state space from ARMA (and viceversa)

Contents map

developed content units	taxonomy levels
realization	u1, e1

prerequisite content units	taxonomy levels
ARMA model	u1, e1
state space model model	u1, e1
matrix inversion	u1, e1
Zeta transforms	u1, e1

Main ILO of sub-module "state space from ARMA (and viceversa)"

Determine the state space structure of an discrete time LTI system starting from an ARMA RR

ARMA models

$$y^{[n]} = a_{n-1} y^{[n-1]} + \ldots + a_0 y + b_m u^{[m]} + \ldots + b_0 u$$

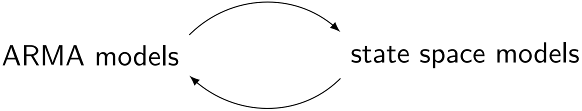
State space representations - Notation

$$
\begin{aligned}
x_1^+ &= f_1\left(x_1, \ldots, x_n,\ u_1, \ldots, u_m \right) \\
&\vdots \\
x_n^+ &= f_n\left(x_1, \ldots, x_n,\ u_1, \ldots, u_m \right) \\
y_1 &= g_1\left(x_1, \ldots, x_n,\ u_1, \ldots, u_m \right) \\
&\vdots \\
y_p &= g_p\left(x_1, \ldots, x_n,\ u_1, \ldots, u_m \right)
\end{aligned}
$$

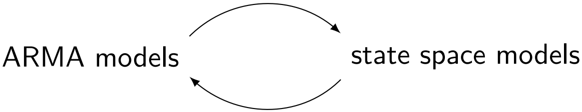
$$
\begin{aligned}
\boldsymbol{x}^+ &= \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}\right) \\
\boldsymbol{y} &= \boldsymbol{g}\left(\boldsymbol{x}, \boldsymbol{u}\right)
\end{aligned}
$$

- $\boldsymbol{f}$ = state transition map
- $\boldsymbol{g}$ = output map

This module:



This module:



But why do we study this?

because from physical laws we get ARMA,
but with state space we get more explainable models

From state space to ARMA

SS to ARMA

Tacit assumption: $\boldsymbol{x}[0] = \mathbf{0}$

$$\begin{cases} \boldsymbol{x}^+ & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases}$$

SS to ARMA

Tacit assumption: $\boldsymbol{x}[0] = \mathbf{0}$

$$\begin{cases} \boldsymbol{x}^{+} & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases} \qquad \rightarrow \qquad \mathcal{Z}\left(\begin{cases} \boldsymbol{x}^{+} & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases} \right)$$

SS to ARMA

Tacit assumption: $\boldsymbol{x}[0] = \mathbf{0}$

$$\begin{cases} \boldsymbol{x}^+ &= A\boldsymbol{x} + Bu \\ y &= C\boldsymbol{x} + Du \end{cases} \quad \rightarrow \quad \mathcal{Z}\left(\begin{cases} \boldsymbol{x}^+ &= A\boldsymbol{x} + Bu \\ y &= C\boldsymbol{x} + Du \end{cases} \right)$$

$$\rightarrow \quad \begin{cases} zX &= AX + BU \\ Y &= CX + DU \end{cases}$$

SS to ARMA

Tacit assumption: $\boldsymbol{x}[0] = \mathbf{0}$

$$\begin{cases} \boldsymbol{x}^+ & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases} \rightarrow \mathcal{Z}\left(\begin{cases} \boldsymbol{x}^+ & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases} \right)$$

$$\rightarrow \begin{cases} zX & = AX + BU \\ Y & = CX + DU \end{cases}$$

$$\rightarrow \begin{cases} (zI - A)X & = BU \\ Y & = CX + DU \end{cases}$$

SS to ARMA

Tacit assumption: $\mathbf{x}[0] = \mathbf{0}$

$$\begin{cases} \mathbf{x}^+ &= A\mathbf{x} + Bu \\ y &= C\mathbf{x} + Du \end{cases} \rightarrow \mathcal{Z}\left(\begin{cases} \mathbf{x}^+ &= A\mathbf{x} + Bu \\ y &= C\mathbf{x} + Du \end{cases}\right)$$

$$\rightarrow \begin{cases} zX &= AX + BU \\ Y &= CX + DU \end{cases}$$

$$\rightarrow \begin{cases} (zI - A)X &= BU \\ Y &= CX + DU \end{cases}$$

$$\rightarrow \begin{cases} X &= (zI - A)^{-1}BU \quad (*) \\ Y &= CX + DU \end{cases}$$

SS to ARMA

Tacit assumption: $\boldsymbol{x}[0] = \mathbf{0}$

$$\begin{cases} \boldsymbol{x}^+ & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases} \quad \rightarrow \quad \mathcal{Z}\left(\begin{cases} \boldsymbol{x}^+ & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases} \right)$$

$$\rightarrow \quad \begin{cases} zX & = AX + BU \\ Y & = CX + DU \end{cases}$$

$$\rightarrow \quad \begin{cases} (zI - A)X & = BU \\ Y & = CX + DU \end{cases}$$

$$\rightarrow \quad \begin{cases} X & = (zI - A)^{-1}BU \quad (*) \\ Y & = CX + DU \end{cases}$$

$$\Rightarrow \quad Y = \left(C(zI - A)^{-1}B + D \right) U$$

SS to ARMA

Tacit assumption: $\boldsymbol{x}[0] = \mathbf{0}$

$$\begin{cases} \boldsymbol{x}^{+} & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases} \quad \rightarrow \quad \mathcal{Z}\left(\begin{cases} \boldsymbol{x}^{+} & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases} \right)$$

$$\rightarrow \quad \begin{cases} zX & = AX + BU \\ Y & = CX + DU \end{cases}$$

$$\rightarrow \quad \begin{cases} (zI - A)X & = BU \\ Y & = CX + DU \end{cases}$$

$$\rightarrow \quad \begin{cases} X & = (zI - A)^{-1}BU \quad (*) \\ Y & = CX + DU \end{cases}$$

$$\Rightarrow \quad Y = \left(C(zI - A)^{-1}B + D \right) U$$

$$\Rightarrow \quad Y(z) = \frac{\text{polynomial in } z}{\text{polynomial in } z} U(z)$$

A note on the last formula

$$Y(z) = \frac{\text{polynomial in } z}{\text{polynomial in } z} U(z) \quad \mapsto \quad \text{ARMA:}$$

$$Y(z) = \frac{z + 3}{2z^3 + 3z} U(z) \quad \mapsto \quad 2y^{+++} + 3y^{+} = u^{+} + 3u$$

A note on the second to last formula

$$Y = \left(C(zI - A)^{-1} B + D \right) U$$

DISCLAIMER: in this course we consider SISO systems, thus C and $B =$ vectors, and $D =$ scalar (if present)

Numerical Example: 2×2 State-Space to ARMA

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \end{bmatrix}$$

Numerical Example: 2×2 State-Space to ARMA

Step 1: State-Space Equations

$$\begin{cases} x_1^+ = x_1 + 2x_2 + u \\ x_2^+ = 3x_1 + 4x_2 \\ y = x_1 \end{cases}$$

Numerical Example: 2×2 State-Space to ARMA

Step 2: Zeta Transform

$$\begin{cases} zX_1(z) = X_1(z) + 2X_2(z) + U(z) \\ zX_2(z) = 3X_1(z) + 4X_2(z) \\ Y(z) = X_1(z) \end{cases}$$

Numerical Example: 2×2 State-Space to ARMA

Step 3: Rearrange in Matrix Form

$$\begin{cases} (zI - A)X(z) = BU(z) \\ Y(z) = CX(z) + DU(z) \end{cases}$$

implies

$$\begin{cases} \begin{bmatrix} z-1 & -2 \\ -3 & z-4 \end{bmatrix} \begin{bmatrix} X_1(z) \\ X_2(z) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} U(z) \\ Y(z) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} X_1(z) \\ X_2(z) \end{bmatrix} \end{cases}$$

Numerical Example: 2×2 State-Space to ARMA

Step 4: Solve for $X(z)$

$$X(z) = (zI - A)^{-1} B U(z)$$

$$(zI - A) = \begin{bmatrix} z-1 & -2 \\ -3 & z-4 \end{bmatrix}$$

$$(zI - A)^{-1} = \frac{1}{(z-1)(z-4) - (-2)(-3)} \begin{bmatrix} z-4 & 2 \\ 3 & z-1 \end{bmatrix}$$

$$\det(zI - A) = (z-1)(z-4) - 6 = z^2 - 5z - 2$$

$$(zI - A)^{-1} = \frac{1}{z^2 - 5z - 2} \begin{bmatrix} z-4 & 2 \\ 3 & z-1 \end{bmatrix}$$

Numerical Example: 2×2 State-Space to ARMA

Step 5: Multiply by B

Now, multiply by B:

$$X(z) = \frac{1}{z^2 - 5z - 2} \begin{bmatrix} z-4 & 2 \\ 3 & z-1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} U(z) = \frac{1}{z^2 - 5z - 2} \begin{bmatrix} z-4 \\ 3 \end{bmatrix} U(z)$$

Numerical Example: 2×2 State-Space to ARMA

Step 6: Solve for $Y(z)$

Substitute $X(z)$ into the output equation:

$$Y(z) = CX(z) + DU(z) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} X_1(z) \\ X_2(z) \end{bmatrix} = X_1(z)$$

Thus:

$$Y(z) = \frac{z - 4}{z^2 - 5z - 2} U(z)$$

Numerical Example: 2×2 State-Space to ARMA

Step 7: Final Result

Transfer function $H(z)$:

$$H(z) = \frac{Y(z)}{U(z)} = \frac{z - 4}{z^2 - 5z - 2}$$

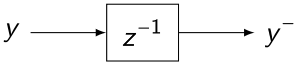
and from this we get the ARMA representation of the system as before

From ARMA to SS

Starting point (blending Zeta notation with time notation)

$$y[k] = \frac{b(z)}{a(z)} u[k] = \frac{b_1 z^{n-1} + \ldots + b_n}{z^n + a_1 z^{n-1} + \ldots + a_n} u[k]$$

Building block = the time-delay (block)

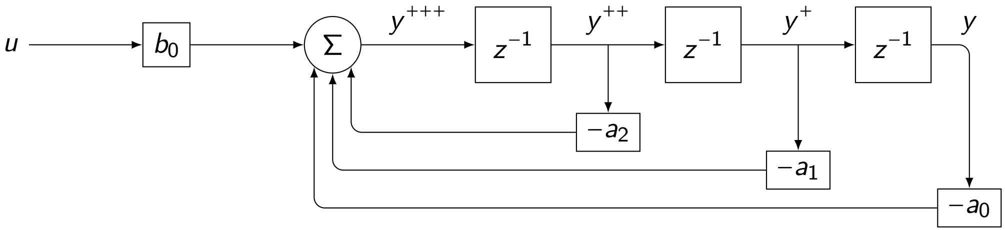


How do we use delays?

$$y^{+++} + a_2 y^{++} + a_1 y^{+} + a_0 y = b_0 u$$

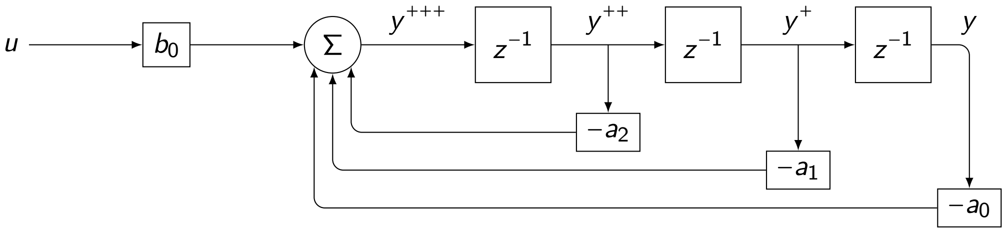
How do we use delays?

$$y^{+++} + a_2 y^{++} + a_1 y^{+} + a_0 y = b_0 u$$
$$\downarrow$$
$$y^{+++} = -a_2 y^{++} - a_1 y^{+} - a_0 y + b_0 u$$



How do we use delays?

$$y^{+++} + a_2 y^{++} + a_1 y^{+} + a_0 y = b_0 u$$
$$\downarrow$$
$$y^{+++} = -a_2 y^{++} - a_1 y^{+} - a_0 y + b_0 u$$



instrumental for later: $x^{[3]} = -a_2 x^{[2]} - a_1 x^{[1]} - a_0 x^{[0]} + b_0 u$

and the state vector is $\left[x^{[2]}, x^{[1]}, x^{[0]} \right]$

Towards SS with a useful trick

$$y[k] = \frac{b(z)}{a(z)} u[k] = \frac{b_1 z^{n-1} + \ldots + b_n}{z^n + a_1 z^{n-1} + \ldots + a_n} u[k]$$

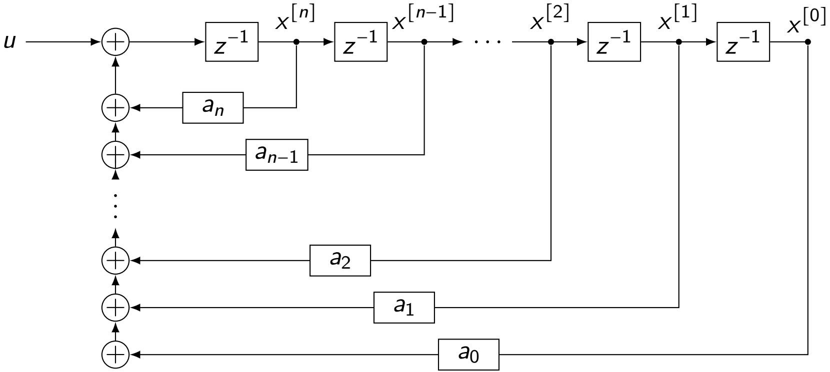
Towards SS with a useful trick

$$y[k] = \frac{b(z)}{a(z)}u[k] = \frac{b_1 z^{n-1} + \ldots + b_n}{z^n + a_1 z^{n-1} + \ldots + a_n}u[k] \rightarrow \begin{cases} x^{[0]} = \dfrac{1}{a(z)}u \\ y = b(z)x^{[0]} \end{cases}$$

This is an AR model on $x^{[0]}$

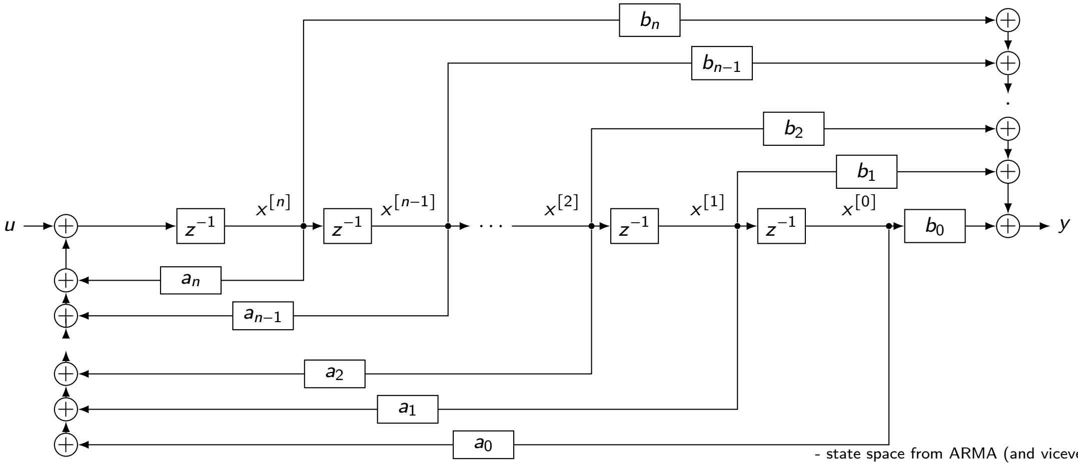
$$x^{[0]} = \frac{1}{a(z)} u \qquad \Longrightarrow \qquad a(z) x^{[0]} = u$$

implies



Completing the picture (a MA from x_n to y)

$$y[k] = b_n x^{[n]}[k] + \ldots + b_0 x^{[0]}[k]$$



From concepts to formulas

$$\begin{cases} y[k] = b_n x^{[n]}[k] + \ldots + b_0 x^{[0]}[k] \\ x^{[n]+}[k] = -a_n x^{[n]}[k] - \ldots - a_0 x^{[0]}[k] + u[k] \\ x^{[i]+}[k] = x^{[i]}[k] \end{cases} \rightarrow \begin{cases} \boldsymbol{x}^+ & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases}$$

From concepts to formulas

$$\begin{cases} y[k] = b_n x^{[n]}[k] + \ldots + b_0 x^{[0]}[k] \\ x^{[n]+}[k] = -a_n x^{[n]}[k] - \ldots - a_0 x^{[0]}[k] + u[k] \\ x^{[i]+}[k] = x^{[i]}[k] \end{cases} \rightarrow \begin{cases} \boldsymbol{x}^+ & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases}$$

$$\boldsymbol{x}^+ := \begin{bmatrix} x^{[n]+} \\ x^{[n-1]+} \\ x^{[n-2]+} \\ \vdots \\ x^{[0]+} \end{bmatrix}$$

From concepts to formulas

$$
\begin{cases}
y[k] = b_n x^{[n]}[k] + \ldots + b_0 x^{[0]}[k] \\
x^{[n]+}[k] = -a_n x^{[n]}[k] - \ldots - a_0 x^{[0]}[k] + u[k] \\
x^{[i]+}[k] = x^{[i]}[k]
\end{cases}
\rightarrow
\begin{cases}
\boldsymbol{x}^+ & = A\boldsymbol{x} + Bu \\
y & = C\boldsymbol{x} + Du
\end{cases}
$$

$$
\boldsymbol{x}^+ := \begin{bmatrix} x^{[n]+} \\ x^{[n-1]+} \\ x^{[n-2]+} \\ \vdots \\ x^{[0]+} \end{bmatrix} = \begin{bmatrix} -a_n & -a_{n-1} & \ldots & \ldots & -a_0 \\ 1 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ & \ddots & \ddots & \ddots & \\ & & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x^{[n]} \\ x^{[n-1]} \\ x^{[n-2]} \\ \vdots \\ x^{[0]} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u
$$

And y?

$$\begin{cases} y[k] = b_n x^{[n]}[k] + \ldots + b_0 x^{[0]}[k] \\ x^{[n]+}[k] = -a_n x^{[n]}[k] - \ldots - a_0 x^{[0]}[k] + u[k] \\ x^{[i]+}[k] = x^{[i]}[k] \end{cases} \rightarrow \begin{cases} \boldsymbol{x}^+ & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases}$$

$$y = \begin{bmatrix} b_n & b_{n-1} & b_{n-2} & \ldots & b_0 \end{bmatrix} \begin{bmatrix} x^{[n]} \\ x^{[n-1]} \\ x^{[n-2]} \\ \vdots \\ x^{[0]} \end{bmatrix}$$

From ARMA to state space (in Control Canonical Form)

$$
\begin{cases}
\begin{bmatrix} x^{[n]+} \\ x^{[n-1]+} \\ x^{[n-2]+} \\ \vdots \\ x^{[0]+} \end{bmatrix} = \begin{bmatrix} -a_n & -a_{n-1} & \dots & \dots & -a_0 \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ & \ddots & \ddots & \ddots & \\ & & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x^{[n]} \\ x^{[n-1]} \\ x^{[n-2]} \\ \vdots \\ x^{[0]} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u \\
y = \begin{bmatrix} b_n & b_{n-1} & b_{n-2} & \dots & b_0 \end{bmatrix} \begin{bmatrix} x^{[n]} \\ x^{[n-1]} \\ x^{[n-2]} \\ \vdots \\ x^{[0]} \end{bmatrix}
\end{cases}
$$

Matlab / Python implementation

```
[A, B, C, D] = tf2ss([bn .. b0], [1 an .. a0])
```

Summarizing

Determine the state space structure of an discrete time LTI system starting from an ARMA RR

- there are some formulas, that you may simply know by heart, or that you may want to understand
- for understanding there is the need to get how the transformations work, and what is what
- likely the most important point is that to go from ARMA to SS the (likely) most simple strategy is to build the states as a chain of delays, and ladder on top of that

Most important python code for this sub-module

These functions have also their opposite, i.e., tf2ss

- `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.ss2tf.html`
- `https://python-control.readthedocs.io/en/latest/generated/control.ss2tf.html`

Self-assessment material

Question 1

Given the discrete-time ARMA model:

$$y^{+++} + a_2 y^{++} + a_1 y^{+} + a_0 y = b_0 u,$$

what is the correct state-space representation in control canonical form?

Potential answers:

I:

$$\begin{cases} x_1^{+} = -a_2 x_1 - a_1 x_2 - a_0 x_3 + b_0 u \\ x_2^{+} = x_1 \\ x_3^{+} = x_2 \\ y = x_3 \end{cases}$$

II:

$$\begin{cases} x_1^{+} = -a_2 x_1 - a_1 x_2 - a_0 x_3 + u \end{cases}$$

Question 2

For the state-space system:

$$\begin{cases} x_1^+ = -3x_1 + 2x_2 + u \\ x_2^+ = x_1 \\ y = 4x_1 + x_2 \end{cases},$$

what is the equivalent ARMA model?

Potential answers:

I: $y^{++} + 3y^{+} - 2y = 4u^{+} + u$
II: $y^{++} + 3y^{+} - 2y = u^{+} + 4u$
III: $y^{++} - 3y^{+} + 2y = u^{+} + 4u$
IV: $y^{++} + 3y^{+} + 2y = 4u^{+} + u$
V: I do not know

Question 3

In discrete-time state-space representations, the delay operator z^{-1} primarily:

Potential answers:

I: Approximates continuous-time integration
II: Implements the time-shift operation $x[k] \to x[k-1]$
III: Adds stochastic noise to the system
IV: Reduces computational complexity
V: I do not know

Question 4

The control canonical form's state matrix A always:

Potential answers:

I: Is diagonal with poles on the diagonal

II: Has AR coefficients in its first row and shifted identity below

III: Makes the B matrix identical to $C^\top$

IV: Minimizes the number of nonzero elements

V: I do not know

Question 5

When converting state-space to ARMA via Z-transform, the operator $(zI - A)^{-1}$:

Potential answers:

I: Directly gives the system's impulse response
II: Is the resolvent matrix needed to solve for $X(z)$
III: Always results in a diagonalizable matrix
IV: Can be omitted if $D \neq 0$
V: I do not know

Recap of sub-module "state space from ARMA (and viceversa)"

- one can go from ARMA to state space and viceversa
- we did not see this, but watch out that the two representations are not equivalent: there are systems that one can represent with state space and not with ARMA, and viceversa
- typically state space is more interpretable, and tends to be the structure used when doing model predictive control

?