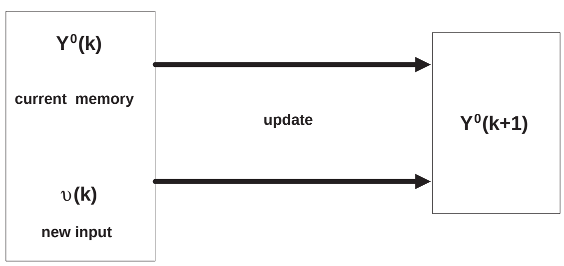
$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ \vdots \\ y(k+p) \end{bmatrix} = \underbrace{\begin{bmatrix} y^0(k+1/k) \\ y^0(k+2/k) \\ \vdots \\ \vdots \\ y^0(k+p/k) \end{bmatrix}}_{\text{Effect of Past Inputs From } Y^0(k)}$$

$$+ \underbrace{\begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ \vdots \\ H_p \end{bmatrix} v(k) + \begin{bmatrix} 0 \\ H_1 \\ \vdots \\ \vdots \\ H_{p-1} \end{bmatrix} v(k+1) + \ldots\ldots + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ H_1 \end{bmatrix} v(k+p-1)}_{\text{Effect of Hypothesized Future Inputs}}$$

We can see that such a defition of states can be very convenient for predictive control.

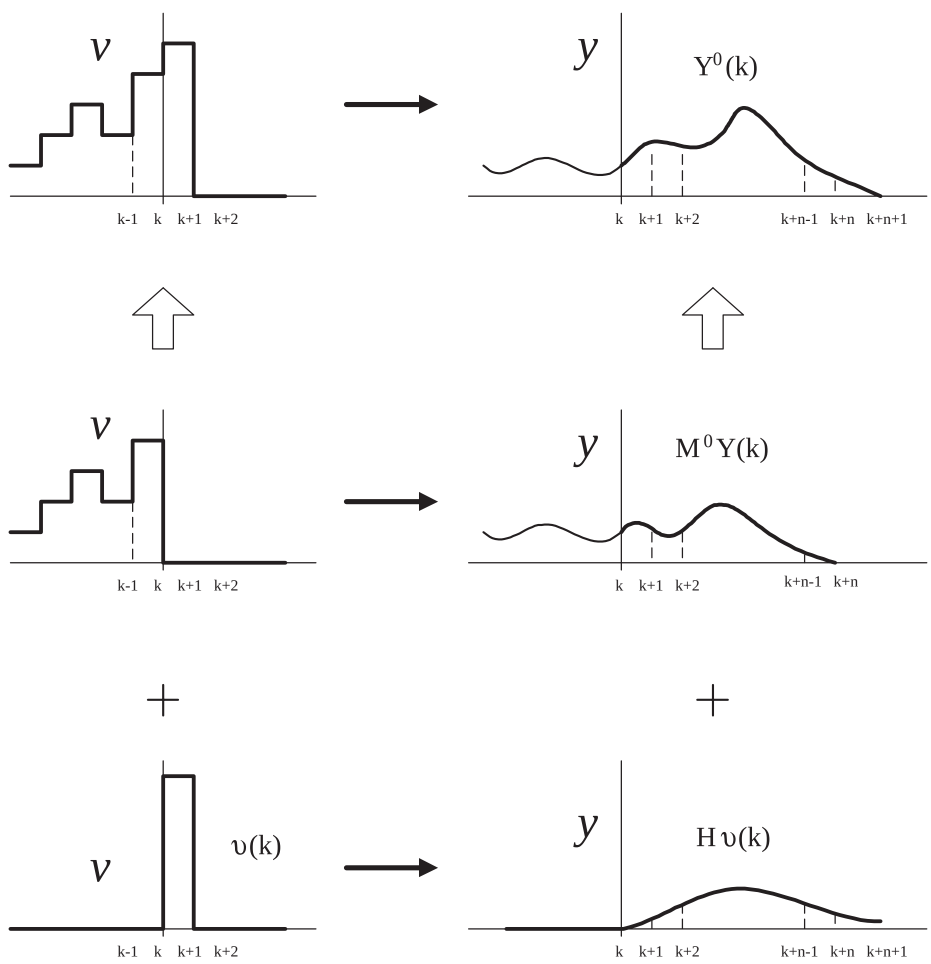
- **Recursive Update of Memory**

  Memory should be updated from one time step to next. For computer implementation, the update should occur in a recursive manner.



$Y^0(k)$ can be updated recursively as follows:

$$
\begin{array}{ccccccc}
Y^0(k) & \rightarrow & M^0Y^0(k) & + & Hv(k) & = & Y^0(k+1) \\
\begin{array}{c}
\left[\begin{array}{c}
y^0(k/k) \\
y^0(k+1/k) \\
\vdots \\
\vdots \\
y^0(k+n-2/k) \\
y^0(k+n-1/k)
\end{array}\right] \\
0 \\
\vdots
\end{array}
& &
\left[\begin{array}{c}
y^0(k+1/k) \\
y^0(k+2/k) \\
\vdots \\
\vdots \\
y^0(k+n-1/k) \\
y^0(k+n/k)
\end{array}\right]
& + &
\left[\begin{array}{c}
H_1 \\
H_2 \\
\vdots \\
\vdots \\
H_{n-1} \\
H_n
\end{array}\right] v(k)
& = &
\left[\begin{array}{c}
y^0(k+1/k+1) \\
y^0(k+2/k+1) \\
\vdots \\
\vdots \\
y^0(k+n-1/k+1) \\
y^0(k+n/k+1)
\end{array}\right]
\end{array}
$$



Mathematically, the above can be represented as

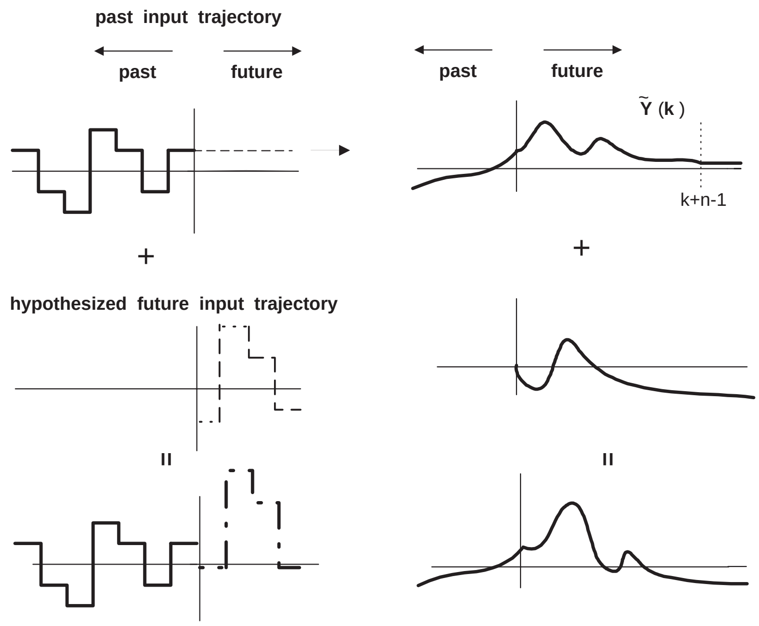
$$Y^0(k+1) = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{M^0} Y^0(k) + \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{n-1} \\ H_n \end{bmatrix} v(k)$$

Note that multiplication by M^0 in the above represents the shift operation (which can be efficiently implemented on a computer).

2.2.3 RECURSIVE MULTI-STEP PREDICTION FOR AN FIR SYSTEM WITH DIFFERENCED INPUT

Multi-step prediction model can be developed in terms of step response coefficients as well.

- **Separating Past and Future Input Effects**
  Apply the superposition as before, but in a slightly different manner:



- **Past Effects As Memory**

  Define $\tilde{Y}(k)$ as *future* output deviation due to *past* input deviation plus current bias:

$$\tilde{Y}(k) = [\tilde{y}(k/k),\ \tilde{y}(k+1/k),\ \ldots\ldots, \tilde{y}(k+n-1/k)]^T$$

where

$$\tilde{y}(i/k) \stackrel{\Delta}{=} y(i) \text{ assuming } \Delta v(k+j) = 0 \text{ for } j \geq 0$$

Note that $\tilde{y}(k+n-1/k) = \tilde{y}(k+n/k) = \ldots = \tilde{y}(\infty/k)$, thus allowing the finite-dimensional representation of future output trajectory. This vector can be chosen as *states*.

Future output can be written as

$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ \vdots \\ y(k+p) \end{bmatrix} = \underbrace{\begin{bmatrix} \tilde{y}(k+1/k) \\ \tilde{y}(k+2/k) \\ \vdots \\ \vdots \\ \tilde{y}(k+p/k) \end{bmatrix}}_{\text{Effect of Past Inputs + Current Bias (from } \tilde{Y}(k))}$$

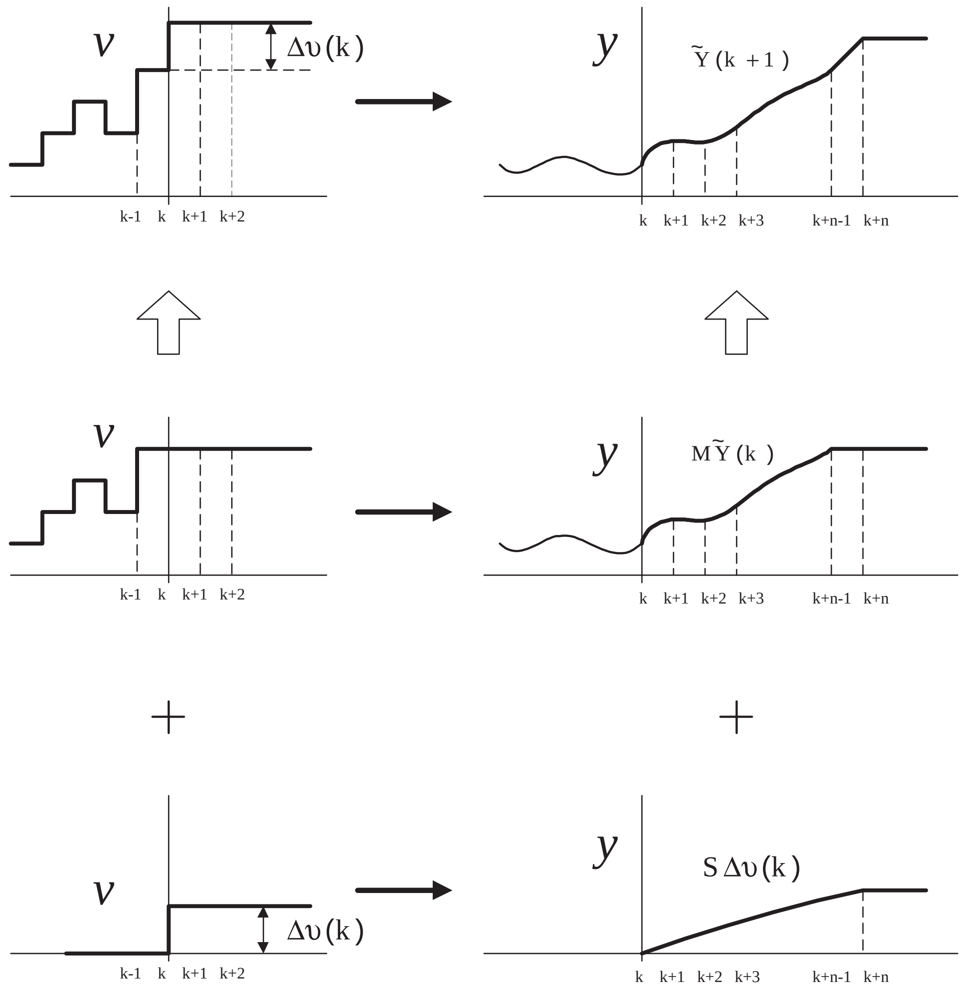
$$+ \underbrace{\begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ \vdots \\ S_p \end{bmatrix} \Delta v(k) + \begin{bmatrix} 0 \\ S_1 \\ \vdots \\ \vdots \\ S_{p-1} \end{bmatrix} \Delta v(k+1) + \ldots + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ S_1 \end{bmatrix} \Delta v(k+p-1)}_{\text{Effect of Hypothesized Future Input Changes}}$$

We can see that such a defition of states can be very convenient for predictive control.

- **Recursive Update of Memory**
  $\tilde{Y}(k)$ can be updated recursively as follows:

$$\begin{array}{cccccc} \tilde{Y}(k) & M\tilde{Y}(k) & + & S\Delta v(k) & = & \tilde{Y}(k+1) \end{array}$$

$$\begin{array}{c} \left[\begin{array}{c} \tilde{y}(k/k) \\ \tilde{y}(k+1/k) \\ \vdots \\ \vdots \\ \tilde{y}(k+n-2/k) \\ \tilde{y}(k+n-1/k) \end{array}\right] \\ \tilde{y}(k+n/k) \\ \vdots \end{array} \quad \begin{bmatrix} \tilde{y}(k+1/k) \\ \tilde{y}(k+2/k) \\ \vdots \\ \vdots \\ \tilde{y}(k+n-1/k) \\ \tilde{y}(k+n/k) \end{bmatrix} + \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ \vdots \\ S_{n-1} \\ S_n \end{bmatrix} \Delta v(k) = \begin{bmatrix} \tilde{y}(k+1/k+1) \\ \tilde{y}(k+2/k+1) \\ \vdots \\ \vdots \\ \tilde{y}(k+n-1/k+1) \\ \tilde{y}(k+n/k+1) \end{bmatrix}$$

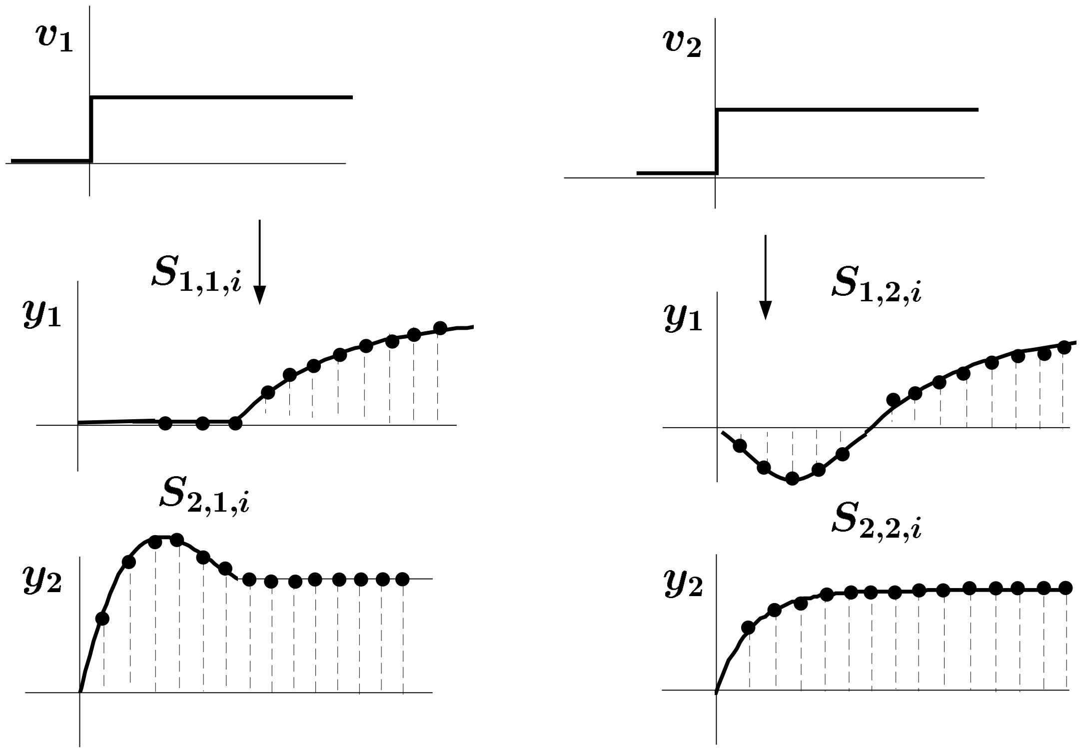


Hence,

$$\tilde{Y}(k+1) = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}_{M} \tilde{Y}(k) + \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{n-1} \\ S_n \end{bmatrix} \Delta v(k)$$

Note that multiplication by M in the above represents a shift operation of different kind (the last element is repeated).

2.2.4 MULTIVARIABLE GENERALIZATION



$$S_i \triangleq i_{\text{th}} \text{ step response coefficient matrix}$$
$$= \begin{bmatrix} S_{1,1,i} & S_{1,2,i} \\ S_{2,1,i} & S_{2,2,i} \end{bmatrix}$$

In general

$$S_i = \begin{bmatrix} S_{1,1,i} & S_{1,2,i} & \cdots & \cdots & S_{1,n_v,i} \\ S_{2,1,i} & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ S_{n_y,1,i} & S_{n_y,2,i} & \cdots & \cdots & S_{n_y,n_v,i} \end{bmatrix}$$

Again, define $\tilde{Y}_{k+1}$ and $\tilde{Y}_k$ in the same manner as before (now they are $(n \cdot n_y)$-dimensional vectors). Then,

$$\tilde{Y}(k+1) = M\tilde{Y}(k) + S\Delta v(k)$$

where

$$M = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ 0 & 0 & 0 & \cdots & I \end{bmatrix}$$

$$S = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{n-1} \\ S_n \end{bmatrix}$$

where I is an $n_y \times n_y$ indentity matrix. Again, it merely represents the shift-operation; such a matrix does not need to be created in reality.

2.3 DYNAMIC MATRIX CONTROL ALGORITHM

2.3.1 MAJOR CONSTITUENTS