Stochastic Model Predictive Control

Pantelis Sopasakis

IMT Institute for Advanced Studies Lucca

February 10, 2016

Outline

1. Intro: stochastic optimal control
2. Classification of SMPC approaches
3. Scenario based SMPC
4. Affine disturbance feedback

I. Introduction

- ✓ Stochastic optimal control
- ✓ Control policies
- ✓ Dynamic programming

Stochastic optimal control

Stochastic optimal control lies at the core of every stochastic MPC formulation.

Stochastic optimal control

Uncertain dynamical system:

$$x_{k+1} = f(x_k, u_k, w_k),$$

where w_k lives in a probability space $(\Omega_k, \mathcal{F}_k, \mathrm{P}_k)$[1].

In stochastic optimal control, we get take our decision $u_{k+j|k}$ at future time $k+j$ taking into account the available information up to that time.

[1] The probability distribution function of w_k may be a function of x_k and u_k, that is $\mathrm{P} = \mathrm{P}(\mathrm{d}w_k \mid x_k, u_k)$. See Bertsekas and Shreve, 1978.

Stochastic OC + Causality = ♡

At $k = j$ we observe x_j and u_j and we decide the control action using

- The initial information x_0 (and w_0),
- The current observation, that is x_j (and w_j)
- The history of control actions

Overall...

$$u_j = \mu_j(x_0, w_0, u_0, \ldots, u_{j-1}, x_j, w_j).$$

We thus construct the space $\Pi_N = (\mu_0, \ldots, \mu_{N-1})$ of (causal) control policies. In some cases it suffices to assume[2]

$$u_j = \mu_j(x_j).$$

[2] These are called *Markov* policies.

Stochastic OC + Causality = ♡

Assume we can observe w_k at time k:

- $k = 0$ Observe x_0, w_0

Stochastic OC + Causality = ♡

Assume we can observe w_k at time k:

- $k = 0$ Observe x_0, w_0
- $k = 0$ Decide $u_0 = \mu_0(x_0, w_0)$

Stochastic OC + Causality = ♡

Assume we can observe w_k at time k:

- $k = 0$ Observe x_0, w_0
- $k = 0$ Decide $u_0 = \mu_0(x_0, w_0)$
- $k = 1$ System response $x_1 = f(x_0, w_0, u_0)$

Stochastic OC + Causality = ♡

Assume we can observe w_k at time k:

- $k = 0$ Observe x_0, w_0
- $k = 0$ Decide $u_0 = \mu_0(x_0, w_0)$
- $k = 1$ System response $x_1 = f(x_0, w_0, u_0)$
- $k = 1$ Observe x_1, w_1

Stochastic OC + Causality = ♡

Assume we can observe w_k at time k:

- $k = 0$ Observe x_0, w_0
- $k = 0$ Decide $u_0 = \mu_0(x_0, w_0)$
- $k = 1$ System response $x_1 = f(x_0, w_0, u_0)$
- $k = 1$ Observe x_1, w_1
- $k = 1$ Decide $u_1 = \mu_1(x_0, w_0, u_0, x_1)$

Stochastic OC + Causality = ♡

Assume we can observe w_k at time k:

- $k = 0$ Observe x_0, w_0
- $k = 0$ Decide $u_0 = \mu_0(x_0, w_0)$
- $k = 1$ System response $x_1 = f(x_0, w_0, u_0)$
- $k = 1$ Observe x_1, w_1
- $k = 1$ Decide $u_1 = \mu_1(x_0, w_0, u_0, x_1)$
- $k = 2$ System response $x_2 = f(x_1, u_1, w_1)$

Stochastic OC + Causality = ♡

Assume we can observe w_k at time k:

- $k = 0$ Observe x_0, w_0
- $k = 0$ Decide $u_0 = \mu_0(x_0, w_0)$
- $k = 1$ System response $x_1 = f(x_0, w_0, u_0)$
- $k = 1$ Observe x_1, w_1
- $k = 1$ Decide $u_1 = \mu_1(x_0, w_0, u_0, x_1)$
- $k = 2$ System response $x_2 = f(x_1, u_1, w_1)$
- $k = 2$ Observe x_2, w_2 ...

Stochastic optimal control

Hereafter we assume $u_k = \mu(x_k)$[3].

Three equivalent formulations:

1. In **nested** form
2. Over a **product** probability space
3. As a **dynamic programming** recursion

[3] This is an essential assumption to formulate the stochastic OCP as a DP recursion. This way, u_k is computed at time k without using historical information of the process, i.e., any of $w_0, w_1, \ldots, w_{k-1}$.

Nested formulation

Formulation as a *nested* problem: The total cost function is (where $\pi = (\mu_0, \mu_1, \ldots, \mu_{N-1})$ with $u_i = \mu_i(x_i)$)[4]

$$V_N(x_0, \pi) = \mathbb{E}_{w_0}\Big[\ell_0(x_0, \mu_0(x_0), w_0) + \mathbb{E}_{w_1}\big[\ell_1(x_1, \mu_1(x_1), w_1) + \mathbb{E}_{w_2}[\cdots + \mathbb{E}_{w_{N-1}}[\ell_{N-1}(x_{N-1}, \mu_{N-1}(x_{N-1}), w_{N-1}) \mid x_{N-1}, \mu_{N-1}(x_{N-1})] \mid \cdots]\big] \mid x_0, \mu_0(x_0)\Big],$$

where the states x_k satisfy

$$x_{k+1} = f(x_k, \mu_k(x_k), w_k)$$

for $k \in \mathbb{N}_{[0,N-2]}$.

[4]It's easy to wedge in a *terminal* cost function of the form $\ell_N(x_N, u_N, w_N) = V_f(x_N, w_N)$.

Product space formulation

We can use the following result to rearrange the terms in V_N. For every measure space $(\Omega, \mathcal{F}, \mathrm{P})$ and measurable $h : \Omega \to \bar{\mathbb{R}}$ and $\lambda \in (-\infty, \infty]$ we have

$$\lambda + \int h \mathrm{d}\,\mathrm{P} = \int (\lambda + h) \mathrm{dP}.$$

And recall that the expectation of a random variable h on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$ is given by the Lebesgue integral

$$\mathbb{E}[h] = \int h \mathrm{dP}.$$

Product space formulation

Assume $\ell_k > -\infty$. Then, it is

$$V_N(x_0,\pi) = \mathbb{E}_{w_0}\Big[\mathbb{E}_{w_1}\Big[\mathbb{E}_{w_2}\big[\dots\mathbb{E}_{w_{N-1}}\big[\sum_{k=0}^{N-1}\ell_k(x_k,\mu_k(x_k),w_k) \mid x_{N-1},\mu_{N-1}(x_{N-1})\big] \mid \cdots\Big] \mid x_0,\mu_0(x_0)\Big],$$

* The product probability space

To define a product probability space, we need to introduce the notion of a **p-system on** Ω[5] which is a collection of sets $\mathcal{A}$ so that

$$A_1, A_2 \in \mathcal{A} \Rightarrow A_1 \cap A_2 \in \mathcal{A}.$$

Example: On $\mathbb{R}$, the class $\mathcal{A} = \{(-\infty, b], b \in \mathbb{R}\}$ is a p-system.

[5] p for product; aka π-system or pi-system.

* The product probability space

If two (probability) measures coincide on a p-system, they coincide everywhere, thus, it suffices to define a measure on a p-system.

Recall that the cartesian product of two set A, B is

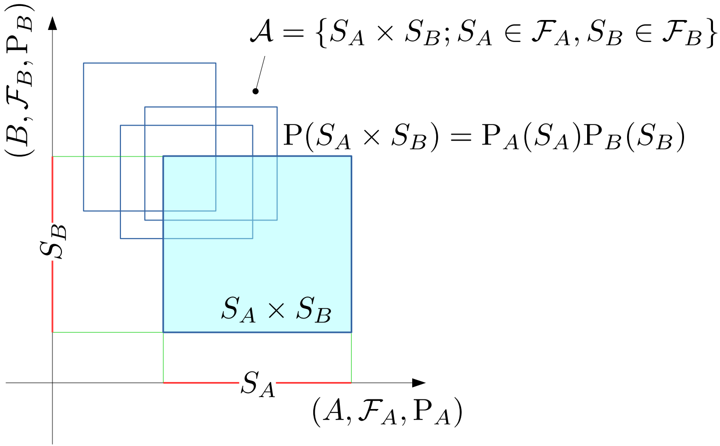
$$A \times B \ni (a, b); a \in A, b \in B.$$

Let $(A, \mathcal{F}_A)$ and $(B, \mathcal{F}_B)$ be measurable spaces; then let

$$\mathcal{A} = \{S_A \times S_B; S_A \in \mathcal{F}_A, S_B \in \mathcal{F}_B\}$$

To define a (prob.) measure on $A \times B$ it suffices to define it on $\mathcal{A}$.

* The product probability space



Product space formulation

If the conditions of Fubini's Theorem are satisfied[6], then

$$V_N(x_0, \pi) = \mathbb{E}\left[\sum_{k=0}^{N-1} \ell_k(x_k, \mu_k(x_k), w_k),\right]$$

where $\mathbb{E}$ is the expectation operator in the product measure space of $(\Omega_k, \mathcal{F}_k, \mathrm{P}_k)$ for $k \in \mathbb{N}_{[0,N-1]}$ and the states $x_1, x_2, \ldots, x_{N-1}$ are functions of x_0 and $w_0, w_1, \ldots, w_{N-1}$ satisfying the system dynamics.

[6]What are these conditions? See: R. Ash, *Real analysis and probability*, Academic press, 1972.

DP recursion

It follows from the nested formulation of V_N that the DP recursion is

$$V_0^\star(x) = 0,$$

and for $j = 0, \ldots, N-1$,

$$V_{j+1}^\star(x) = \inf_{u \in U_{N-j}(x)} \mathbb{E}_{w_{N-j}} \Big[\ell_{N-j}(x, u, w_{N-j}) + V_j^\star(f(x, u, w_{N-j})) \Big].$$

Notice that tacitly we have assumed $u_k = \mu_k(x_k)$; this is a *condicio sine qua non*[7] for DP.

[7] In Latin, *condicio sine qua non* refers to an indispensable and essential action, condition, or ingredient. We will study later the case of scenario trees where we can aberrate from this rule.

Stochastic programming vs DP

1. In DP we are bound to assume $u_k = \mu_k(x_k)$[8]; in stochastic programming we can have $u_k = \mu_k(x_0, w_0, \ldots, w_{k-1}, x_k)$
2. In DP we assume that the underlying random process $w_0, w_1, \ldots, w_{N-1}$ is stagewise independent.
3. There are cases where we can use apply DP without assuming stagewise independence; e.g., scenario trees (later).

[8] Such policies are known as *Markovian*. Whether a non-Markovian policy can be better than a Markovian one is a nontrivial question which is treated in Bertsekas & Shreve, 1978.

Remarks

Stochastic programming problems are very difficult to solve even for (ostensibly) simple cases such as unconstrained linear systems.

We usually have to resort to simplifying assumptions, such as:

1. Assume that the underlying process is
   1.1 iid
   1.2 iid and normal
2. Discretisation of probability distributions (scenario trees)
3. Optimise over Markovian policies only, i.e., $u_k = \mu_k(x_k)$
4. Optimise over semi-Markovian policies only, i.e., $u_k = \mu_k(x_0, x_k)$
5. Parametrisation of inputs, e.g., $u_k = \sum_{i=0}^{k-1} H_i w_i + h_i$

A little exercise

Assume $\ell_k(\cdot, \cdot, w)$ are *convex* for all $w \in \Omega_k$ and the system dynamics is linear,

$$x_{k+1} = A(w_k)x_k + B(w_k)u_k + d(w_k).$$

We impose the constraints $u_k \in U$ where U is a nonempty convex closed set. Assume that w_k are stagewise independent and $u_k = \mu_k(x_k)$.

Show that $V_N^\star(x)$ is a *convex* function.

Exercise

Assume $\ell_k(x,u,w) = x'Q_kx + u'R_ku$, with $Q_k \in S_+^n$, $R_k \in S_{++}^n$ and the system dynamics is given by

$$x_{k+1} = A_kx_k + Bu_k + v_k$$

where $A_k \sim \mathcal{MN}_{n\times n}(\bar{A}_k, U_k, V_k)$[9] and $v_k \sim \mathcal{N}_n(\bar{d}_k, \Sigma_k)$; A_k and v_k are *independent* and neither of those is known at time k. Determine $V_2^\star(x)$ using DP.

[9] $A(w_k)$ is a random matrix and it follows the *matrix normal* distribution whose definition and many useful properties can be found in: A.K. Gupta and D.K. Nagar, *Matrix variate distributions*, Chapman & Hall, 2000.

Further reading

1. D.P. Bertsekas and S.E. Shreve, *Stochastic optimal control: the discrete time case*, Academic press, 1978.
2. A. Shapiro, D. Dentcheva and A.Ruszczyński, *Lectures on stochastic programming – modeling and theory*, MPS-SIAM series on optimization, 2009.

II. SMPC taxonomy

System dynamics

✓ Linear

$$x_{k+1} = A(w_k)x_k + B(w_k)u_k + d(w_k)$$

✓ Nonlinear

$$x_{k+1} = f(x_k, u_k, w_k)$$

Type of uncertainty

- ✓ Additive

$$x_{k+1} = f(x_k, u_k) + w_k$$

- ✓ Parametric (linear case)

$$x_{k+1} = A(w_k)x_k + B(w_k)u_k$$

- ✓ Both

Uncertainty over time #1

- ✓ Time-varying

$$x_{k+1} = f(x_k, u_k, w_k),$$

- ✓ Time-invariant

$$x_{k+1} = f(x_k, u_k, w),$$

Uncertainty over time #2

- ✓ **IID** – all w_k have the same probability distribution & they are independent,
- ✓ **Markovian** – the probability distribution of w_{k+1} is conditioned by w_k.

Control policy parametrisation

✓ Affine policy parametrization[10]

$$u_{k+j|k} = \mu_j(w_k, w_{k+1|k}, \ldots, w_{k+j-1|k})$$
$$= H_j \mathbf{w}_{k+j-1|k} + b_j$$

✓ Blocking affine policy parametrization

✓ Prestabilising feedback control as in stochastic tube MPC[11]

✓ Open-loop control actions[12]

[10] Kouvaritakis, Cannon and Munoz-Carpintero 2013; Oldewurtel *et al.* 2008; Korda, Gondhalekar, Cigler, Oldewurtel 2011.

[11] Cannon, Kouvaritakis, Ng 2009; Cannon *et al.* 2011.

[12] Kim and Braatz 2013; Bernardini and Bemporad, 2009, 2012.

Type of constraints

- ✓ Hard constraints

$$(x_k, u_k) \in Z$$

- ✓ Probabilistic constraints
  - ✓ Individual

$$\mathrm{P}[G_{(i)}x(t) \leq g_{(i)}] \geq 1 - \alpha_i \forall i$$

  - ✓ Joint

$$\mathrm{P}[G_{(i)}x(t) \leq g_{(i)}, \forall i] \geq 1 - \alpha$$

  - ✓ Expectation[13]
- ✓ Saturation of inputs[14]

[13] Hokayem, Cinquemani *et al.* 2012.
[14] Hokayem, Chatterjee and Lygeros 2009.

Uncertainty propagation

- ✓ Stochastic tube

$$x_k = z_k + e_k$$
$$u_k = Kx_k + c_k$$

- ✓ Scenario-based
- ✓ Gaussian mixture
- ✓ Other

Availability of feedback information

- ✓ State
  - ✓ Full state feedback
  - ✓ Output feedback
- ✓ Disturbance
  - ✓ Measured disturbance
  - ✓ Not measured

Further reading

1. A. Mesbah, “Stochastic Model Predictive Control: A Review,” IEEE Control Systems Magazine, 2016.
2. M. Kamgarpour, P. Hokayem, D. Chatterjee, M. Prandini, S. Garatti and A. Abate, “Final report on model predictive control for stochastic hybrid systems,” report of project “Moves”: http://www.movesproject.eu/deliverables/WP3/D3.2.pdf

III. Scenario trees

- ✓ The scenario tree structure
- ✓ Causality
- ✓ DP on a scenario tree

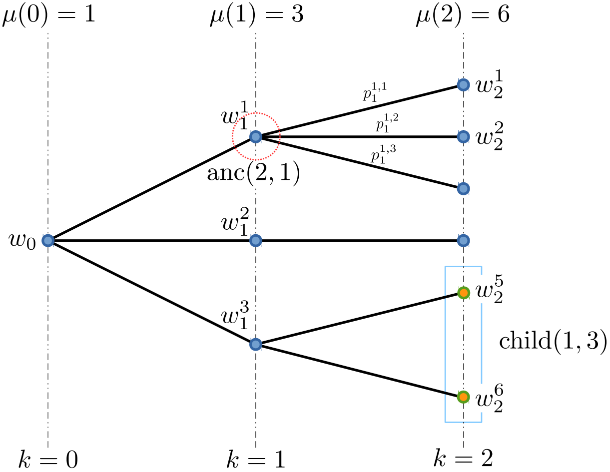
Motivation

1. Useful for numerical computations
2. Can be constructed from observations (data-driven)
3. They model non-iid processes
4. They provide a model for uncertainty propagation
5. Assumption: Ω_k are finite

Applications

- Micro-grids [Hans *et al.* '15]
- Drinking water networks [Sampathirao *et al.* '15]
- HVAC [Long *et al.* '13, Zhang *et al.* '13, Parisio *et al.* '13]
- Financial systems [Patrinos *et al.* '11, Bemporad *et al.*, '14]
- Chemical process [Lucia *et al.* '13]
- Distillation column [Garrido and Steinbach, '11]

Scenario tree structure



Definitions

- Let $N-1$ be the last stage of the tree

Definitions

- Let $N-1$ be the last stage of the tree
- At stage k we have $\mu(k)$ nodes and $\mu(0)=1$

Definitions

- Let $N-1$ be the last stage of the tree
- At stage k we have $\mu(k)$ nodes and $\mu(0)=1$
- The nodes at stage $N-1$ are called **leaf nodes**

Definitions

- Let $N-1$ be the last stage of the tree
- At stage k we have $\mu(k)$ nodes and $\mu(0)=1$
- The nodes at stage $N-1$ are called **leaf nodes**
- The node at stage 0 is the **root node**

Definitions

- Let $N-1$ be the last stage of the tree
- At stage k we have $\mu(k)$ nodes and $\mu(0)=1$
- The nodes at stage $N-1$ are called **leaf nodes**
- The node at stage 0 is the **root node**
- A **scenario** is an admissible path from the root node to a leaf node

Definitions

- Let $N-1$ be the last stage of the tree
- At stage k we have $\mu(k)$ nodes and $\mu(0)=1$
- The nodes at stage $N-1$ are called **leaf nodes**
- The node at stage 0 is the **root node**
- A **scenario** is an admissible path from the root node to a leaf node
- The tree counts $\mu(N-1)$ **scenarios**

Definitions

- Let $N-1$ be the last stage of the tree
- At stage k we have $\mu(k)$ nodes and $\mu(0)=1$
- The nodes at stage $N-1$ are called **leaf nodes**
- The node at stage 0 is the **root node**
- A **scenario** is an admissible path from the root node to a leaf node
- The tree counts $\mu(N-1)$ **scenarios**
- The **value** of ω at stage k, node i is denoted by ω_k^i

Definitions

- Let $N-1$ be the last stage of the tree
- At stage k we have $\mu(k)$ nodes and $\mu(0)=1$
- The nodes at stage $N-1$ are called **leaf nodes**
- The node at stage 0 is the **root node**
- A **scenario** is an admissible path from the root node to a leaf node
- The tree counts $\mu(N-1)$ **scenarios**
- The **value** of ω at stage k, node i is denoted by ω_k^i
- Each node $i \in \mathbb{N}_{[1,\mu(k)]}$ at stage $k \in \mathbb{N}_{[0,N-2]}$ has a set of **children** $\mathrm{child}(k,i) \subseteq \mathbb{N}_{[1,\mu(k+1)]}$

Definitions

- Let $N-1$ be the last stage of the tree
- At stage k we have $\mu(k)$ nodes and $\mu(0)=1$
- The nodes at stage $N-1$ are called **leaf nodes**
- The node at stage 0 is the **root node**
- A **scenario** is an admissible path from the root node to a leaf node
- The tree counts $\mu(N-1)$ **scenarios**
- The **value** of ω at stage k, node i is denoted by ω_k^i
- Each node $i \in \mathbb{N}_{[1,\mu(k)]}$ at stage $k \in \mathbb{N}_{[0,N-2]}$ has a set of **children** $\mathrm{child}(k,i) \subseteq \mathbb{N}_{[1,\mu(k+1)]}$
- Each node $i \in \mathbb{N}_{[1,\mu(k)]}$ at stage $k \in \mathbb{N}_{[0,N-2]}$ has a unique **ancestor** which is a node $j \in \mathbb{N}_{[1,\mu(k-1)]}$ at stage $k-1$

Definitions

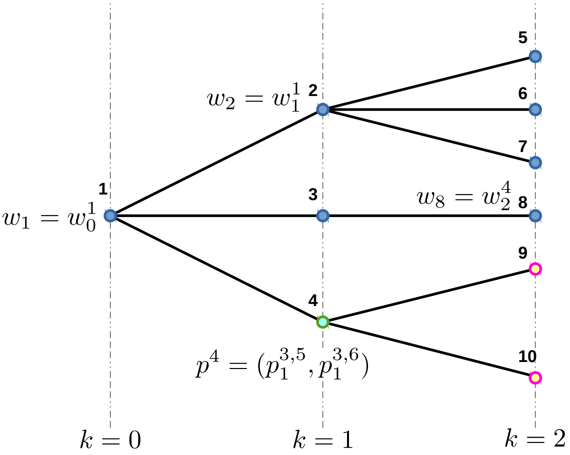
Conditional probability:

$$p_k^{i,j} = \mathrm{P}[\omega_{k+1} = \omega_{k+1}^j \mid \omega_k = \omega_k^i]$$

We have

$$\sum_{j \in \mathrm{child}(k,i)} p_k^{i,j} = 1, \text{ for all } k \in \mathbb{N}_{[0,N-2]},\ i \in \mathbb{N}_{[1,\mu(k)]}$$

* Alternative definitions



* Alternative definitions

- We enumerate all nodes of the tree by an index $\alpha \in \mathbb{N}_{[1,A]}$

* Alternative definitions

- We enumerate all nodes of the tree by an index $\alpha \in \mathbb{N}_{[1,A]}$
- The value of ω at node α is denoted by ω_α

* Alternative definitions

- We enumerate all nodes of the tree by an index $\alpha \in \mathbb{N}_{[1,A]}$
- The value of ω at node α is denoted by ω_α
- The set of nodes at stage k is denoted by Ω_k

* Alternative definitions

- We enumerate all nodes of the tree by an index $\alpha \in \mathbb{N}_{[1,A]}$
- The value of ω at node α is denoted by ω_α
- The set of nodes at stage k is denoted by Ω_k
- Each $\alpha \in \Omega_k$, $k \in \mathbb{N}_{[0,N-1]}$ defines the set of **children** nodes $\mathrm{child}(\alpha) \subseteq \Omega_{k+1}$

* Alternative definitions

- We enumerate all nodes of the tree by an index $\alpha \in \mathbb{N}_{[1,A]}$
- The value of ω at node α is denoted by ω_α
- The set of nodes at stage k is denoted by Ω_k
- Each $\alpha \in \Omega_k$, $k \in \mathbb{N}_{[0,N-1]}$ defines the set of **children** nodes $\operatorname{child}(\alpha) \subseteq \Omega_{k+1}$
- Each $\alpha \in \Omega_k$, $k \in \mathbb{N}_{[1,N]}$ defines a unique **ancestor** $\operatorname{anc}(\alpha) \in \Omega_{k-1}$

* Alternative definitions

- We enumerate all nodes of the tree by an index $\alpha \in \mathbb{N}_{[1,A]}$
- The value of ω at node α is denoted by ω_α
- The set of nodes at stage k is denoted by Ω_k
- Each $\alpha \in \Omega_k$, $k \in \mathbb{N}_{[0,N-1]}$ defines the set of **children** nodes $\mathrm{child}(\alpha) \subseteq \Omega_{k+1}$
- Each $\alpha \in \Omega_k$, $k \in \mathbb{N}_{[1,N]}$ defines a unique **ancestor** $\mathrm{anc}(\alpha) \in \Omega_{k-1}$
- We define the **probability vectors** p^α for $\alpha \in \Omega_k$ so that

* Alternative definitions

- We enumerate all nodes of the tree by an index $\alpha \in \mathbb{N}_{[1,A]}$
- The value of ω at node α is denoted by ω_α
- The set of nodes at stage k is denoted by Ω_k
- Each $\alpha \in \Omega_k$, $k \in \mathbb{N}_{[0,N-1]}$ defines the set of **children** nodes $\mathrm{child}(\alpha) \subseteq \Omega_{k+1}$
- Each $\alpha \in \Omega_k$, $k \in \mathbb{N}_{[1,N]}$ defines a unique **ancestor** $\mathrm{anc}(\alpha) \in \Omega_{k-1}$
- We define the **probability vectors** p^α for $\alpha \in \Omega_k$ so that
  - ✓ $p^\alpha \in \mathbb{R}^{|\,\mathrm{child}(\alpha)|}$

* Alternative definitions

- We enumerate all nodes of the tree by an index $\alpha \in \mathbb{N}_{[1,A]}$
- The value of ω at node α is denoted by ω_α
- The set of nodes at stage k is denoted by Ω_k
- Each $\alpha \in \Omega_k$, $k \in \mathbb{N}_{[0,N-1]}$ defines the set of **children** nodes $\operatorname{child}(\alpha) \subseteq \Omega_{k+1}$
- Each $\alpha \in \Omega_k$, $k \in \mathbb{N}_{[1,N]}$ defines a unique **ancestor** $\operatorname{anc}(\alpha) \in \Omega_{k-1}$
- We define the **probability vectors** p^α for $\alpha \in \Omega_k$ so that
  - ✓ $p^\alpha \in \mathbb{R}^{|\operatorname{child}(\alpha)|}$
  - ✓ $\sum_{\beta \in \operatorname{child}(\alpha)} p^\alpha(\beta) = 1$

* Alternative definitions

- We enumerate all nodes of the tree by an index $\alpha \in \mathbb{N}_{[1,A]}$
- The value of ω at node α is denoted by ω_α
- The set of nodes at stage k is denoted by Ω_k
- Each $\alpha \in \Omega_k$, $k \in \mathbb{N}_{[0,N-1]}$ defines the set of **children** nodes $\mathrm{child}(\alpha) \subseteq \Omega_{k+1}$
- Each $\alpha \in \Omega_k$, $k \in \mathbb{N}_{[1,N]}$ defines a unique **ancestor** $\mathrm{anc}(\alpha) \in \Omega_{k-1}$
- We define the **probability vectors** p^α for $\alpha \in \Omega_k$ so that
  - ✓ $p^\alpha \in \mathbb{R}^{|\,\mathrm{child}(\alpha)|}$
  - ✓ $\sum_{\beta \in \mathrm{child}(\alpha)} p^\alpha(\beta) = 1$
  - ✓ $p^\alpha(\beta) = \mathrm{P}[\omega_{k+1} = \beta \mid \omega_k = \alpha]$

A few properties

For every $k \in \mathbb{N}_{[0,N-2]}$ we have

$$\Omega_{k+1} = \bigcup_{i \in \mathbb{N}_{[1,\mu(k)]}} \mathrm{child}(k,i)$$

and, for fixed k, sets $\{\mathrm{child}(k,i)\}_i$ are disjoint.

A few properties

For $\alpha \in \Omega_k$ the family of sets $\{\text{child}(\alpha)\}_{\alpha \in \Omega_k}$ defines a **partition** in Ω_{k+1}, that is

$$\Omega_{k+1} = \bigcup_{\alpha \in \Omega_k} \text{child}(\alpha),$$

and

$$\alpha_1 \neq \alpha_2 \Rightarrow \text{child}(\alpha_1) \cap \text{child}(\alpha_2) = \varnothing.$$

A few properties

The probability of a scenario[15], identified by a leaf node $\alpha \in \Omega_N$, is defined as

$$\pi_\alpha = \mathrm{P}[\omega_N = \alpha \mid \omega_1 = \omega_0^1],$$

and is given by

$$\pi_\alpha = \prod_{i=1}^{N-1} p^{\mathrm{anc}^i(\alpha)},$$

where $\mathrm{anc}^1(\alpha) = \mathrm{anc}(\alpha)$ and

$$\mathrm{anc}^{k+1}(\alpha) = \mathrm{anc}(\mathrm{anc}^k(\alpha)).$$

[15] Detail: This probability is defined in the product space $(\prod_k \Omega_k, \otimes_k \mathcal{F}_k)$.

Notes

- We allow $w_k^i = w_k^j$ for $i \neq j$ – it is not the value that identifies the node[16]!

[16] it is, instead, its history

Notes

- We allow $w_k^i = w_k^j$ for $i \neq j$ – it is not the value that identifies the node[16]!
- We have as many scenarios as leaf nodes, that is $\mu(N-1)$,

[16] it is, instead, its history

Notes

- We allow $w_k^i = w_k^j$ for $i \neq j$ – it is not the value that identifies the node[16]!
- We have as many scenarios as leaf nodes, that is $\mu(N-1)$,
- Scenarios are sequences $(w_k^{i_k})_k$ where $i_{k-1} = \mathrm{anc}(k, i_k)$,

[16] it is, instead, its history

Notes

- We allow $w_k^i = w_k^j$ for $i \neq j$ – it is not the value that identifies the node[16]!
- We have as many scenarios as leaf nodes, that is $\mu(N-1)$,
- Scenarios are sequences $(w_k^{i_k})_k$ where $i_{k-1} = \operatorname{anc}(k, i_k)$,
- Every node w_k^i of the tree can be identified by their stage k an a scenario running through it or

[16] it is, instead, its history

Notes

- We allow $w_k^i = w_k^j$ for $i \neq j$ – it is not the value that identifies the node[16]!
- We have as many scenarios as leaf nodes, that is $\mu(N-1)$,
- Scenarios are sequences $(w_k^{i_k})_k$ where $i_{k-1} = \mathrm{anc}(k, i_k)$,
- Every node w_k^i of the tree can be identified by their stage k an a scenario running through it or
- by a sequence $(w_j^{i_j})_j$ leading to w_k^i,

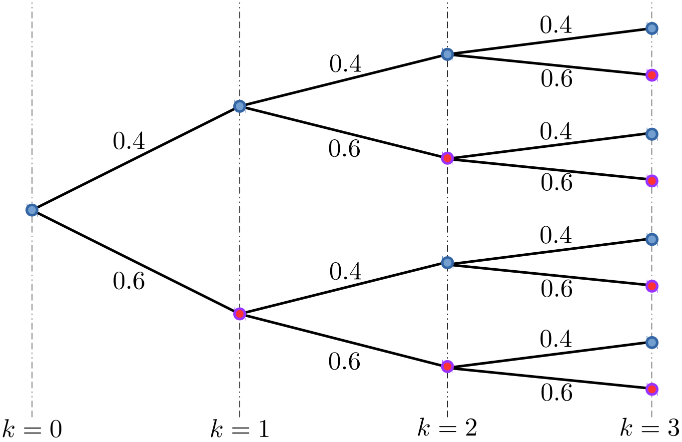
[16] it is, instead, its history

Notes

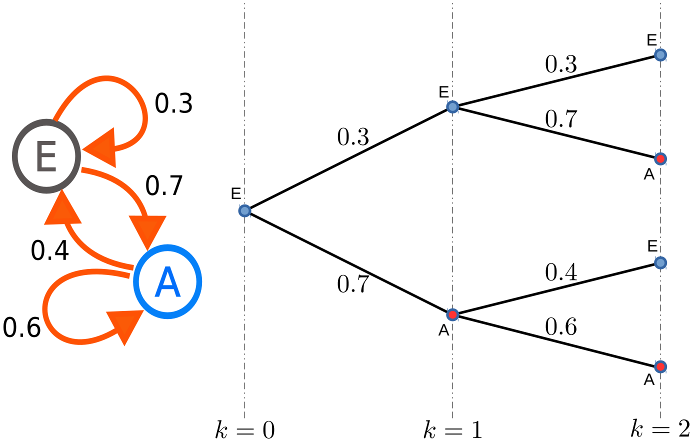
- We allow $w_k^i = w_k^j$ for $i \neq j$ – it is not the value that identifies the node[16]!
- We have as many scenarios as leaf nodes, that is $\mu(N-1)$,
- Scenarios are sequences $(w_k^{i_k})_k$ where $i_{k-1} = \mathrm{anc}(k, i_k)$,
- Every node w_k^i of the tree can be identified by their stage k an a scenario running through it or
- by a sequence $(w_j^{i_j})_j$ leading to w_k^i,
- The probability of a scenario $(w_k^{i_k})_k$ (on the space of scenarios[17]), which is identified by a leaf node $i \in \mathbb{N}_{[1,\mu(N-1)]}$ is given by the product of conditional probabilities that connect its nodes.

[16] it is, instead, its history
[17] We are going to give a formal definition of this space later.

IID processes



Markov chains



Filtration

A *filtration* is an increasing sequence of σ-algebras which, here, we may construct as follows

Filtration

A *filtration* is an increasing sequence of σ-algebras which, here, we may construct as follows

- $\mathcal{F}_{N-1} = 2^{\Omega_{N-1}}$: take all subsets of Ω_{N-1}

Filtration

A *filtration* is an increasing sequence of σ-algebras which, here, we may construct as follows

- $\mathcal{F}_{N-1} = 2^{\Omega_{N-1}}$: take all subsets of Ω_{N-1}
- Let $\mathcal{F}_{N-2} \subseteq \mathcal{F}_{N-1}$ be the smallest σ-algebra containing all $\mathrm{child}(N-2, i)$ for all i.

Filtration

A *filtration* is an increasing sequence of σ-algebras which, here, we may construct as follows

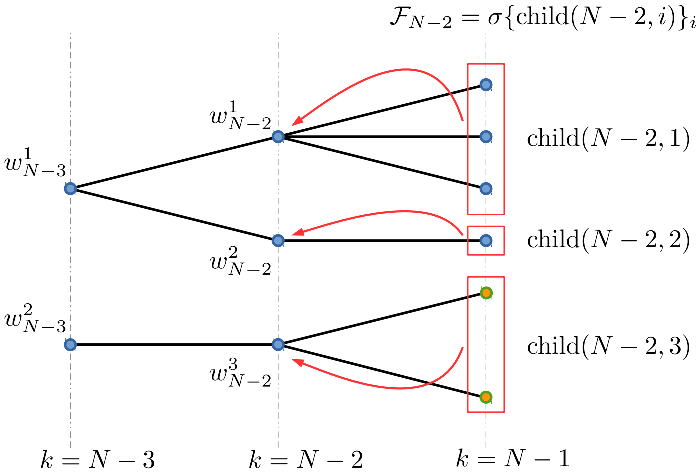
- $\mathcal{F}_{N-1} = 2^{\Omega_{N-1}}$: take all subsets of Ω_{N-1}
- Let $\mathcal{F}_{N-2} \subseteq \mathcal{F}_{N-1}$ be the smallest σ-algebra containing all $\mathrm{child}(N-2, i)$ for all i.
- Recursively, construct $\mathcal{F}_{N-j} \subseteq \mathcal{F}_{N-j+1}$

Filtration

A *filtration* is an increasing sequence of σ-algebras which, here, we may construct as follows

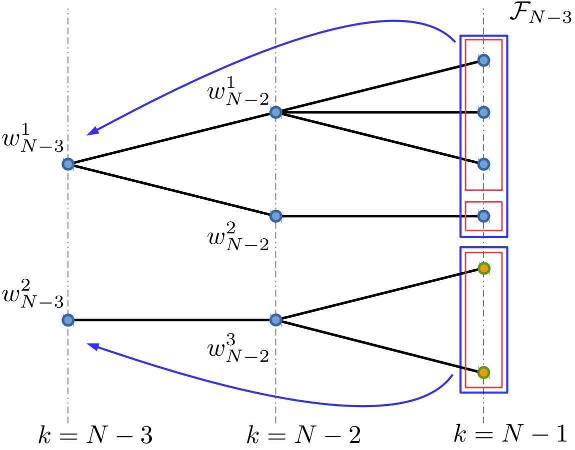
- $\mathcal{F}_{N-1} = 2^{\Omega_{N-1}}$: take all subsets of Ω_{N-1}
- Let $\mathcal{F}_{N-2} \subseteq \mathcal{F}_{N-1}$ be the smallest σ-algebra containing all $\mathrm{child}(N-2, i)$ for all i.
- Recursively, construct $\mathcal{F}_{N-j} \subseteq \mathcal{F}_{N-j+1}$
- Eventually, $\mathcal{F}_0 = \{\varnothing, \Omega_{N-1}\}$ and recall that w_0 is deterministic.

Filtration

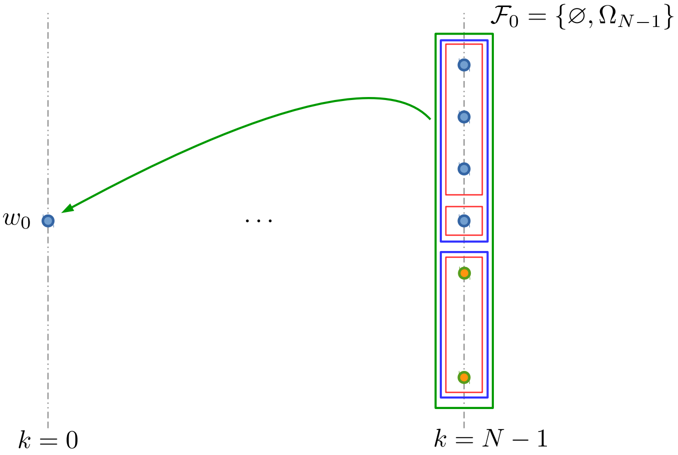


$\mathcal{F}_{N-2} = \sigma\{\text{child}(N-2,i)\}_i$
w^1_{N-2}
w^1_{N-3}
child$(N-2,1)$
child$(N-2,2)$
w^2_{N-2}
w^2_{N-3}
child$(N-2,3)$
w^3_{N-2}
$k = N-3$
$k = N-2$
$k = N-1$

Filtration



Filtration



Filtration

Some remarks:

- Every node $i \in \mathbb{N}_{[1,\mu(k)]}$ at a stage k corresponds to an event in $\mathcal{F}_k$.
- The cardinality of $\mathcal{F}_k$ is $2^{|\Omega_k|}$ (why?)
- The product space $(\Omega, \mathcal{F}, \mathrm{P}) = \prod_k (\Omega_k, \mathcal{F}_k, \mathrm{P}_k)$, equipped with the filtration $\{\mathcal{F}_k\}_k$ becomes a *filtered* probability space.

State sequence

The state dynamics is given by $x_{k+1} = f(x_k, u_k, w_k)$, so, starting at $k = 0$ from a state x_0 and knowing w_0 the predicted state sequence is[18]

$$\begin{aligned} x_1 &= f(x_0, \mu_0(x_0, w_0), w_0) \\ &= f(x_0, u_0, w_0) \end{aligned}$$

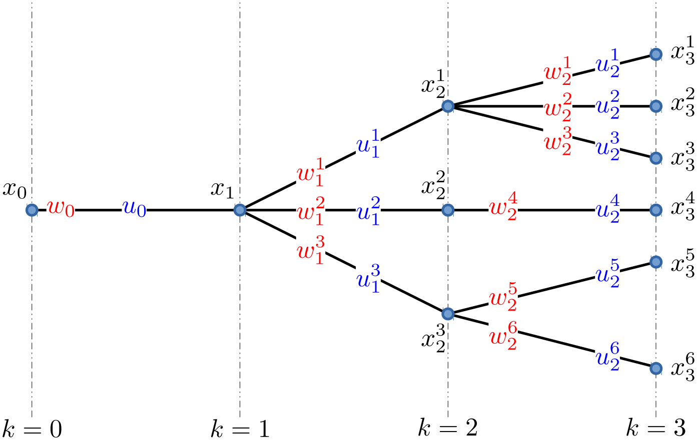
and for all $j \in \mathbb{N}_{[0,N-2]}$

$$x_{j+1} = f(x_j, \mu_j(x_0, \mathbf{w}_j, x_j), w_j)$$

where $\mathbf{w}_j = (w_0, w_1, \ldots, w_j)$.

[18]Warning: abuse of notation! Here x_0 is an observation whereas x_k is an estimate of a future state (which is a random variable). Thus, x_2 is not the state measurement at time $k = 2$; A more proper notation would be $x_{k+2|k}$.

State sequence



Decision making across the tree nodes

- At $k = 0$ we know x_0 and w_0, so we decide u_0
- At $k = 1$ the state will be $x_1 = f(x_0, u_0, w_0)$ and we observe $w_1 \in \Omega_1$
- For each $i \in \mathbb{N}_{[1,\mu(1)]}$ we decide a u_1^i and apply it to the system
- The next state will be $x_2^i = f(x_1, u_1^i, w_1^i)$, $i \in \mathbb{N}_{[1,\mu(1)]}$
- At stage k we decide the input u_k according to the information available so far, so
$$u_k = \mu_k(x_0, \mathbf{w}_j, x_j)$$
- or, equivalenty, we choose u_k^i at each node of the tree at stage k.

State sequence

The state sequence becomes

$$x_{k+1}^{i} = f(x_k^j, u_k^i, w_k^i),$$

for all $k \in \mathbb{N}_{[0,N-2]}$ and $i \in \mathbb{N}_{[1,\mu(k+1)]}$, where $j = \mathrm{anc}(k, i)$.

Adaptation to $\mathcal{F}_k$

By construction $\{u_k\}_k$ is an $\mathcal{F}_k$-adapted random process (why?) and

$$\mathbb{E}\left[u_k \mid \mathcal{F}_k\right] = u_k.$$

Sequence $\{x_k\}_k$ is a *predictable process* wrt $\{\mathcal{F}_k\}_k$, i.e., x_{k+1} is $\mathcal{F}_k$-adapted, that is

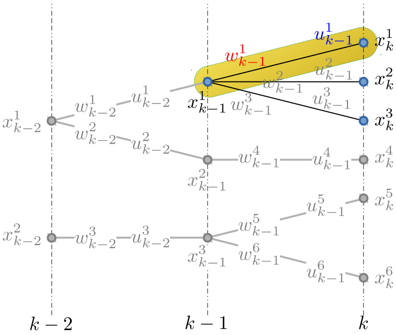
$$\mathbb{E}\left[x_{k+1} \mid \mathcal{F}_k\right] = x_{k+1}.$$

Indeed, recall that

$$x_{k+1} = f(x_k, \mu_k(x_0, \mathbf{w}_k, x_k), w_k).$$

DP on scenario trees

Assuming x_k and w_k are observable at time k; we decide u_k^i at every edge of the tree, i.e., one input for every node of the w-tree.



DP on scenario trees

Exercise.
Solve the DP problem assuming x_k and w_k are observable at time k.
Assume linear dynamics of the form[19]

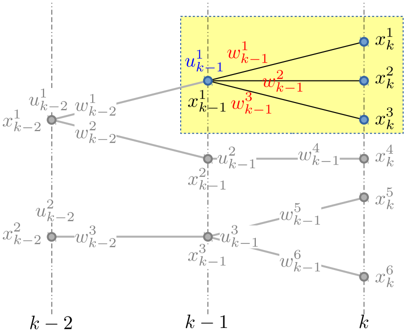
$$x_{k+1}^i = A(w_k^i)x_k^j + B(w_k^i)u_k^i + w_k^i,$$

for all $k \in \mathbb{N}_{[0,N-2]}$ and $i \in \mathbb{N}_{[1,\mu(k+1)]}$, where $j = \operatorname{anc}(k,i)$.

[19] For convenience we may denote $A_k^i = A(w_k^i)$ and $B_k^i = B(w_k^i)$. Since w_k is observable at time k, we also observe A_k and B_k.

DP on scenario trees

Assuming only x_k is observable at time k; we decide u_k^i at every edge of the x-tree, i.e., as in the following figure:



Scenario-based SMPC

We'll see how to design a MS-stabilising SMPC using scenario tree representations of uncertainty.

Definition of $\mathcal{L}V$

Consider the linear autonomous system

$$x_{k+1} = f(x_k, w_k).$$

Assume w_k is not observable at time k and let V be a function which maps a $x \in \mathbb{R}^n$ to a $\mathbb{R}_+$-valued random variable[20] for which we define the random variable

$$\mathcal{L}V(x_k) := \mathbb{E}[V(x_{k+1}) - V(x_k) \mid \mathcal{F}_k].$$

[20] We'll avoid the details to keep the notation reasonably simple.

A useful property of $\mathcal{L}V$

A useful property of $\mathcal{L}V$ (the discrete version of *Dynkin's formula*.): For $0 \leq k_1 \leq k_2$:

$$\mathbb{E}\left[V(x_{k_2}) - V(x_{k_1}) \mid \mathcal{F}_{k_1}\right] = \mathbb{E}\left[\sum_{j=k_1}^{k_2} \mathcal{L}V(x_k) \,\middle|\, \mathcal{F}_{k_1}\right]$$

Note: To prove this we only need to use the tower property:

$$\mathcal{H}_1 \subseteq \mathcal{H}_2 \Rightarrow \mathbb{E}\left[\mathbb{E}\left[X \mid \mathcal{H}_2\right] \mid \mathcal{H}_1\right] = \mathbb{E}[X \mid \mathcal{H}_1],$$

where X is a r.v. on $(\Omega, \mathcal{F}, \mathrm{P})$ and $\mathcal{H}_i$ are sub-σ-algebras of $\mathcal{F}$.

A useful property of $\mathcal{L}V$

Proof. Take $0 \leq k_1 \leq k_2$ and

$$\begin{aligned}
V(x_{k_2}) - V(x_{k_1}) &= \sum_{j=k_1}^{k_2} V(x_{j+1}) - V(x_j) \\
\Rightarrow \mathbb{E}\left[V(x_{k_2}) - V(x_{k_1}) \mid \mathcal{F}_{k_1}\right] &= \mathbb{E}\left[\sum_{j=k_1}^{k_2} V(x_{j+1}) - V(x_j) \mid \mathcal{F}_{k_1}\right] \\
&= \mathbb{E}\left[\sum_{j=k_1}^{k_2} \mathbb{E}[V(x_{j+1}) - V(x_j) \mid \mathcal{F}_j] \mid \mathcal{F}_{k_1}\right] \\
&= \mathbb{E}\left[\sum_{j=k_1}^{k_2} \mathcal{L}V(x_k) \,\middle|\, \mathcal{F}_{k_1}\right].
\end{aligned}$$

Lyapunov theorem (MSS)

Assume that for all $x \in \mathbb{R}^n$ we have

$$\mathcal{L}V(x) \leq -\gamma \|x\|^2,$$

for some $\gamma > 0$, then $x_{k+1} = f(x_k, w_k)$ is MSS[21].

Proof (**Exercise**). Use the property of $\mathcal{L}V$ with $k_1 = 0$ and $k_2 = k$.

[21] We further have that $\{\mathbb{E}\left[\|x_k\|^2 \mid \mathcal{F}_0\right]\}_k$ is an ℓ^2 sequence.

Lyapunov theorem (MSS)

If for all $x \in \mathbb{R}^n$ we have

$$\mathcal{L}V(x) \leq -\alpha(\|x\|^2),$$

for some convex $\mathcal{K}$-class function α, then $x_{k+1} = f(x_k, w_k)$ is MSS.

Proof. The proof is an **exercise**. Show with an example that the convexity requirement cannot be omitted. Show also that the we can alternatively use the condition $\mathcal{L}V(x) \leq -x'Lx$, where $L = L' \succ 0$.

Lyapunov theorem (MSES)

If for all $x \in \mathbb{R}^n$ we have

$$\mathcal{L}V(x) \leq -\gamma \|x\|^2,$$
$$\alpha \|x\|^2 \leq V(x) \leq \beta \|x\|^2,$$

for some $\alpha, \beta, \gamma > 0$, then $x_{k+1} = f(x_k, w_k)$ is MSES.

Proof. *Easy exercise.*

MS-stabilising stochastic MPC

Assume that w_k over $(\Omega, \mathcal{F}, \mathrm{P})$ is IID. We formulate the following SMPC problem (unconstrained case) [Bernardini and Bemporad, 2012]:

$$V^\star(x_k) = \min_{\pi=\{\mu_i\}_{i=0}^{N-1}} \mathbb{E} V_N(x_k, \pi)$$

$$x_{k|k} = x_k,$$

$$x_{k+i+1|k} = A(w_{k+i|k})x_{k+i|k} + B(w_{k+i|k})\mu_i(x_{k+i|k}), \forall i \in \mathbb{N}_{[0,N-1]}$$

$$\mathcal{L}V(x_{k|k}) \leq -x_{k|k}' L x_{k|k},$$

for some $L = L' \succ 0$, where $V(x) = x'Px$ and

$$\mathcal{L}V(x_{k|k}) = \mathbb{E}[V(x_{k+1|k}) - V(x_k) \mid x_{k|k}]$$

MS-stabilising stochastic MPC

Because of the constraint

$$\mathcal{L}V(x_{k|k}) \leq -x'_{k|k} L x_{k|k},$$

the control law

$$u_k = \mu_0(x_k),$$

leads to a **MSES** closed-loop system (this SMPC problem is **recursively feasible**).

MS-stabilising stochastic MPC

NOTE:
We can choose **any** cost function $V_N(x_k, \pi)$!

What about those trees?

Hold on... we'll get there.

Inexact knowledge of $\mathrm{P}(\mathrm{d}w_k)$

We previously assumed that:

- w_k is an IID process on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$

Inexact knowledge of $\mathrm{P}(\mathrm{d}w_k)$

We previously assumed that:

- w_k is an IID process on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$
- The probability measure P is exactly known (and time invariant)

Inexact knowledge of $\mathrm{P}(\mathrm{d}w_k)$

We previously assumed that:

- w_k is an IID process on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$
- The probability measure P is exactly known (and time invariant)
- In case Ω is finite and $\mathcal{F} = 2^{\Omega}$ if suffices to know $p = (\mathrm{P}(w^i))_{i=1}^{s}$ for all $w^i \in \Omega$,

Inexact knowledge of $\mathrm{P}(\mathrm{d}w_k)$

We previously assumed that:

- w_k is an IID process on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$
- The probability measure P is exactly known (and time invariant)
- In case Ω is finite and $\mathcal{F} = 2^{\Omega}$ if suffices to know $p = (\mathrm{P}(w^i))_{i=1}^{s}$ for all $w^i \in \Omega$,
- When Ω is finite we can identify probability measures with vectors

Inexact knowledge of $\mathrm{P}(\mathrm{d}w_k)$

We previously assumed that:

- w_k is an IID process on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$
- The probability measure P is exactly known (and time invariant)
- In case Ω is finite and $\mathcal{F} = 2^\Omega$ if suffices to know $p = (\mathrm{P}(w^i))_{i=1}^s$ for all $w^i \in \Omega$,
- When Ω is finite we can identify probability measures with vectors
- Let
$$\mathcal{D} = \left\{ p \in \mathbb{R}^s : p \geq 0, 1_s' p = 1 \right\},$$
thus any probability measure p is an element of $\mathcal{D}$.

Inexact knowledge of $\mathrm{P}(\mathrm{d}w_k)$

We previously assumed that:

- w_k is an IID process on a probability space $(\Omega, \mathcal{F}, \mathrm{P})$
- The probability measure P is exactly known (and time invariant)
- In case Ω is finite and $\mathcal{F} = 2^\Omega$ if suffices to know $p = (\mathrm{P}(w^i))_{i=1}^s$ for all $w^i \in \Omega$,
- When Ω is finite we can identify probability measures with vectors
- Let
$$\mathcal{D} = \left\{p \in \mathbb{R}^s : p \geq 0, 1_s' p = 1\right\},$$
thus any probability measure p is an element of $\mathcal{D}$.
- A set $\mathcal{P} \subseteq \mathcal{D}$ is a set of probability measures.

Inexact knowledge of $\mathrm{P}(\mathrm{d}w_k)$

We will now drop the IID assumption and assume that w_k is a random variable over $(\Omega, \mathcal{F}, \mathrm{P}_k)$ where $\Omega = \{w^i\}_{i=1}^s$ is finite.

We define for $\mathrm{P} \in \mathcal{D}$

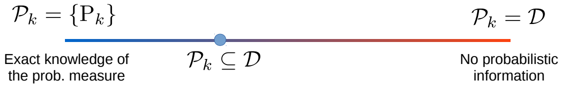
$$\begin{aligned}\mathcal{L}V(x,\mu,\mathrm{P}) &:= \mathbb{E}_{\mathrm{P}}[V(x_{k+1|k}) - V(x_{k|k}) \mid x_{k|k} = x] \\ &= \int_{\Omega} V(x_{k+1|k}) - V(x_k)\,\mathrm{P}(\mathrm{d}w \mid x_{k|k} = x) \\ &= \sum_{i=1}^{s} \underbrace{\mathrm{P}[w_k = w^i]}_{p_i} V(A(w^i)x + B(w^i)\mu(x)) - V(x)\end{aligned}$$

Inexact knowledge of $\mathrm{P}(\mathrm{d}w_k)$

In that case, the MS-stabilising constraint becomes:

$$\mathcal{L}V(x, \mu_0, \mathrm{P}) \leq -x' L x, \forall\, \mathrm{P} \in \mathcal{D},$$

for some μ_0.



This is reminiscent of the *worst-case distribution* approach[22].

[22] See A. Shapiro, "Worst-case distribution analysis of stochastic programs," Math. Prog. 107(1), pp. 91-96, 2005.

MS-stabilising scenario-based SMPC

MS-stabilising SMPC formulation:

$$
\begin{aligned}
&V^\star(x_k) = \min_{\pi=\{\mu_i\}_{i=0}^{N-1}} \mathbb{E}V_N(x_k,\pi)\\
&x_{k|k} = x_k,\\
&x^l_{k+i+1|k} = A(w^l_{k+i|k})x^\iota_{k+i|k} + B(w^l_{k+i|k})u^l_{k+i|k},\\
&\quad \forall i \in \mathbb{N}_{[0,N-1]}, \iota = \operatorname{anc}(i,l), l \in \mathbb{N}_{[1,\mu(i+1)]}\\
&\mathcal{L}V(x_{k+1|k}, u_{k|k}, \mathrm{P}) \leq -x_k' L x_k, \forall\, \mathrm{P} \in \mathcal{P}
\end{aligned}
$$

If $\mathcal{P} \subseteq \mathcal{D}$ is a polytope with vertices $\{\mathrm{P}_\kappa\}_{\kappa=1}^K$, we need to impose the above stabilising constraint only for those vertices, i.e.,

$$\mathcal{L}V(x_1, u_0, \mathrm{P}_\kappa) \leq -x_0' L x_0, \forall \kappa \in \mathbb{N}_{[1,K]}$$

The constrained case

- The same approach applies to state/input-constrained system so long as the formulation is **recursively feasible**
- For details see: D. Bernardini and A. Bemporad, "Stabilizing Model Predictive Control of Stochastic Constrained Linear Systems," IEEE TAC 57(6), pp. 1468–1480, 2012.

IV. Affine disturbance feedback

Problem statement – dynamics

We'll be studying a very simple case. The system dynamics is given by

$$x_{k+1} = Ax_k + Bu_k + Gw_k,$$

where w_k is an iid process with $w_k \sim \mathcal{N}(0, I)$. The disturbance w_k is not observable at time k.

Problem statement – constraints

The system is subject to hard input constraints

$$H_u u_k \leq K_u,$$

and *probabilistic* state constraints (stage-wise):

$$\mathrm{P}[H_x^l \cdot x_k \leq K_x^l] \geq 1 - \alpha^l,\ l \in \mathbb{N}_{[1,s]}$$

where H_x^l are vectors and K_x^l are scalars and $\alpha^l \in [0, 1]$.

Problem statement – policies

Along the horizon the inputs $u_{k+j|k}$ are determined by causal control laws of the form

$$
\begin{aligned}
u_{k+j|k} &= \psi_{k+j|k}(x_{k|k}, x_{k+1|k}, \ldots, x_{k+j|k}) \\
&= \mu_{k+j|k}(x_{k|k}, w_{k|k}, \ldots, w_{k+j-1|k}),
\end{aligned}
$$

so, along the horizon, we need to determine

$$
\boldsymbol{\mu}_k = \begin{pmatrix} u_{k|k}, & \mu_{k+1|k}, & \cdots & \mu_{k+N-1|k} \end{pmatrix}
$$

which is a sequence of functions. Functions are **infinite dimensional** objects, so the optimisation problem becomes **intractable**.

The stochastic MPC problem

We formulate the following optimisation problem:

$$V_N^{\star}(x_k) = \min_{\boldsymbol{\mu}_k} \mathbb{E} V_N(x_k, \boldsymbol{\mu}_k),$$

subject to

$$\begin{aligned}
&x_{k+j+1|k} = Ax_{k+j|k} + Bu_{k+j|k} + Gw_{k+j|k}, \forall j \in \mathbb{N}_{[0,N-1]}\\
&u_{k+j|k} = \mu_{k+j|k}(x_{k|k}, w_{k|k}, \ldots, w_{k+j-1|k}), \forall j \in \mathbb{N}_{[0,N-1]}\\
&H_u u_{k+j|k} \leq K_u, \forall j \in \mathbb{N}_{[0,N-1]}\\
&\mathrm{P}[h_x^l \cdot x_{k+j|k} \leq k_x^l] \geq 1 - \alpha^l,\ \forall l \in \mathbb{N}_{[1,s]}, \forall j \in \mathbb{N}_{[1,N]}\\
&w_{k+j|k} \sim \mathcal{N}(0, I), \forall j \in \mathbb{N}_{[0,N-1]}\\
&x_{k|k} = x_k
\end{aligned}$$

Affine disturbance feedback

For convenience define

$$\mathbf{w}_k = (w_{k|k}, w_{k+1|k}, \ldots, w_{k+N-1|k}).$$

We restrict ourselves to causal policies whose functions have the simple form

$$\mu_{k+j|k}(\mathbf{w}_k) = m_j + \sum_{i=0}^{j-1} M_{j,i} w_{k+j|k}$$

We then need to determine $M_{j,i}$ and m_j, so the opt. problem becomes finite dimensional.

Affine disturbance feedback

The affine disturbance feedback policy can be concisely written as

$$\boldsymbol{\mu}_k(\mathbf{w}_k) = \mathbf{M}\mathbf{w}_k + \mathbf{m},$$

where

$$\mathbf{M} = \begin{bmatrix} 0 & \dots & \dots & 0 & 0 \\ M_{1,0} & 0 & \dots & 0 & 0 \\ \vdots & & & & \\ M_{N-1,0} & M_{N-1,1} & \dots & M_{N-1,N-2} & 0 \end{bmatrix}, \; \mathbf{m} = \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{N-1} \end{bmatrix}$$

Finite dimensional problem

Define also

$$\mathbf{x}_k = (x_k, x_{k+1|k,\ldots,x_{k+N|k}}),$$
$$\mathbf{u}_k = (u_{k|k}, u_{k+1|k,\ldots,u_{k+N-1|k}}).$$

The state evolution of the system, $\mathbf{x}_k$ given a sequence of inputs $\mathbf{u}_k$ and a sequence of disturbances $\mathbf{w}_k$ is given by[23]

$$\mathbf{x}_k = \mathbf{A}x_k + \mathbf{B}\mathbf{u}_k + \mathbf{G}\mathbf{w}_k.$$

[23]Exercise: Determine $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{G}$.

Probabilistic constraints

... and substituting $\mathbf{u}_k = \boldsymbol{\mu}_k(\mathbf{w}_k)$:

$$\begin{aligned}\mathbf{x}_k &= \mathbf{A}x_k + \mathbf{B}\mathbf{u}_k + \mathbf{G}\mathbf{w}_k \\ &= \mathbf{A}x_k + \mathbf{B}\mathbf{M}\mathbf{w}_k + \mathbf{G}\mathbf{w}_k + \mathbf{B}\mathbf{m} \\ &= \mathbf{A}x_k + (\mathbf{B}\mathbf{M} + \mathbf{G})\mathbf{w}_k + \mathbf{B}\mathbf{m}\end{aligned}$$

Then, the probabilistic constraints are written as

$$\begin{aligned}&\mathrm{P}[\mathbf{H}_x^l(\mathbf{A}x_k + (\mathbf{B}\mathbf{M} + \mathbf{G})\mathbf{w}_k + \mathbf{B}\mathbf{m}) \leq \mathbf{K}_x^l] \geq 1 - \alpha^l \\ \Leftrightarrow\ &\mathrm{P}[\mathbf{H}_x^l(\mathbf{B}\mathbf{M} + \mathbf{G})\mathbf{w}_k + \mathbf{H}_x^l(\mathbf{A}x_k + \mathbf{B}\mathbf{m}) \leq \mathbf{K}_k^l] \geq 1 - \alpha^l\end{aligned}$$

Distribution function

Let X be a random value over a probability space $(\mathbb{R}, \mathcal{B}_{\mathbb{R}}, \mathrm{P})$. The **distribution** of X is a measure $\mu : \mathcal{B}_{\mathbb{R}} \to [0,1]$ with

$$\begin{aligned} \mu(B) &= \mathrm{P}(X \in B) \\ &= \mathrm{P}(\{\omega \mid X(\omega) \in B\}). \end{aligned}$$

The distribution of X is identified by the following function (why?) known as **distribution function** of X, $c : \mathbb{R} \to [0,1]$

$$\begin{aligned} c(x) &= \mu((-\infty, x]) \\ &= \mathrm{P}(X < x) \\ &= \mathrm{P}(\{\omega \in \Omega \mid X(\omega) < x\}). \end{aligned}$$

Quantile function

We define the **quantile function** of X, $Q : [0, 1] \to \mathbb{R}$ as

$$\begin{aligned} Q(p) &= \inf\{x \in \mathbb{R} : p \leq c(x)\} \\ &= \inf\{x \in \mathbb{R} : \mathrm{P}[X \leq x] \geq p\} \end{aligned}$$

This is a type of inverse of the distribution function.

Normal distribution

When X is normally distributed, $X \sim \mathcal{N}(\mu, \sigma)$, $Q(p)$ is given by an explicit formula, that is

$$Q(p) = \mu + 2\sqrt{\sigma}\,\mathrm{erf}^{-1}(2p - 1).$$

As a result

$$\begin{aligned} &\mathrm{P}[X \leq x] \geq 1 - \alpha \\ \Leftrightarrow\ & x \geq Q(1 - \alpha). \end{aligned}$$

Multivariate Normal distribution

When X is normally distributed, $X \sim \mathcal{N}(\bar{x}, \Sigma)$ and $Z := AX + b$, then

$$Z \sim \mathcal{N}(A\bar{x} + b, A\Sigma A').$$

A little observation

Let $y_1 \sim \mathcal{N}(0_n, I_n)$ and $y_2 \sim \mathcal{N}(0_n, I_n)$ and y_1, y_2 are independent random variables. Then

$$y := (y_1, y_2) \sim \mathcal{N}(0_{2n}, I_{2n}).$$

In our case

$$\mathbf{w}_k \sim \mathcal{N}(0, I).$$

Probabilistic constraints revisited

The probabilistic constraints are written as

$$\mathrm{P}[\underbrace{\mathbf{H}_x^l(\mathbf{BM}+\mathbf{G})}_{\text{row-vector}}\underbrace{\mathbf{w}_k}_{\mathcal{N}(0,I)} \leq g^l] \geq 1-\alpha^l$$

where $g^l = \mathbf{K}_k^l - \mathbf{H}_x^l(\mathbf{A}x_k + \mathbf{Bm})$. This becomes

$$Q(1-\alpha^l)\|\mathbf{H}_x^l(\mathbf{BM}+\mathbf{G})\|_2 \leq g^l,$$

where Q is the quantile function of $\mathcal{N}(0,1)$. This leads to the formulation of a second-order cone problem.

Simplification

Consider the following constraints with parameter $\beta^l > 0$

$$\begin{cases} \mathbf{H}_x^l(\mathbf{BM}+\mathbf{G})\mathbf{w}_k^l + \mathbf{H}_x^l(\mathbf{A}x_k + \mathbf{Bm}) \leq \mathbf{K}_x^l \\ \|\mathbf{w}_k^l\|_\infty \leq \beta^l \end{cases}$$

Then, *a posteriori* we can determine the probability

$$\mathrm{P}[\mathbf{H}_x^l \mathbf{x}_k \leq \mathbf{K}_x^l \mid \|\mathbf{w}_k^l\|_\infty \leq \beta^l],$$

given that $\mathbf{w}_k^l \sim \mathcal{N}(0, I)$

Simplification

But we know that

$$\mathrm{P}[\mathbf{H}_x^l \mathbf{x}_k > \mathbf{K}_k^l \mid \|\mathbf{w}_k^l\| \leq \beta^l] \leq \sqrt{e}\beta^l e^{-\frac{(\beta^l)^2}{2}},$$

so for the probabilistic constraints to be satisfied it suffices to choose β^l so that

$$\sqrt{e}\beta^l e^{-\frac{(\beta^l)^2}{2}} \leq \alpha^l$$

Discussion

✓ Affine disturbance feedback and affine state feedback are **equivalent** [Goulart *et al.*, '05]

Discussion

- ✓ Affine disturbance feedback and affine state feedback are **equivalent** [Goulart *et al.*, '05]
- ✓ it is a **suboptimal** choice in the space of measurable functions

Discussion

- ✓ Affine disturbance feedback and affine state feedback are **equivalent** [Goulart *et al.*, '05]
- ✓ it is a **suboptimal** choice in the space of measurable functions
- ✓ But it is a computationally **tractable** approach

Discussion

- ✓ Affine disturbance feedback and affine state feedback are **equivalent** [Goulart *et al.*, '05]
- ✓ it is a **suboptimal** choice in the space of measurable functions
- ✓ But it is a computationally **tractable** approach
- ✓ The problem size **explodes** as the prediction horizon increases

Discussion

- ✓ Affine disturbance feedback and affine state feedback are **equivalent** [Goulart *et al.*, '05]
- ✓ it is a **suboptimal** choice in the space of measurable functions
- ✓ But it is a computationally **tractable** approach
- ✓ The problem size **explodes** as the prediction horizon increases
- ✓ But there are **approximating** techniques such as the *blocking* affine parametrization.

Discussion (cont'd)

✓ Recursive feasibility is **impossible** to guarantee when the disturbance is unbounded

Discussion (cont'd)

- ✓ Recursive feasibility is **impossible** to guarantee when the disturbance is unbounded
- ✓ Recursively feasible formulations are — as expected — too **conservative** [M. Korda *et al.* '11]

Discussion (cont'd)

- ✓ Recursive feasibility is **impossible** to guarantee when the disturbance is unbounded
- ✓ Recursively feasible formulations are — as expected — too **conservative** [M. Korda *et al.* '11]
- ✓ We can however design a **probabilistically resolvable** control scheme [M. Ono, '12]

Further reading

1. M. Lorenzen, F. Dabbene, R. Tempo and F. Allgöwer, *Constraint-Tightening and Stability in Stochastic Model Predictive Control*, arXiv: 1511.03488v1, 2015.
2. M. Oldewurtel, C.N. Jones, M. Morari, *A Tractable Approximation of Chance Constrained Stochastic MPC based on Affine Disturbance Feedback*, 2008.
3. M. Ono, *Closed-Loop Chance-Constrained MPC with Probabilistic Resolvability*, 51st IEEE CDC, Maui, Hawaii, USA, 2012.
4. M. Cannon, B. Kouvaritakis, D. Ng, *Probabilistic tubes in linear stochastic model predictive control*, Systems & Control Letters 58, pp. 747–753, 2009.
5. D. Chatterjee, P. Hokayem, J. Lygeros, *Stochastic receding horizon control with bounded control inputs: a vector space approach*, arXiv:0903.5444v3, 2009.
6. M. Korda, R. Gondhalekar, J. Cigler, and F. Oldewurtel, *Strongly Feasible Stochastic Model Predictive Control*, IEEE CDC, 2011.
7. M.Cannon, P. Couchman and B. Kouvaritakis, *MPC for stochastic systems*, Chapter in *Assessment and Future Directions of Nonlinear Model Predictive Control*, Springer, 2007.

Further reading

8. M. Cannon, *Stochastic Model Predictive Control: State space methods*, Lecture notes available online at http://users.ox.ac.uk/~engs0169/pdf/cannon_ifac08c.pdf

9. H. Kushner, *Introduction to stochastic control*, Holt, Rinehart and Winston Inc., New York, 1971.

10. M. Agarwal, E. Cinquemani, D. Chatterjee and J. Lygeros, *On convexity of stochastic optimization problems with constraints*, European Control Conference (ECC), Budapest, Hungary, 2009.