Reconstruction of Boolean networks and optimal control

Elena Valcher (joint work with E. Fornasini)

University of Padova, Italy

IEEE CDC 2023, Singapore

Outline of the talk

- Boolean Networks
- Reconstructability of Boolean Networks

Outline of the talk

- Boolean Networks
- Reconstructability of Boolean Networks
- Boolean Control Networks
- Reconstructability of Boolean Control Networks

Outline of the talk

- Boolean Networks
- Reconstructability of Boolean Networks
- Boolean Control Networks
- Reconstructability of Boolean Control Networks
- Finite Horizon Optimal Control problem: statement

Outline of the talk

- Boolean Networks
- Reconstructability of Boolean Networks
- Boolean Control Networks
- Reconstructability of Boolean Control Networks
- Finite Horizon Optimal Control problem: statement
- Finite Horizon Optimal Control problem: solution

BNs: from logic to algebraic representations

A Boolean network is described by the following equations

$$\begin{aligned} X(t+1) &= f(X(t)), \\ Y(t) &= h(X(t)), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{1}$$

BNs: from logic to algebraic representations

A Boolean network is described by the following equations

$$\begin{aligned} X(t+1) &= f(X(t)), \\ Y(t) &= h(X(t)), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{1}$$

$X(\cdot)$ is the n-dimensional Boolean state

BNs: from logic to algebraic representations

A Boolean network is described by the following equations

$$\begin{aligned} X(t+1) &= f(X(t)), \\ Y(t) &= h(X(t)), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{1}$$

$X(\cdot)$ is the n-dimensional Boolean state
$Y(\cdot)$ is the p-dimensional Boolean output

BNs: from logic to algebraic representations

A Boolean network is described by the following equations

$$\begin{aligned} X(t+1) &= f(X(t)), \\ Y(t) &= h(X(t)), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{1}$$

$X(\cdot)$ is the n-dimensional Boolean state
$Y(\cdot)$ is the p-dimensional Boolean output
f and h are logic functions.

BNs: from logic to algebraic representations

A Boolean network is described by the following equations

$$\begin{aligned} X(t+1) &= f(X(t)), \\ Y(t) &= h(X(t)), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{1}$$

$X(\cdot)$ is the n-dimensional Boolean state
$Y(\cdot)$ is the p-dimensional Boolean output
f and h are logic functions.

If we represent the Boolean vectors $X(t)$ and $Y(t)$ by means of their “canonical equivalent" $\mathbf{x}(t)$ and $\mathbf{y}(t)$, the BN (1) can be described as

$$\begin{aligned} \mathbf{x}(t+1) &= L\mathbf{x}(t), \\ \mathbf{y}(t) &= H\mathbf{x}(t), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{2}$$

BNs: from logic to algebraic representations

A Boolean network is described by the following equations

$$\begin{aligned} X(t+1) &= f(X(t)), \\ Y(t) &= h(X(t)), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{1}$$

$X(\cdot)$ is the n-dimensional Boolean state
$Y(\cdot)$ is the p-dimensional Boolean output
f and h are logic functions.

If we represent the Boolean vectors $X(t)$ and $Y(t)$ by means of their "canonical equivalent" $\mathbf{x}(t)$ and $\mathbf{y}(t)$, the BN (1) can be described as

$$\begin{aligned} \mathbf{x}(t+1) &= L\mathbf{x}(t), \\ \mathbf{y}(t) &= H\mathbf{x}(t), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{2}$$

where $L \in \mathcal{L}_{2^n \times 2^n}$ and $H \in \mathcal{L}_{2^p \times 2^n}$.

BNs: reconstructability definition

Definition 1 A BN (2) is reconstructable if there exists $T \in \mathbb{Z}_+$ such that the knowledge of the output trajectory $\mathbf{y}(t), t \in \{0, 1, \ldots, T\}$, allows to uniquely determine $\mathbf{x}(T)$.

BNs: reconstructability definition

Definition 1 A BN (2) is reconstructable if there exists $T \in \mathbb{Z}_+$ such that the knowledge of the output trajectory $\mathbf{y}(t), t \in \{0, 1, \dots, T\}$, allows to uniquely determine $\mathbf{x}(T)$.

In other words, a BN (2) is reconstructable if $\exists\ T \geq 0$ such that

$$y(0), y(1), \dots, y(T) \quad \underset{\text{uniquely}}{\longrightarrow} \quad x(T).$$

BNs: reconstructability characterization (1)

Every BN (2) of dimension $N = 2^n$ can be associated with a directed graph.

BNs: reconstructability characterization (1)

Every BN (2) of dimension $N = 2^n$ can be associated with a directed graph.

The graph has N vertices, say $\{1, 2, \ldots, N\}$, each of them associated to one of the possible states $\delta_N^i, i \in \{1, 2, \ldots, N\}$.

BNs: reconstructability characterization (1)

Every BN (2) of dimension $N = 2^n$ can be associated with a directed graph.

The graph has N vertices, say $\{1, 2, \ldots, N\}$, each of them associated to one of the possible states $\delta_N^i, i \in \{1, 2, \ldots, N\}$.

There is an arc, denoted by (j, i), from j to i, if $L\delta_N^j = \delta_N^i$,

BNs: reconstructability characterization (1)

Every BN (2) of dimension $N = 2^n$ can be associated with a directed graph.

The graph has N vertices, say $\{1, 2, \dots, N\}$, each of them associated to one of the possible states $\delta_N^i, i \in \{1, 2, \dots, N\}.$

There is an arc, denoted by (j, i), from j to i, if $L\delta_N^j = \delta_N^i$, namely δ_N^i is the successor of δ_N^j in the network evolution.

BNs: reconstructability characterization (1)

Every BN (2) of dimension $N = 2^n$ can be associated with a directed graph.

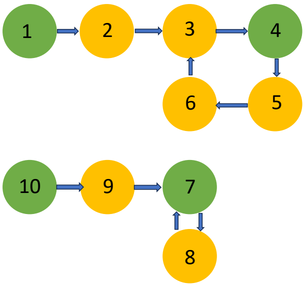
The graph has N vertices, say $\{1, 2, \ldots, N\}$, each of them associated to one of the possible states $\delta_N^i, i \in \{1, 2, \ldots, N\}.$

There is an arc, denoted by (j, i), from j to i, if $L\delta_N^j = \delta_N^i$, namely δ_N^i is the successor of δ_N^j in the network evolution.

To each vertex i we associate a output label representing the value of the output associated with the state δ_N^i, namely $y = H\delta_N^i$.

BNs: reconstructability characterization (2)

Example:



$$L = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&1&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&1&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

$$H = \begin{bmatrix} 1&0&0&1&0&0&1&0&0&1\\ 0&1&1&0&1&1&0&1&1&0 \end{bmatrix}$$

$$\mathcal{C}_1 = \{(3,4),(4,5),(5,6),(6,3)\}$$
$$\mathcal{C}_2 = \{(7,8),(8,7)\}.$$

BNs: reconstructability characterization (3)

Theorem 1 For a BN (2) the following facts are equivalent:

BNs: reconstructability characterization (3)

Theorem 1 For a BN (2) the following facts are equivalent:

- the BN is reconstructable;

BNs: reconstructability characterization (3)

Theorem 1 For a BN (2) the following facts are equivalent:

- the BN is reconstructable;
- in the graph associated with the BN, each cycle of length k identifies a periodic output label sequence of minimal period k, and distinct cycles identify distinct periodic output label sequences;

BNs: reconstructability characterization (3)

Theorem 1 For a BN (2) the following facts are equivalent:

- the BN is reconstructable;
- in the graph associated with the BN, each cycle of length k identifies a periodic output label sequence of minimal period k, and distinct cycles identify distinct periodic output label sequences;
- for every pair of distinct periodic state trajectories of the same minimal period k, described by the two ordered $k$tuples

$$(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k) \neq (\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \ldots, \bar{\mathbf{x}}_k),$$

BNs: reconstructability characterization (3)

Theorem 1 For a BN (2) the following facts are equivalent:

- the BN is reconstructable;
- in the graph associated with the BN, each cycle of length k identifies a periodic output label sequence of minimal period k, and distinct cycles identify distinct periodic output label sequences;
- for every pair of distinct periodic state trajectories of the same minimal period k, described by the two ordered $k$tuples

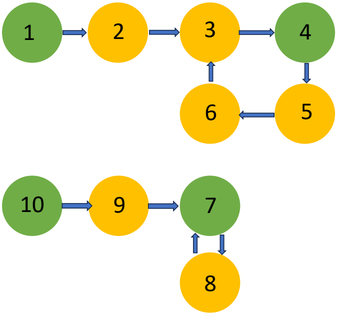
$$(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k) \neq (\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \dots, \bar{\mathbf{x}}_k),$$

the corresponding output trajectories are periodic of minimal period k and described by two different ordered $k$tuples, i.e.

$$(H\mathbf{x}_1, H\mathbf{x}_2, \dots, H\mathbf{x}_k) \neq (H\bar{\mathbf{x}}_1, H\bar{\mathbf{x}}_2, \dots, H\bar{\mathbf{x}}_k)$$

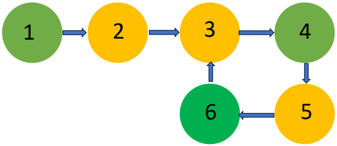
BNs: reconstructability characterization (4)

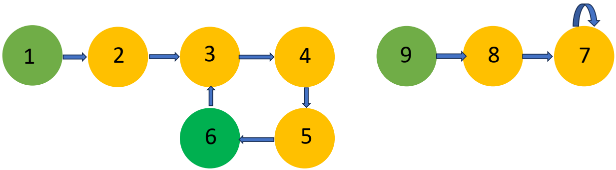
Example:



BNs: reconstructability characterization (5)

Examples:





BNs: remarks on reconstructability (1)

Every logical matrix $L \in \mathcal{L}_{2^n \times 2^n}$ can be brought, by means of row-column permutations, to the following form:

$$L = \begin{bmatrix} W & 0 \\ T & C \end{bmatrix},$$

BNs: remarks on reconstructability (1)

Every logical matrix $L \in \mathcal{L}_{2^n \times 2^n}$ can be brought, by means of row-column permutations, to the following form:

$$L = \begin{bmatrix} W & 0 \\ T & C \end{bmatrix},$$

where W is nilpotent and

$$C = \begin{bmatrix} C_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C_r \end{bmatrix}$$

BNs: remarks on reconstructability (1)

Every logical matrix $L \in \mathcal{L}_{2^n \times 2^n}$ can be brought, by means of row-column permutations, to the following form:

$$L = \begin{bmatrix} W & 0 \\ T & C \end{bmatrix},$$

where W is nilpotent and

$$C = \begin{bmatrix} C_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C_r \end{bmatrix}$$

each C_i being a cyclic permutation matrix.

BNs: remarks on reconstructability (2)

We can accordingly partition H as

$$H = \begin{bmatrix} H_W & H_C \end{bmatrix}.$$

BNs: remarks on reconstructability (2)

We can accordingly partition H as

$$H = \begin{bmatrix} H_W & H_C \end{bmatrix}.$$

The BN described by (L, H) is reconstructable if and only if all columns of the matrix

$$\mathcal{O}_C := \begin{bmatrix} H_C \\ H_C C \\ \vdots \\ H_C C^{N-1} \end{bmatrix}$$

are distinct.

BCNs: from logic to algebraic representations (1)

A Boolean control network (BCN) is described by the following equations

$$\begin{aligned} X(t+1) &= f(X(t), U(t)), \\ Y(t) &= h(X(t)), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{3}$$

BCNs: from logic to algebraic representations (1)

A Boolean control network (BCN) is described by the following equations

$$\begin{aligned} X(t+1) &= f(X(t), U(t)), \\ Y(t) &= h(X(t)), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{3}$$

$X(\cdot)$, $U(\cdot)$ and $Y(\cdot)$ are the Boolean state (dim = n), input (dim = m), and output (dim = p)

BCNs: from logic to algebraic representations (1)

A Boolean control network (BCN) is described by the following equations

$$\begin{aligned} X(t+1) &= f(X(t), U(t)), \\ Y(t) &= h(X(t)), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{3}$$

$X(\cdot)$, $U(\cdot)$ and $Y(\cdot)$ are the Boolean state (dim = n), input (dim = m), and output (dim = p)
f and h are logic functions.

BCNs: from logic to algebraic representations (1)

A Boolean control network (BCN) is described by the following equations

$$\begin{aligned} X(t+1) &= f(X(t), U(t)), \\ Y(t) &= h(X(t)), \qquad t \in \mathbb{Z}_+, \end{aligned} \tag{3}$$

$X(\cdot)$, $U(\cdot)$ and $Y(\cdot)$ are the Boolean state (dim = n), input (dim = m), and output (dim = p)
f and h are logic functions.

If we represent the Boolean vectors by means of their "canonical equivalent", the BCN (3) can be described as

$$\begin{aligned} \mathbf{x}(t+1) &= L \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad t \in \mathbb{Z}_+, \\ \mathbf{y}(t) &= H\mathbf{x}(t) \end{aligned} \tag{4}$$

where $L \in \mathcal{L}_{2^n \times 2^{(n+m)}}$ and $H \in \mathcal{L}_{2^p \times 2^n}$.

BCNs: reconstructability definition

Definition 2 A BCN (4) is reconstructable if if there exists $T \in \mathbb{Z}_+$ such that, for every input sequence and every initial condition $\mathbf{x}(0)$, the knowledge of the input and of the corresponding output trajectory, $\mathbf{u}(t)$ and $\mathbf{y}(t), t \in \{0, 1, \dots, T\}$, allows to uniquely determine $\mathbf{x}(T)$.

BCNs: reconstructability definition

Definition 2 A BCN (4) is reconstructable if if there exists $T \in \mathbb{Z}_+$ such that, for every input sequence and every initial condition $\mathbf{x}(0)$, the knowledge of the input and of the corresponding output trajectory, $\mathbf{u}(t)$ and $\mathbf{y}(t), t \in \{0, 1, \dots, T\}$, allows to uniquely determine $\mathbf{x}(T)$.

More in detail, a BCN (4) is reconstructable if $\exists\, T \geq 0$ such that

$$\left\{ \begin{array}{c} u(0), u(1), \dots, u(T-1) \\ y(0), y(1), \dots, y(T) \end{array} \right\} \underset{\text{uniquely}}{\longrightarrow} x(T).$$

BCNs: reconstructability characterization (1)

Every BCN (4) of dimension $N = 2^n$ with $M = 2^m$ inputs can be associated with a directed graph.

BCNs: reconstructability characterization (1)

Every BCN (4) of dimension $N = 2^n$ with $M = 2^m$ inputs can be associated with a directed graph.

The graph has N vertices, say $\{1, 2, \dots, N\}$, each of them associated to one of the possible states $\delta_N^i, i \in \{1, 2, \dots, N\}$.

BCNs: reconstructability characterization (1)

Every BCN (4) of dimension $N = 2^n$ with $M = 2^m$ inputs can be associated with a directed graph.

The graph has N vertices, say $\{1, 2, \ldots, N\}$, each of them associated to one of the possible states $\delta_N^i, i \in \{1, 2, \ldots, N\}$.

There is an arc of type k, denoted by $(j, i)_k$, from j to i, if $L \ltimes \delta_M^k \ltimes \delta_N^j = \delta_N^i$.

BCNs: reconstructability characterization (1)

Every BCN (4) of dimension $N = 2^n$ with $M = 2^m$ inputs can be associated with a directed graph.

The graph has N vertices, say $\{1, 2, \dots, N\}$, each of them associated to one of the possible states $\delta_N^i, i \in \{1, 2, \dots, N\}$.

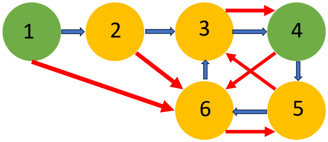
There is an arc of type k, denoted by $(j, i)_k$, from j to i, if $L \ltimes \delta_M^k \ltimes \delta_N^j = \delta_N^i$.

To each state i we associate a label representing the value of the output associated with it, namely $y = H\delta_N^i$.

BCNs: reconstructability characterization (2)

$$L \ltimes \delta_2^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$



$$L \ltimes \delta_2^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

BCNs: reconstructability characterization (3)

Theorem 2 For a BCN (4) the following facts are equivalent:

BCNs: reconstructability characterization (3)

Theorem 2 For a BCN (4) the following facts are equivalent:

- the BCN is reconstructable

BCNs: reconstructability characterization (3)

Theorem 2 For a BCN (4) the following facts are equivalent:

- the BCN is reconstructable
- in the graph associated with the BCN, distinct cycles, consisting of distinct vertices, but having the same type of arcs,

$$(i_1, i_2)_{k_1}, (i_2, i_3)_{k_2}, \dots, (i_h, i_1)_{k_h}, \qquad (j_1, j_2)_{k_1}, (j_2, j_3)_{k_2}, \dots, (j_h, j_1)_{k_h},$$

identify distinct periodic output label sequences;

BCNs: reconstructability characterization (3)

Theorem 2 For a BCN (4) the following facts are equivalent:

• the BCN is reconstructable

• in the graph associated with the BCN, distinct cycles, consisting of distinct vertices, but having the same type of arcs,

$$(i_1, i_2)_{k_1}, (i_2, i_3)_{k_2}, \dots, (i_h, i_1)_{k_h}, \qquad (j_1, j_2)_{k_1}, (j_2, j_3)_{k_2}, \dots, (j_h, j_1)_{k_h},$$

identify distinct periodic output label sequences;

• for every pair of distinct periodic state-input trajectories of the same minimal period k, described by the $k$tuples

$$((\mathbf{x}_1, \mathbf{u}_1), \dots, (\mathbf{x}_k, \mathbf{u}_k)) \neq ((\bar{\mathbf{x}}_1, \mathbf{u}_1), \dots, (\bar{\mathbf{x}}_k, \mathbf{u}_k)),$$

BCNs: reconstructability characterization (3)

Theorem 2 For a BCN (4) the following facts are equivalent:

- the BCN is reconstructable

- in the graph associated with the BCN, distinct cycles, consisting of distinct vertices, but having the same type of arcs,

$$(i_1, i_2)_{k_1}, (i_2, i_3)_{k_2}, \dots, (i_h, i_1)_{k_h}, \qquad (j_1, j_2)_{k_1}, (j_2, j_3)_{k_2}, \dots, (j_h, j_1)_{k_h},$$

identify distinct periodic output label sequences;

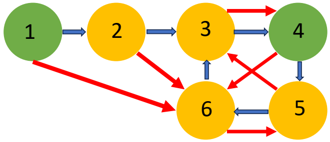
- for every pair of distinct periodic state-input trajectories of the same minimal period k, described by the $k$tuples

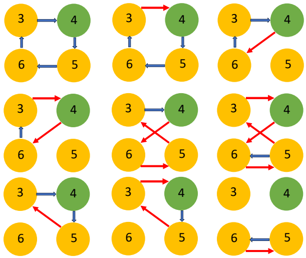
$$((\mathbf{x}_1, \mathbf{u}_1), \dots, (\mathbf{x}_k, \mathbf{u}_k)) \neq ((\bar{\mathbf{x}}_1, \mathbf{u}_1), \dots, (\bar{\mathbf{x}}_k, \mathbf{u}_k)),$$

the corresponding output trajectories are periodic of minimal period k and described by two different $k$tuples, i.e.

$$(H\mathbf{x}_1, H\mathbf{x}_2, \dots, H\mathbf{x}_k) \neq (H\bar{\mathbf{x}}_1, H\bar{\mathbf{x}}_2, \dots, H\bar{\mathbf{x}}_k).$$

BCNs: reconstructability characterization (4)





The BCN is reconstructable.

Finite Horizon Optimal Control: Problem statement

Problem: Given a BCN (4), with initial state $\mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{L}_N$,

Finite Horizon Optimal Control: Problem statement

Problem: Given a BCN (4), with initial state $\mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{L}_N$, determine an input sequence that minimizes the cost function:

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathcal{Q}_f(\mathbf{x}(T)) + \sum_{t=0}^{T-1} \mathcal{Q}(\mathbf{u}(t), \mathbf{x}(t)), \tag{5}$$

Finite Horizon Optimal Control: Problem statement

Problem: Given a BCN (4), with initial state $\mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{L}_N$, determine an input sequence that minimizes the cost function:

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathcal{Q}_f(\mathbf{x}(T)) + \sum_{t=0}^{T-1} \mathcal{Q}(\mathbf{u}(t), \mathbf{x}(t)), \tag{5}$$

where
$\mathcal{Q}_f(\cdot)$ is any function defined on Δ_N,

Finite Horizon Optimal Control: Problem statement

Problem: Given a BCN (4), with initial state $\mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{L}_N$, determine an input sequence that minimizes the cost function:

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathcal{Q}_f(\mathbf{x}(T)) + \sum_{t=0}^{T-1} \mathcal{Q}(\mathbf{u}(t), \mathbf{x}(t)), \tag{5}$$

where
$\mathcal{Q}_f(\cdot)$ is any function defined on Δ_N, and
$\mathcal{Q}(\cdot,\cdot)$ is any function defined on $\Delta_M \times \Delta_N$,

Finite Horizon Optimal Control: Problem statement

Problem: Given a BCN (4), with initial state $\mathbf{x}(0) = \mathbf{x}_0 \in \mathcal{L}_N$, determine an input sequence that minimizes the cost function:

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathcal{Q}_f(\mathbf{x}(T)) + \sum_{t=0}^{T-1} \mathcal{Q}(\mathbf{u}(t), \mathbf{x}(t)), \qquad (5)$$

where
$\mathcal{Q}_f(\cdot)$ is any function defined on Δ_N, and
$\mathcal{Q}(\cdot, \cdot)$ is any function defined on $\Delta_M \times \Delta_N$,
$N = 2^n$, $M = 2^m$.

By exploiting the fact that the domain of $\mathcal{Q}_f(\cdot)$ and $\mathcal{Q}(\cdot,\cdot)$ is a finite set,

By exploiting the fact that the domain of $\mathcal{Q}_f(\cdot)$ and $\mathcal{Q}(\cdot,\cdot)$ is a finite set, we can equivalently rewrite the index $J_T(\mathbf{x}_0, \mathbf{u}(\cdot))$ as:

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{c}_f^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad (6)$$

where $\mathbf{c}_f \in \mathbb{R}^N$ and $\mathbf{c} \in \mathbb{R}^{NM}$.

By exploiting the fact that the domain of $\mathcal{Q}_f(\cdot)$ and $\mathcal{Q}(\cdot,\cdot)$ is a finite set, we can equivalently rewrite the index $J_T(\mathbf{x}_0, \mathbf{u}(\cdot))$ as:

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{c}_f^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad (6)$$

where $\mathbf{c}_f \in \mathbb{R}^N$ and $\mathbf{c} \in \mathbb{R}^{NM}$.
Also, for every choice of $\alpha, \beta \in \mathbb{R}$, the input sequence that minimizes the cost function (6) (for any given $\mathbf{x}_0$) is the same one that minimizes

$$\tilde{J}_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = [\mathbf{c}_f + \alpha \mathbf{1}_N]^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} [\mathbf{c} + \beta \mathbf{1}_{NM}]^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t),$$

By exploiting the fact that the domain of $\mathcal{Q}_f(\cdot)$ and $\mathcal{Q}(\cdot,\cdot)$ is a finite set, we can equivalently rewrite the index $J_T(\mathbf{x}_0, \mathbf{u}(\cdot))$ as:

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{c}_f^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad (6)$$

where $\mathbf{c}_f \in \mathbb{R}^N$ and $\mathbf{c} \in \mathbb{R}^{NM}$.

Also, for every choice of $\alpha, \beta \in \mathbb{R}$, the input sequence that minimizes the cost function (6) (for any given $\mathbf{x}_0$) is the same one that minimizes

$$\tilde{J}_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = [\mathbf{c}_f + \alpha \mathbf{1}_N]^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} [\mathbf{c} + \beta \mathbf{1}_{NM}]^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t),$$

where $\mathbf{1}_k$ is the k-dimensional vector with all unitary entries.

By exploiting the fact that the domain of $\mathcal{Q}_f(\cdot)$ and $\mathcal{Q}(\cdot,\cdot)$ is a finite set, we can equivalently rewrite the index $J_T(\mathbf{x}_0, \mathbf{u}(\cdot))$ as:

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{c}_f^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \qquad (6)$$

where $\mathbf{c}_f \in \mathbb{R}^N$ and $\mathbf{c} \in \mathbb{R}^{NM}$.

Also, for every choice of $\alpha, \beta \in \mathbb{R}$, the input sequence that minimizes the cost function (6) (for any given $\mathbf{x}_0$) is the same one that minimizes

$$\tilde{J}_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = [\mathbf{c}_f + \alpha \mathbf{1}_N]^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} [\mathbf{c} + \beta \mathbf{1}_{NM}]^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t),$$

where $\mathbf{1}_k$ is the k-dimensional vector with all unitary entries.
So, one can assume that in (6)
the weight vectors $\mathbf{c}_f$ and $\mathbf{c}$ are nonnegative.

Finite Horizon Optimal Control: Problem solution

For every choice of a family of real vectors $\mathbf{m}(t), t \in [0, T]$, and every state trajectory of the BCN $\mathbf{x}(t), t \in [0, T]$, one has

$$\begin{aligned} 0 &= \sum_{t=0}^{T-1} [\mathbf{m}(t+1)^\top \mathbf{x}(t+1) - \mathbf{m}(t)^\top \mathbf{x}(t)] \\ &+ \mathbf{m}(0)^\top \mathbf{x}(0) - \mathbf{m}(T)^\top \mathbf{x}(T). \end{aligned}$$

Finite Horizon Optimal Control: Problem solution

For every choice of a family of real vectors $\mathbf{m}(t), t \in [0, T]$, and every state trajectory of the BCN $\mathbf{x}(t), t \in [0, T]$, one has

$$\begin{aligned} 0 &= \sum_{t=0}^{T-1} [\mathbf{m}(t+1)^\top \mathbf{x}(t+1) - \mathbf{m}(t)^\top \mathbf{x}(t)] \\ &+ \mathbf{m}(0)^\top \mathbf{x}(0) - \mathbf{m}(T)^\top \mathbf{x}(T). \end{aligned}$$

Consequently, the cost function can be equivalently rewritten

$$\begin{aligned} J(\mathbf{x}_0, \mathbf{u}(\cdot)) &= \mathbf{m}(0)^\top \mathbf{x}(0) + [\mathbf{c}_f - \mathbf{m}(T)]^\top \mathbf{x}(T) \\ &+ \sum_{t=0}^{T-1} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) + \sum_{t=0}^{T-1} [\mathbf{m}(t+1)^\top \mathbf{x}(t+1) - \mathbf{m}(t)^\top \mathbf{x}(t)]. \end{aligned}$$

By making use of the state update equation of the BCN (4) and of the fact that, for every choice of $\mathbf{u}(t) \in \mathcal{L}_M$, one has

$$\mathbf{m}(t)^\top \mathbf{x}(t) = \begin{bmatrix} \mathbf{m}(t)^\top & \mathbf{m}(t)^\top & \dots & \mathbf{m}(t)^\top \end{bmatrix} \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t),$$

By making use of the state update equation of the BCN (4) and of the fact that, for every choice of $\mathbf{u}(t) \in \mathcal{L}_M$, one has

$$\mathbf{m}(t)^\top \mathbf{x}(t) = \begin{bmatrix} \mathbf{m}(t)^\top & \mathbf{m}(t)^\top & \dots & \mathbf{m}(t)^\top \end{bmatrix} \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t),$$

we get this final version of the cost function:

$$J(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{m}(0)^\top \mathbf{x}(0) + [\mathbf{c}_f - \mathbf{m}(T)]^\top \mathbf{x}(T)$$

$$+ \sum_{t=0}^{T-1} \left(\mathbf{c}^\top + \mathbf{m}(t+1)^\top L - \begin{bmatrix} \mathbf{m}(t)^\top & \dots & \mathbf{m}(t)^\top \end{bmatrix} \right) \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t).$$

Set and

$$L = \begin{bmatrix} L_1 & L_2 & \dots & L_M \end{bmatrix}$$

Set and

$$L = \begin{bmatrix} L_1 & L_2 & \dots & L_M \end{bmatrix}$$

and

$$\mathbf{c}^\top = \begin{bmatrix} \mathbf{c}_1^\top & \mathbf{c}_2^\top & \dots & \mathbf{c}_M^\top \end{bmatrix}.$$

Set and

$$L = \begin{bmatrix} L_1 & L_2 & \dots & L_M \end{bmatrix}$$

and

$$\mathbf{c}^\top = \begin{bmatrix} \mathbf{c}_1^\top & \mathbf{c}_2^\top & \dots & \mathbf{c}_M^\top \end{bmatrix}.$$

Then, the term in the summation becomes:

$$\left(\mathbf{c}^\top + \mathbf{m}(t+1)^\top L - \begin{bmatrix} \mathbf{m}(t)^\top & \dots & \mathbf{m}(t)^\top \end{bmatrix} \right) \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) =$$

$$\begin{bmatrix} \mathbf{c}_1^\top + \mathbf{m}(t+1)^\top L_1 - \mathbf{m}(t)^\top & \dots & \mathbf{c}_M^\top + \mathbf{m}(t+1)^\top L_M - \mathbf{m}(t)^\top \end{bmatrix}$$

$$\ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t).$$

Now, since the vectors $\mathbf{m}(t), t \in [0, T]$, can be freely chosen without affecting the value of the index, we choose them according to the following ALGORITHM:

Now, since the vectors $\mathbf{m}(t), t \in [0, T]$, can be freely chosen without affecting the value of the index, we choose them according to the following ALGORITHM:

- [Initialization] Set $\mathbf{m}(T) := \mathbf{c}_f$;

Now, since the vectors $\mathbf{m}(t), t \in [0, T]$, can be freely chosen without affecting the value of the index, we choose them according to the following ALGORITHM:

- [Initialization] Set $\mathbf{m}(T) := \mathbf{c}_f$;
- [Recursion] For $t = T-1, T-2, \dots, 1, 0$, the jth entry of the vector $\mathbf{m}(t)$ is chosen according to the recursive rule:

$$[\mathbf{m}(t)]_j := \min_{i \in [1,M]} \Big([\mathbf{c}_i^\top + \mathbf{m}(t+1)^\top L_i]_j \Big), \ \forall\, j \in [1, N].$$

As a result, the index takes the form

$$J(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{m}(0)^\top \mathbf{x}(0)$$

$$+ \sum_{t=0}^{T-1} \begin{bmatrix} \mathbf{w}_1(t)^\top & \mathbf{w}_2(t)^\top & \dots & \mathbf{w}_M(t)^\top \end{bmatrix} \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t)$$

and for each $j \in [1, N]$, there is some row vector $\mathbf{w}_i(t)$ whose jth entry is zero, while the jth entry of all the other row vectors $\mathbf{w}_k(t)$ is nonnegative.

As a result, the index takes the form

$$J(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{m}(0)^\top \mathbf{x}(0)$$

$$+ \sum_{t=0}^{T-1} \begin{bmatrix} \mathbf{w}_1(t)^\top & \mathbf{w}_2(t)^\top & \dots & \mathbf{w}_M(t)^\top \end{bmatrix} \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t)$$

and for each $j \in [1, N]$, there is some row vector $\mathbf{w}_i(t)$ whose jth entry is zero, while the jth entry of all the other row vectors $\mathbf{w}_k(t)$ is nonnegative.

Therefore the cost function is minimized by any input sequence $\mathbf{u}(t), t \in [0, T-1]$, that is obtained according to this rule:

As a result, the index takes the form

$$J(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{m}(0)^\top \mathbf{x}(0)$$

$$+ \sum_{t=0}^{T-1} \begin{bmatrix} \mathbf{w}_1(t)^\top & \mathbf{w}_2(t)^\top & \dots & \mathbf{w}_M(t)^\top \end{bmatrix} \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t)$$

and for each $j \in [1, N]$, there is some row vector $\mathbf{w}_i(t)$ whose jth entry is zero, while the jth entry of all the other row vectors $\mathbf{w}_k(t)$ is nonnegative.

Therefore the cost function is minimized by any input sequence $\mathbf{u}(t), t \in [0, T-1]$, that is obtained according to this rule:

$$\mathbf{x}(t) = \delta_N^j \qquad \longrightarrow \qquad \mathbf{u}(t) = \delta_M^{i^*(j,t)},$$

where

$$i^*(j,t) = \arg \min_{i \in [1,M]} [\mathbf{w}_i(t)]_j = \arg \min_{i \in [1,M]} \Big([\mathbf{c}_i + \mathbf{m}(t+1)^\top L_i]_j \Big).$$

In this way,

- $J^*(\mathbf{x}_0) = \min_{\mathbf{u}(\cdot)} J(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{m}(0)^\top \mathbf{x}_0,$

In this way,

- $J^*(\mathbf{x}_0) = \min_{\mathbf{u}(\cdot)} J(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{m}(0)^\top \mathbf{x}_0,$
  where $\mathbf{m}(0)$ is obtained according to the previous algorithm.

In this way,

- $J^*(\mathbf{x}_0) = \min_{\mathbf{u}(\cdot)} J(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{m}(0)^\top \mathbf{x}_0$,
  where $\mathbf{m}(0)$ is obtained according to the previous algorithm.
- The optimal control input can be implemented by means of a time-varying feedback law:

$$\mathbf{u}(t) = K(t)\mathbf{x}(t),$$

In this way,

- $J^*(\mathbf{x}_0) = \min_{\mathbf{u}(\cdot)} J(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{m}(0)^\top \mathbf{x}_0,$
  where $\mathbf{m}(0)$ is obtained according to the previous algorithm.
- The optimal control input can be implemented by means of a time-varying feedback law:

$$\mathbf{u}(t) = K(t)\mathbf{x}(t),$$

  where the (not necessarily unique) feedback matrix is expressed as

$$K(t) = \begin{bmatrix} \delta_M^{i^*(1,t)} & \delta_M^{i^*(2,t)} & \dots & \delta_M^{i^*(N,t)} \end{bmatrix}.$$

Thanks for your attention!

Questions?