Systems Laboratory, Spring 2025

Damiano Varagnolo –

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
Laplace transforms	u1, e1

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

Determine the state space structure of an
LTI system starting from an ARMA ODE

ARMA models

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

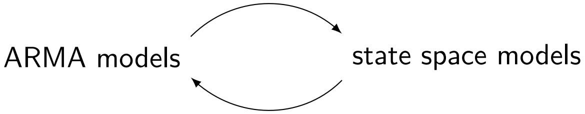
State space representations - Notation

$$
\begin{aligned}
\dot{x}_1 &= f_1\left(x_1, \ldots, x_n,\ u_1, \ldots, u_m \right) \\
&\vdots \\
\dot{x}_n &= f_n\left(x_1, \ldots, x_n,\ u_1, \ldots, u_m \right) \\
y_1 &= g_1\left(x_1, \ldots, x_n,\ u_1, \ldots, u_m \right) \\
&\vdots \\
y_p &= g_p\left(x_1, \ldots, x_n,\ u_1, \ldots, u_m \right)
\end{aligned}
$$

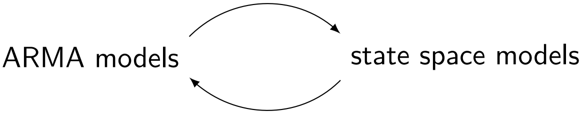
$$
\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}\right) \\
\boldsymbol{y} &= \boldsymbol{g}\left(\boldsymbol{x}, \boldsymbol{u}\right)
\end{aligned}
$$

- $\boldsymbol{f}$ = 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: $\boldsymbol{x}(0) = \mathbf{0}$

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

SS to ARMA

Tacit assumption: $\boldsymbol{x}(0) = \mathbf{0}$

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

SS to ARMA

Tacit assumption: $\boldsymbol{x}(0) = \mathbf{0}$

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

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

SS to ARMA

Tacit assumption: $\boldsymbol{x}(0) = \mathbf{0}$

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

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

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

SS to ARMA

Tacit assumption: $\boldsymbol{x}(0) = \mathbf{0}$

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

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

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

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

SS to ARMA

Tacit assumption: $\boldsymbol{x}(0) = \mathbf{0}$

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

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

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

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

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

SS to ARMA

Tacit assumption: $\boldsymbol{x}(0) = \mathbf{0}$

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

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

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

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

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

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

A note on the last formula

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

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

A note on the second to last formula

$$Y = \Big(C(sI - A)^{-1} B + D \Big) U$$

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

$$\begin{cases} \dot{x}_1 = x_1 + 2x_2 + u \\ \dot{x}_2 = 3x_1 + 4x_2 \\ y = x_1 \end{cases}$$

Numerical Example: 2×2 State-Space to ARMA

Step 2: Laplace Transform

$$\begin{cases} sX_1(s) = X_1(s) + 2X_2(s) + U(s) \\ sX_2(s) = 3X_1(s) + 4X_2(s) \\ Y(s) = X_1(s) \end{cases}$$

Numerical Example: 2×2 State-Space to ARMA

Step 3: Rearrange in Matrix Form

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

implies

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

Numerical Example: 2×2 State-Space to ARMA

Step 4: Solve for $X(s)$

$$X(s) = (sI - A)^{-1} BU(s)$$

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

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

$$\det(sI - A) = (s-1)(s-4) - 6 = s^2 - 5s - 2$$

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

Numerical Example: 2×2 State-Space to ARMA

Step 5: Multiply by B

Now, multiply by B:

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

Numerical Example: 2×2 State-Space to ARMA

Step 6: Solve for $Y(s)$

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

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

Thus:

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

Numerical Example: 2×2 State-Space to ARMA

Step 7: Final Result

Transfer function $H(s)$:

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

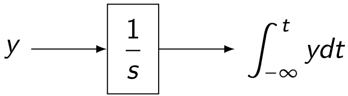
and from this we get the ARMA representation of the system as before

From ARMA to SS

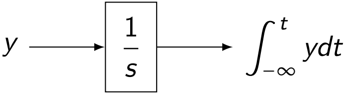
Starting point (blending Laplace notation with time notation)

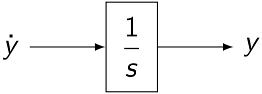
$$y(t) = \frac{b(s)}{a(s)} u(t) = \frac{b_1 s^{n-1} + \ldots + b_n}{s^n + a_1 s^{n-1} + \ldots + a_n} u(t)$$

Building block = the integrator (block)

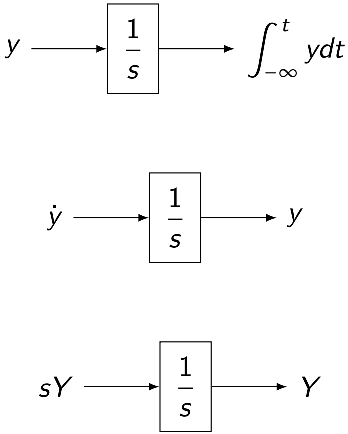


Building block = the integrator (block)





Building block = the integrator (block)



How do we use integrators?

$$\dddot{y} + a_1\ddot{y} + a_2\dot{y} + a_3 y = b_1 u$$

How do we use integrators?

$$\dddot{y} + a_1 \ddot{y} + a_2 \dot{y} + a_3 y = b_1 u$$

$$\downarrow$$

$$\dddot{y} = -a_1 \ddot{y} - a_2 \dot{y} - a_3 y + b_1 u$$

Towards SS with a useful trick

$$y(t) = \frac{b(s)}{a(s)} u(t) = \frac{b_1 s^{n-1} + \ldots + b_n}{s^n + a_1 s^{n-1} + \ldots + a_n} u(t)$$

Towards SS with a useful trick

$$y(t) = \frac{b(s)}{a(s)}u(t) = \frac{b_1 s^{n-1} + \ldots + b_n}{s^n + a_1 s^{n-1} + \ldots + a_n}u(t) \rightarrow \begin{cases} x_n(t) = \dfrac{1}{a(s)}u(t) \\ y(t) = b(s)x_n(t) \end{cases}$$

This is an AR model on x_n

$$x_n(t) = \frac{1}{a(s)} u(t) \qquad \Longrightarrow \qquad a(s) x_n(t) = u(t)$$

implies

$$x_n = \frac{1}{s} x_{n-1} \qquad \rightarrow \qquad x_{n-1} = s x_n$$

This is an AR model on x_n

$$x_n(t) = \frac{1}{a(s)} u(t) \qquad \Longrightarrow \qquad a(s) x_n(t) = u(t)$$

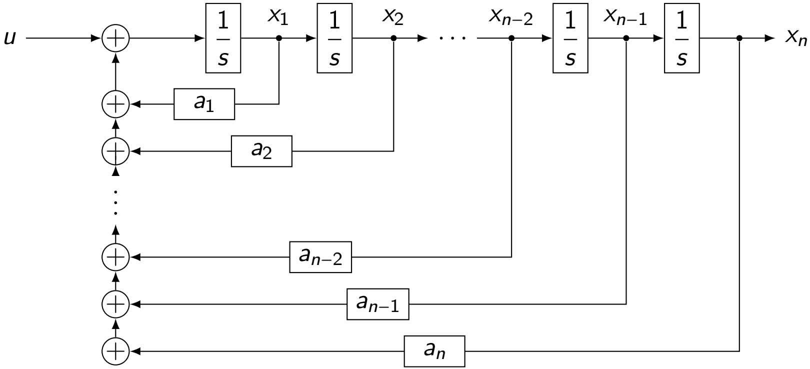
implies

$$x_n = \frac{1}{s} x_{n-1} \quad \rightarrow \quad x_{n-1} = s x_n \quad \rightarrow \quad \begin{cases} x_{n-1} & = s x_n \\ x_{n-2} & = s^2 x_n \\ \vdots & \\ x_2 & = s^{n-2} x_n \\ x_1 & = s^{n-1} x_n \end{cases}$$

This is an AR model on x_n

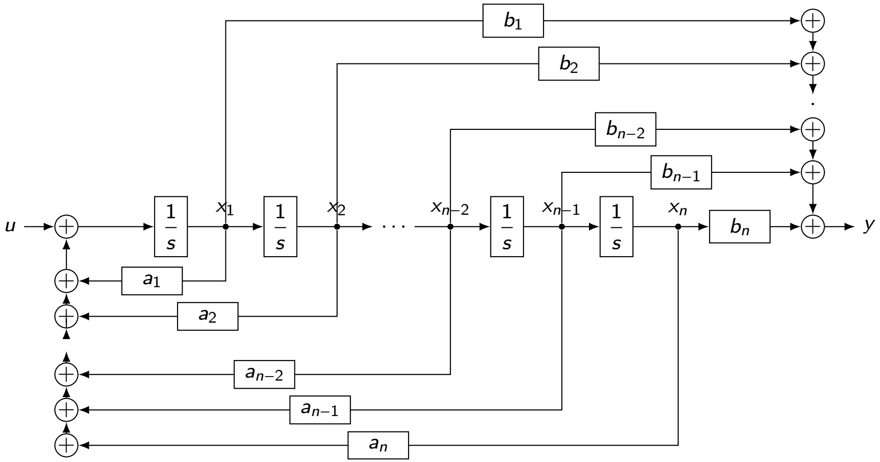
$$x_n(t) = \frac{1}{a(s)} u(t) \qquad \Longrightarrow \qquad a(s) x_n(t) = u(t)$$

implies



Completing the picture (a MA from x_n to y)

$$y(t) = b(s)x_n(t) = b_1x_1(t) + \ldots + b_nx_n(t)$$



From concepts to formulas

$$\begin{cases} y(t) = b_1 x_1(t) + \ldots + b_n x_n(t) \\ \dot{x}_1(t) = -a_1 x_1(t) - \ldots - a_n x_n(t) + u(t) \\ \dot{x}_i(t) = x_{i-1}(t) \end{cases} \rightarrow \begin{cases} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + Bu \\ y &= C\boldsymbol{x} + Du \end{cases}$$

From concepts to formulas

$$\begin{cases} y(t) = b_1 x_1(t) + \ldots + b_n x_n(t) \\ \dot{x}_1(t) = -a_1 x_1(t) - \ldots - a_n x_n(t) + u(t) \\ \dot{x}_i(t) = x_{i-1}(t) \end{cases} \rightarrow \begin{cases} \dot{\boldsymbol{x}} & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases}$$

$$\dot{\boldsymbol{x}} := \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_n \end{bmatrix}$$

From concepts to formulas

$$\begin{cases} y(t) = b_1 x_1(t) + \ldots + b_n x_n(t) \\ \dot{x}_1(t) = -a_1 x_1(t) - \ldots - a_n x_n(t) + u(t) \\ \dot{x}_i(t) = x_{i-1}(t) \end{cases} \rightarrow \begin{cases} \dot{\boldsymbol{x}} & = A\boldsymbol{x} + Bu \\ y & = C\boldsymbol{x} + Du \end{cases}$$

$$\dot{\boldsymbol{x}} := \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & \ldots & \ldots & -a_n \\ 1 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ & \ddots & \ddots & \ddots & \\ & & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u$$

And y?

$$
\begin{cases}
y(t) = b_1 x_1(t) + \ldots + b_n x_n(t) \\
\dot{x}_1(t) = -a_1 x_1(t) - \ldots - a_n x_n(t) + u(t) \\
\dot{x}_i(t) = x_{i-1}(t)
\end{cases}
\rightarrow
\begin{cases}
\dot{\mathbf{x}} & = A\mathbf{x} + Bu \\
y & = C\mathbf{x} + Du
\end{cases}
$$

$$
y = \begin{bmatrix} b_1 & b_2 & b_3 & \ldots & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}
$$

From ARMA to state space (in Control Canonical Form)

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & \dots & \dots & -a_n \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ & \ddots & \ddots & \ddots & \\ & & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u \\ y = \begin{bmatrix} b_1 & b_2 & b_3 & \dots & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \end{cases}$$

Matlab / Python implementation

```
[A, B, C, D] = tf2ss([b1 b2 .. bn], [1 a1 a2 .. an])
```

Summarizing

Determine the state space structure of an
LTI system starting from an ARMA ODE

- 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 integrators, 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

What is the role of $(sI - A)^{-1}$ in the derivation of the transfer function from a state-space model?

Potential answers:

I: It represents the output matrix C.
II: It is used to solve for the state vector $X(s)$ in the Laplace domain.
III: It defines the input matrix B.
IV: It is the Laplace transform of the state transition matrix.
V: I do not know

Question 2

What is the structure of the A matrix in the control canonical form of a state-space model?

Potential answers:

I: An upper Hessenberg matrix with a lower diagonal of ones and coefficients on the first row from the denominator polynomial.
II: A diagonal matrix with the eigenvalues of the system.
III: A lower triangular matrix with zeros on the diagonal.
IV: A symmetric matrix with off-diagonal elements equal to zero.
V: I do not know

Question 3

What is the purpose of the integrator block in the conversion from ARMA to state-space models?

Potential answers:

I: To differentiate the input signal.
II: To invert the Laplace transform of the output.
III: To construct the state variables as a chain of scaled integrators.
IV: To compute the determinant of the state matrix.
V: I do not know

Question 4

What does the transfer function $H(s) = \frac{Y(s)}{U(s)}$ represent in the context of state-space models?

Potential answers:

I: The state transition matrix.
II: The input matrix B.
III: The determinant of the state matrix.
IV: The relationship between the input $U(s)$ and the output $Y(s)$ in the Laplace domain.
V: I do not know

Question 5

In the context of SISO systems, what are the dimensions of the matrices C and B in a state space representation?

Potential answers:

I: C is a scalar, and B is a vector.
II: C is a row vector, and B is a column vector.
III: C is a square matrix, and B is a scalar.
IV: C is a column vector, and B is a row vector.
V: I do not know

Question 6

Given the state-space matrices $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$, and $D = \begin{bmatrix} 0 \end{bmatrix}$, what is the transfer function $H(s)$?

Potential answers:

I: $H(s) = \dfrac{s - 4}{s^2 - 5s - 2}$

II: $H(s) = \dfrac{s - 1}{s^2 - 5s - 2}$

III: $H(s) = \dfrac{s + 3}{s^2 - 5s - 2}$

IV: $H(s) = \dfrac{s - 2}{s^2 - 5s - 2}$

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

?