Current Trends in Model Predictive Control

Alberto Bemporad

http://www.imtlucca.it/alberto.bemporad

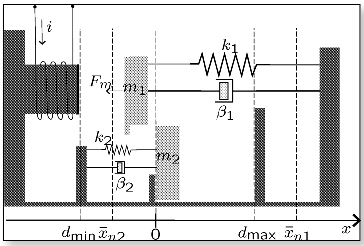
IMT Institute for Advanced Studies Lucca

MPC research is driven by applications

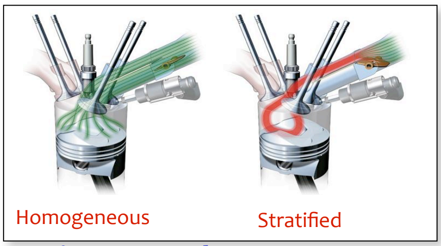
- **Process** control → **linear** MPC (some **nonlinear** too) *1980-2000*
- **Automotive** control → **explicit, hybrid** MPC *2001-2010*
- **Aerospace** systems and UAVs → **linear time-varying** MPC *>2005*
- **Information and Communication Technologies (ICT)** (wireless nets, cloud computers) → **distributed**/**decentralized** MPC *>2005*
- **Energy, finance, automotive** → **stochastic** MPC *>2010*

Automotive applications of MPC

PhD students: Bernardini, Borrelli, Di Cairano, Giorgetti, Ripaccioli, Trimboli (2001-2011)
& Hrovat, Kolmanovsky, Tseng (Ford)



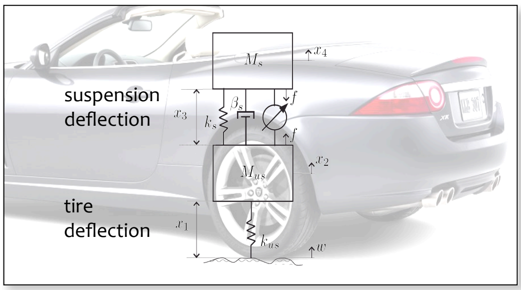
magnetic actuators



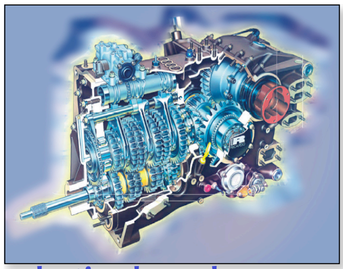
engine control

active steering

traction control



semiactive suspensions



robotized gearbox

Ford JAGUAR CENTRO RICERCHE FIAT DENSO

Linear time-varying MPC

- MPC can easily handle **linear time-varying (LTV)** problems

$$\begin{cases} x_{k+1} &= A_k(t,x(t))x_k + B_k(t,x(t))u_k + f_k(t,x(t)) \\ y_k &= C_k(t,x(t))x_k + D_k(t,x(t))u_k + g_k(t,x(t)) \end{cases}$$

$$E_k(t,x(t))x_k + F_k(t,x(t))u_k \le h_k(t,x(t)) \qquad k = 0,1,\ldots,N-1$$

$$\min \sum_{k=0}^{N} \ell_k(y_k, u_k, r(t+k), t, x(t)) \qquad \ell_k = \text{quadratic function of } y_k, u_k$$

$$\begin{array}{ll} \min\limits_{U} & \frac{1}{2}U'H(t)U + F(t)'U + \cancel{\alpha(t)} \\ \text{s.t.} & G(t)U \le W(t) \end{array}$$

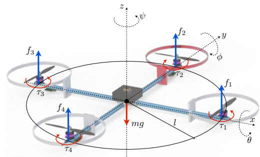
LTV-MPC still leads to a **Quadratic Program (QP)** !

- Applications: time-varying systems (e.g.: aerospace), NL systems

MPC applications in aerospace

- **Linear Time-Varying MPC** for wheel momentum damping by thrust orientation mechanism

(Bemporad, Losa, Piliego, Ramirez-Prado, 2009)

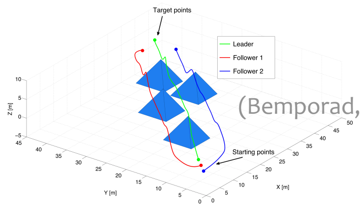


- **LTI MPC** for stabilization, **Hybrid MPC** for navigation of small UAVs

(Bemporad, Pascucci, Rocchi, 2009)

- **Hybrid MPC** for formation flying of small UAVs

(Bemporad, Rocchi, IFAC 2011)



- Decentralized **LTV-MPC** for formation flying

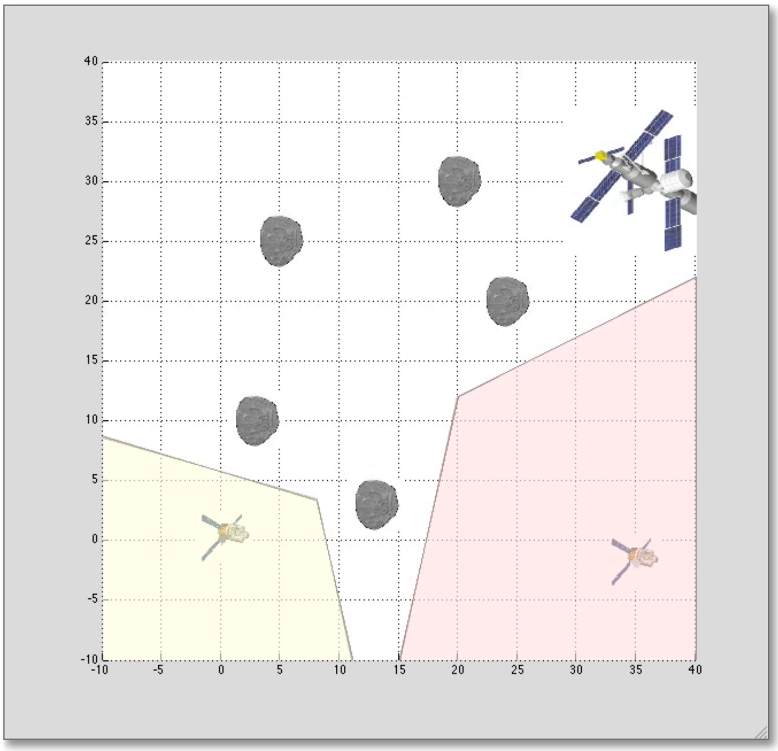
(Bemporad, Rocchi, CDC 2011 ?)

- Many other contributors:

- A. Richards, P. Trodden (Bristol, UK)
- J. Maciejowski, E.N. Hartley (Cambridge, UK)
- G. Balas, F. Borrelli, T. Keviczky (Minnesota)
- R. Murray, W.B. Dunbar (Caltech)
- J. How, L. Breger, M. Tillerson (MIT)
- (...)

Example: LTV-MPC of UAVs

- Two unmanned aerial vehicles (UAVs) avoiding each other and obstacles



- Feasible space approximated as a (time-varying) polyhedron
- Each UAV solves its own MPC problem
- Previous optimal sequences exchanged to improve accuracy in predicting future locations of the other UAV

esa

ROBMPC project
Robust Model Predictive Control (MPC) for Space Constrained Systems

MPCSofT Toolbox for MATLAB

(Bemporad, 2010-2011)

Linear MPC: summary of computation effort

MPC type	off-line computations	on-line computations
LTI model, **explicit** MPC	build QP/LP problem, solve multiparametric problem	evaluate PWA function
LTI model, **implicit**	build QP/LP problem	solve QP/LP problem
LTV model	none	build QP/LP problem solve QP/LP problem

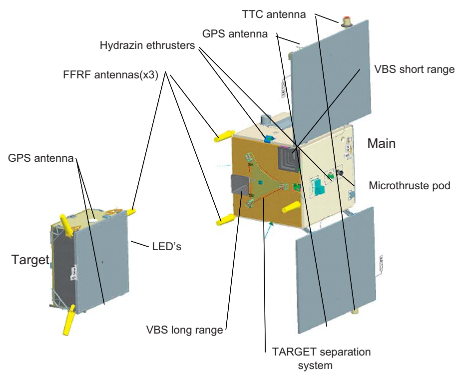
LTI = Linear Time-Invariant

LTV = Linear Time-Varying

MPC applications in aerospace

- **PRISMA** project for autonomous formation flying (S. Persson, S. Veldman, P. Bodin, 2009)

http://www.prismasatellites.se/



- MPC objective: **minimize fuel consumption** subject to keeping motion within a **box constraint** (solved by **linear programming**)

$$\min_{dV(t)} \sum_{t=0}^{\infty} \|dV(t)\|_1 \quad \text{s.t.} \quad \|r - y_r\|_1 \le y_{\text{box}}$$

MPC controller **operational in space !**

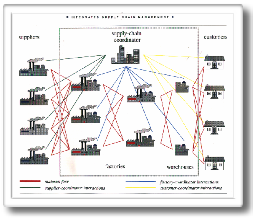
Decentralized MPC

Control of large-scale systems


traffic networks

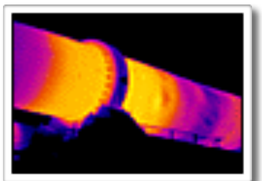

smart grids

- Characterized by spatial distribution
- Lack of centralized information


supply chains


cooperating vehicles

- Lack of centralized computing capabilities


distributed parameter structures


water networks

Need for a decentralized scheme

Centralized vs decentralized/distributed control

Centralized control:

- Need a **global model** of the overall system (and its maintenance)
- Complexity **not scalable** with plant size
- Computation complexity may become prohibitive

- Control design hard to commission, start-up, and **maintain** (many tuning "knobs")
- High **risk**: a single controller is running the whole plant
- Good **theoretical** properties (e.g. closed-loop stability)

Centralized vs decentralized/distributed control

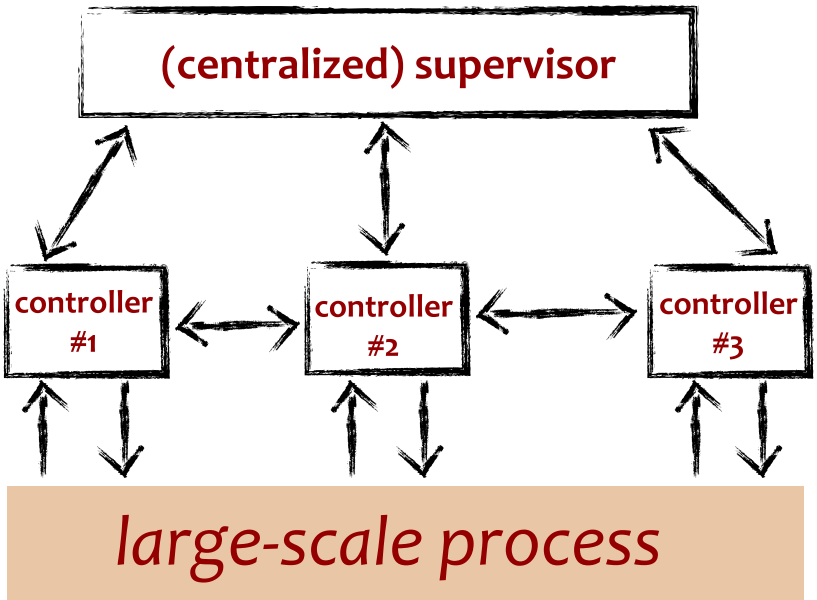
Decentralized control:

- **Local submodels** of system components are enough
- Computational tasks are **parallelized**, each task is simple
- Data gathering is simpler (**local measurements** used only locally)
- Commissioning, start-up, and **maintenance** more practical (controller updates do not require a whole plant shutdown)
- **Global properties** (stability, performance) hard to assess, especially in the presence of input/state constraints

Careful cooperation of controllers is needed to ensure global properties (such as stability and constraint fulfillment)

Typical decentralized approach

- Measure/estimate **local** states
- Compute control actions **locally**
- **Exchange** decisions with neighbors, possibly reiterate local computations
- Apply the current command input to local actuator(s)
- Possibly interact with upper level of decision making (**hierarchical control**)



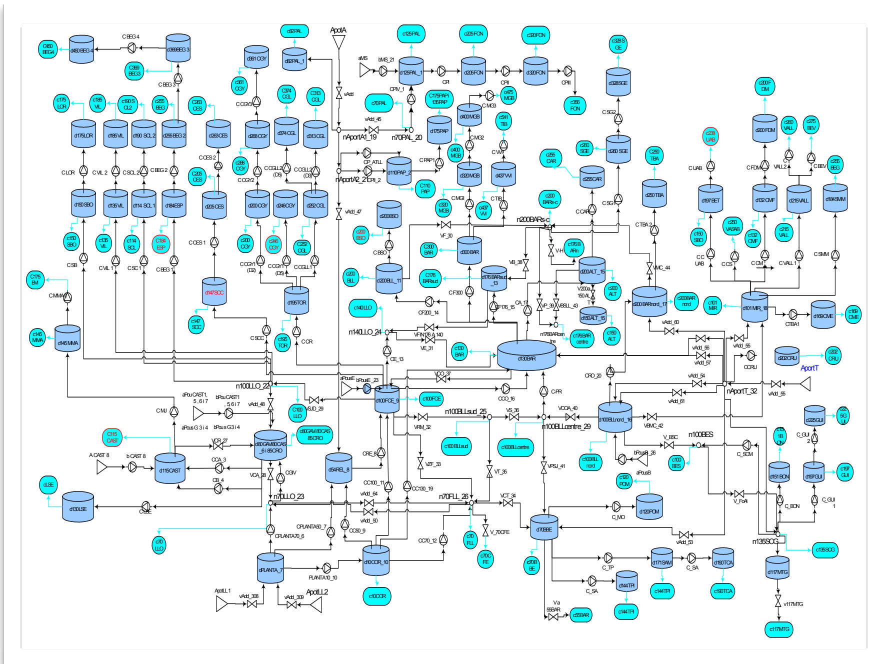
Main issues: Global closed-loop stability ? Feasibility of global constraints ? Loss of performance w.r.t. centralized control ?

Decentralized/distributed MPC

submodels	constraints	intersampling iterations	broadcast prediction	state constraints	stability constraints	authors
coupled	local inputs	no	no	no	none	Alessio, Barcelli, Bemporad
coupled	local inputs	yes	no	no	none	Venkat, Rawlings, Wright
coupled	local inputs	yes	yes	no	none	Mercangöz, Doyle
decoupled	local inputs	no	yes	yes	compatibility	Dunbar, Murray
decoupled		no	yes	yes	none	Keviczy, Borrelli, Balas
coupled	local states	no	yes	yes	contractive	Jia, Krogh
coupled/NL	local inputs	no	no	no	contractive	Magni, Scattolini

Alternative approach: ***distribute the optimization*** problem associated with the centralized MPC formulation, instead of distributing the problem formulation

Distributed MPC of Barcelona water network



- General overview:

Municipalities supplied	23
Supply area	424 km^2
Population supplied	2.922.773
Average demand	7 m^3/s

- Network parameters:

Pipes length	4.645 km
Pressure floors	113
Sectors	218

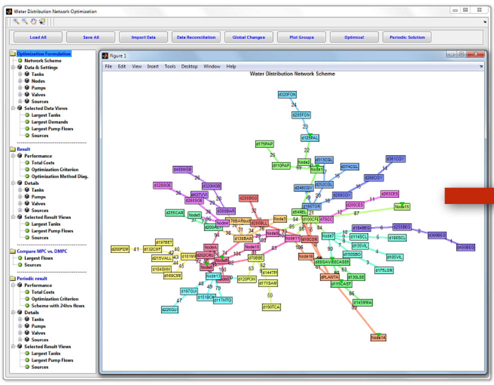
- Facilities

Remote stations	98
Water storage tanks	81
Valves	64
Flow meters	92
Pumps / Pumping stations	180 / 84
Chlorine dosing devices	23
Chlorine analyzers	74

European FP7-ICT project WIDE
"DEcentralized and WIreless Control of Large-Scale Systems"

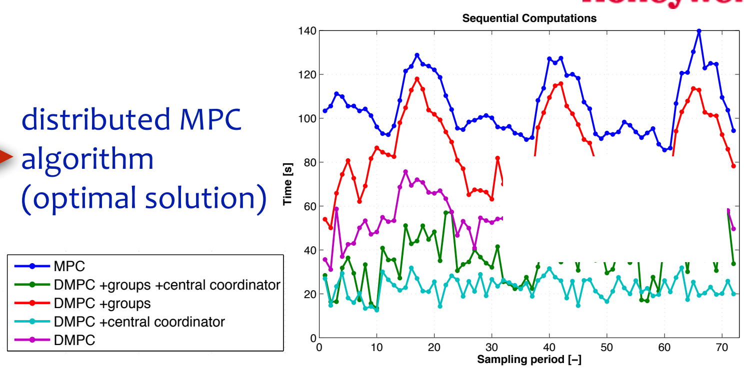
DMPC of Barcelona water network

(Trnka, Pekar, Havlena, IFAC 2011)

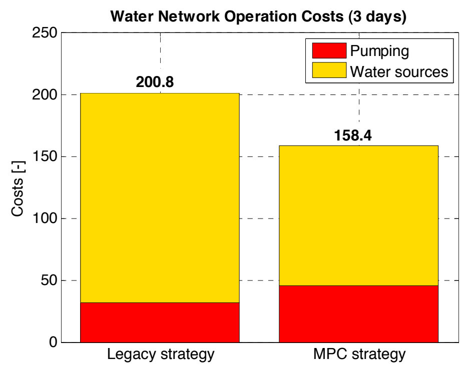


Network separation to groups

distributed MPC algorithm (optimal solution)

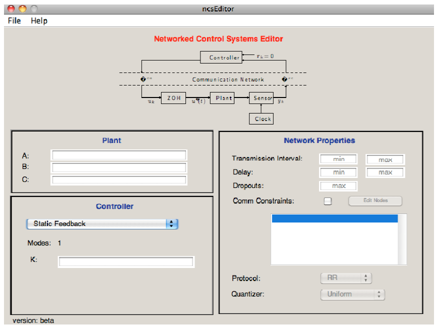


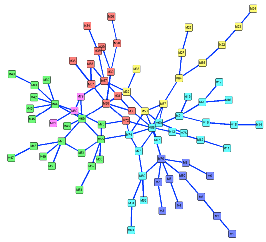
- Benefits evaluated on 3 days historic data set
- Benefits evaluation is complicated by water accumulation (different final accumulation for different control strategies)
- To get comparable data MPC was forced to fill tanks to the same final levels as in historical data (sub-optimal)
- ~20% direct cost savings (pumping and water sources)
- Indirect savings by smooth MV's operation -> leakage prevention by small pressure surges and reduced equipment tear & wear

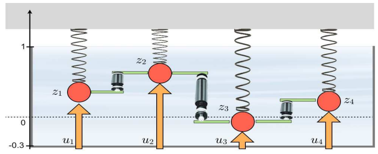


WIDE Toolbox for MATLAB

- **Networked control systems**: modeling, stability analysis, linear control synthesis
- **Model management** of large-scale systems (model reduction, create submodels, ...)
- **Data acquisition** from physical WSN (Telos motes, E-Senza's nodes)
- **WSN simulation** (generate TrueTime code)
- **MPC control**: decentralized, hierarchical, network-aware





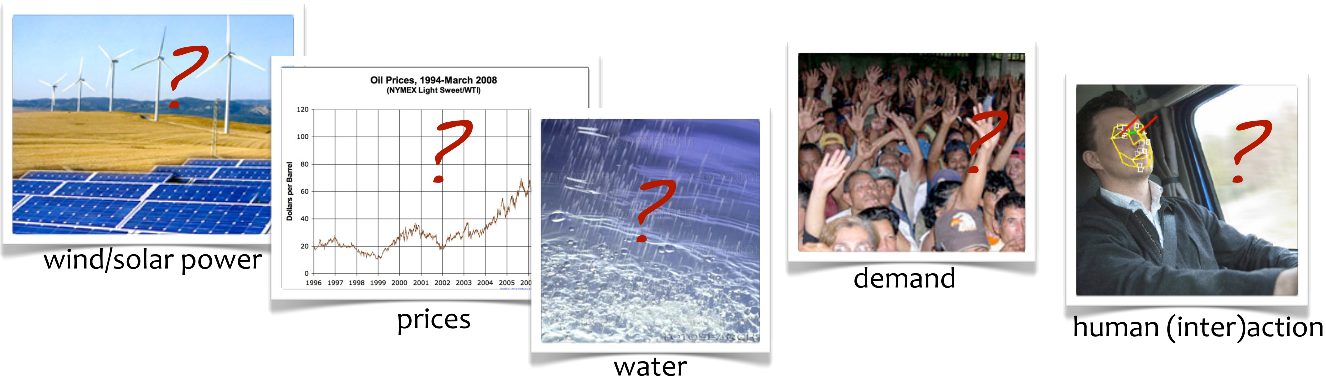


- Available for public download on September 1st 2011
  http://ist-wide.dii.unisi.it/

Stochastic MPC

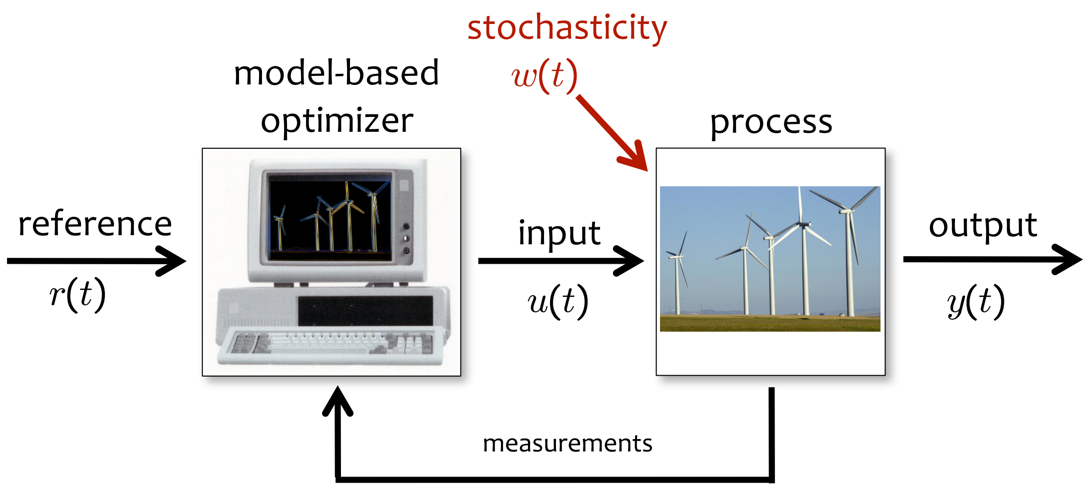
Stochastic systems

- In many control problems decisions must be taken under **uncertainty**
- **Robust** control approaches do not model uncertainty (only assume that is bounded) and pessimistically consider the worst case
- **Stochastic models** provide additional information about uncertainty



Need to include stochastic models in control problem formulation

Stochastic Model Predictive Control (SMPC)



*Use a **stochastic** dynamical **model** of the process to **predict** its possible future evolutions and choose the “best” **control** action*

Stochastic Model Predictive Control

- At time t: solve a **stochastic optimal control** problem over a finite future horizon of N steps:

$$\begin{array}{ll}\min\limits_{u} & E_w\left[\sum\limits_{k=0}^{N-1} \ell(y_k - r(t+k), u_k)\right] \\ \text{s.t.} & x_{k+1} = f(x_k, u_k, w_k) \\ & y_k = g(x_k, u_k, w_k) \\ & u_{\min} \leq u_{t+k} \leq u_{\max} \\ & y_{\min} \leq y_k \leq y_{\max}, \quad \forall w \\ & x_0 = x(t)\end{array}$$

$x(t)$ = process state
$u(t)$ = manipulated vars
$y(t)$ = controlled output
$w(t)$ = **stochastic disturbances**

- Only apply the first optimal move $u^*(t)$, discard $u^*(t+1)$, $u^*(t+2)$, ...
- At time t+1: Get new measurement $x(t+1)$, repeat the optimization. And so on ...

Linear stochastic MPC w/ discrete disturbance

- Linear stochastic prediction model

$$x(t+1) = A(w(t))x(t) + B(w(t))u(t) + E(w(t))$$

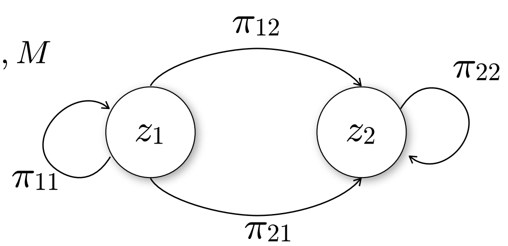
- Discrete disturbance

$$w(t) \in \{w_1, \ldots, w_s\} \qquad p_j(t) = \mathbf{Pr}[w(t) = w_j] \qquad \sum_{j=1}^{s} p_j(t) = 1$$

- Probabilities $p_j(k)$ can have their own dynamics. Example: Markov chain

$$\pi_{ih} = \mathbf{Pr}[z(t+1) = z_h \mid z(t) = z_i], \ i, h = 1, \ldots, M$$

$$p_j(t) = \begin{cases} e_{1j} & \text{if } z(t) = z_1 \\ \vdots & \vdots \\ e_{Mj} & \text{if } z(t) = z_M \end{cases}$$



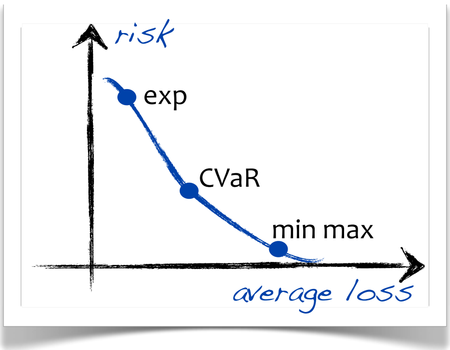
- Discrete distributions can be estimated from historical data (and adapted on-line)

Cost functions for SMPC to minimize

performance $\longrightarrow J(u,w) \triangleq \sum_{k=0}^{N-1} \ell(y_k - r(t+k), u_k)$

- **Expected performance**

$$\min_u E_w\left[J(u,w)\right]$$



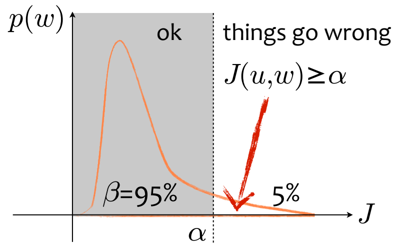
- Tradeoff between **expected performance & risk**

$$\min_u E_w\left[J(u,w)\right] + \rho \,\mathrm{Var}\left[J(u,w)\right]$$

- Conditional Value-at-Risk (**CVaR**)

$$\min_{u,\alpha}\left\{\alpha + \frac{1}{1-\beta} E[\max(J(u,w)-\alpha, 0)]\right\}$$ (Rockafellar, Uryasev, 2000)

= minimize expected loss when things go wrong (convex if J convex !)



- **Min-max**

$$\min_u\{\max_w J(u,w)\}$$ =minimize worst case performance

Linear stochastic MPC formulation

- Performance index $\quad \min E_w\left[x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k\right]$

- Goal: ensure mean-square convergence $E[x'(t)x(t)] = 0$ (for H=0)

- The existence of a stochastic Lyapunov function $V(x) = x'Px$

$$E_{w(t)}[V(x(t+1)] - V(x(t)) \leq -x(t)'Lx(t), \quad \forall t \geq 0$$

(Morozan, 1983)

ensures mean-square stability

- Existing SMPC approaches:

(Schwarme & Nikolaou, 1999)
(Wendt & Wozny, 2000)
(Batina, Stoorvogel, Weiland, 2002)
(van Hessem & Bosgra 2002)
(Munoz de la Pena, Bemporad, Alamo, 2005)
(Couchman, Cannon, Kouvaritakis, 2006)
(Primbs, 2007)
(Bemporad, Di Cairano, 2005)
(Ono, Williams, 2008)
(Oldewurtel, Jones, Morari, 2008)
(Bernardini & Bemporad, 2009)

Stochastic program

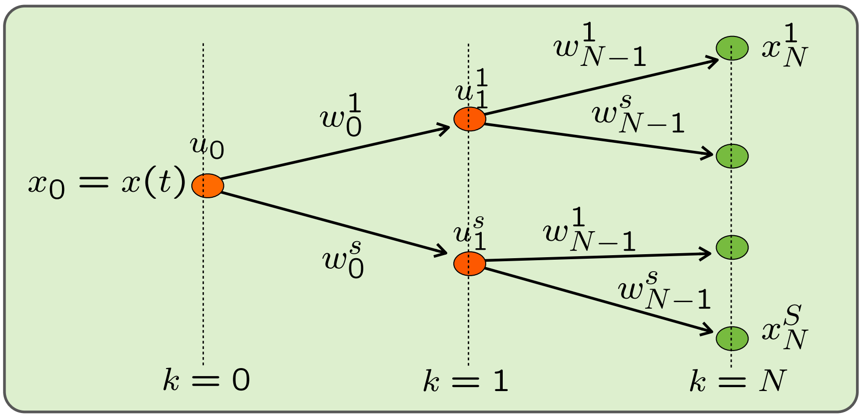
- Enumerate all possible scenarios $\{w_0^j, w_1^j, \ldots, w_{N-1}^j\},\ j = 1, \ldots, S$
- Each scenario has probability $p^j = \prod_{k=0}^{n-1} \mathbf{Pr}[w_k = w_k^j]$
- Each scenario has its own evolution $x_{k+1}^j = A(w_k^j)x_k^j + B(w_k^j)u_k^j$
- Expectations become simple sums

$$\min E_w \left[x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \right]$$

$$\min \sum_{j=1}^{S} p^j \left((x_N^j)' P x_N^j + \sum_{k=0}^{N-1} (x_k^j)' Q x_k^j + (u_k^j)' R u_k^j \right)$$

This is again a (large & sparse) QP

Scenario tree



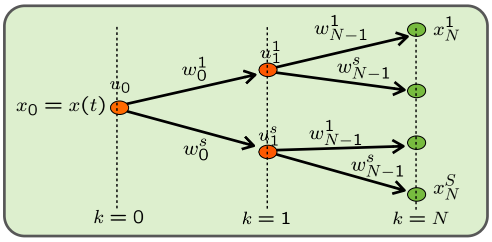
- Scenario = path on the tree
- Number S of scenarios = number of leaf nodes
- Some paths can be removed if their probability is very small (at your own risk)

$\min \ldots + p^j (x_k^j)' Q x_k^j + \ldots$

$y_{\min} \le y_k \le y_{\max},\ \forall w$

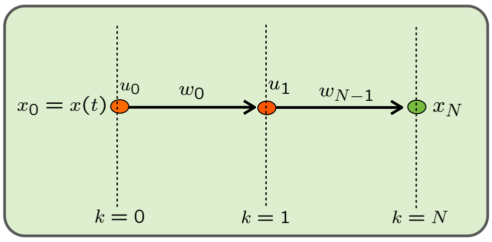
- **Causality constraint:** $u_k^j = u_k^h$ when scenarios j and h share the same node at prediction time k (for example: $u_0^j \equiv u_0^h$ at root node $k=0$)

Scenario enumeration



scenario tree

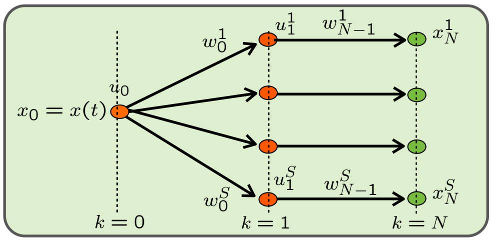
Causality is exploited: decision u_k only depends on past disturbance realizations $\{w_0, w_1, \ldots, w_{k-1}\}$



deterministic

Only a sequence of disturbances is considered

- frozen-time: $w_k \equiv w(t), \ \forall k$ (causal prediction)
- anticipative action: $w_k \equiv w(t+k)$ (non-causal)
- "expected" problem: $w_k = E[w(t+k)|t]$ (causal)



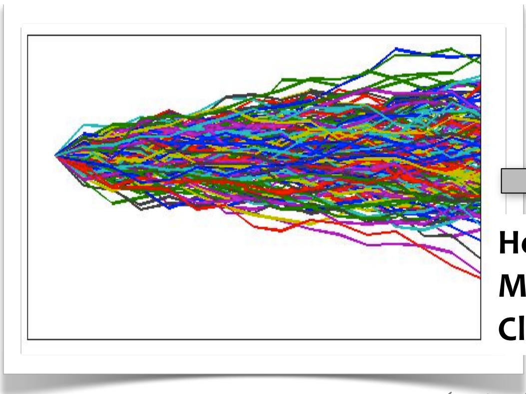
scenario "fan"

- generate a set of scenarios (Monte Carlo simulation)
- decision u_k also depends on future disturbance realizations $\{w_k, w_{k+1}, \ldots, w_{N-1}\}$

Scenario tree generation from data

- Scenario trees can be generated by **clustering** sample paths
- Paths can be obtained by Monte Carlo simulation of (arbitrarily complex) models, or from historical data

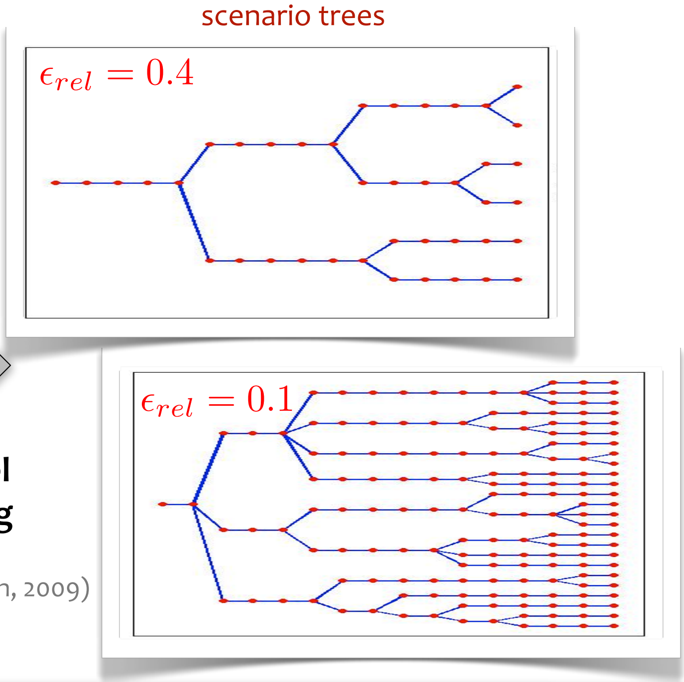
scenario "fan" (collection of sample paths)



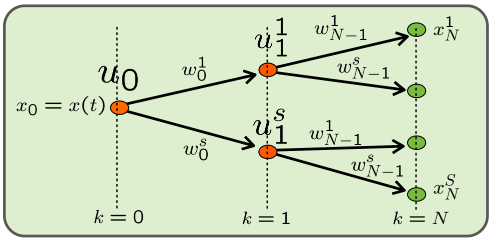
Heuristic Multilevel Clustering

(Heitsch, Römisch, 2009)

scenario trees

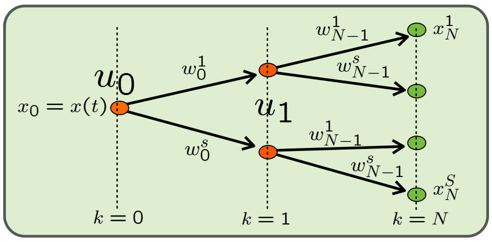


Open-loop vs. closed-loop prediction



closed-loop prediction

A proper move u is optimized to counteract each possible outcome of the disturbance w



open-loop prediction

Only a sequence of inputs $\{u_0, u_1, \ldots, u_{N-1}\}$ is optimized, the same u must be good for all possible disturbance w

- Intuitively: OL prediction is more conservative than CL in handling constraints
- OL problem = CL problem + additional constraints $u^j \equiv u, \ \forall j = 1, \ldots, S$ (=less degrees of freedom)

Linear stochastic stabilization

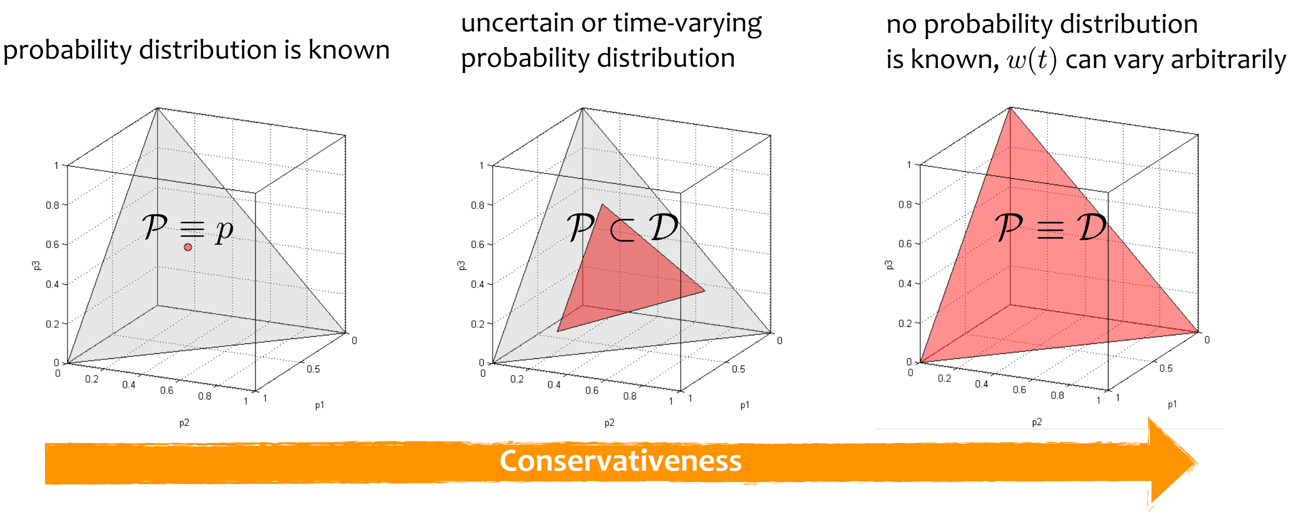
- Assume $w(t) \in \{w_1, \ldots, w_s\}$ and **constant** probability $p(t) \equiv p, \ \forall t$
- The **stochastic convergence** condition $E_{w(t)}[V(x(t+1)] - V(x(t)) \leq -x(t)'Lx(t)$ can be recast as the **LMI** condition

$$\begin{bmatrix} Q & Q & \sqrt{p_1}(A_1Q+B_1Y)' & \cdots & \sqrt{p_s}(A_sQ+B_sY)' \\ Q & W & 0 & \cdots & 0 \\ \sqrt{p_1}(A_1Q+B_1Y) & 0 & Q & & \\ \vdots & \vdots & & \ddots & \\ \sqrt{p_s}(A_sQ+B_sY) & 0 & & & Q \end{bmatrix} \succeq 0$$

- The Lyapunov function is $V(x) = x'Q^{-1}x$
- Mean-square stability guaranteed by linear feedback $u(k) = Kx(k), \ K = YQ^{-1}$
- A minimum decrease rate L can be imposed

Linear stochastic stabilization

- The approach can be generalized to uncertain probabilities $p(t) \in \mathcal{P}$ (Example: time-varying probabilities)



- If $\mathcal{P} \equiv \mathcal{D}$ we have a **robust** control problem (*robust convergence*)
- The more information we have about the probability distribution $p(t)$ of $w(t)$ the less conservative is the control action

Stabilizing SMPC

(Bernardini, Bemporad, CDC'09)

- Impose stochastic stability constraint in SMPC problem (=quadratic constraint w.r.t. u_0)

$$\begin{array}{ll} \min_u & E_w\left[\sum_{k=0}^{N-1} \ell(x_k, u_k)\right] \\ \text{s.t.} & x_{k+1} = A(w_k)x_k + B(w_k)u_k \\ & E\left[V(A(w_0)x_0 + B(w_0)u_0)\right] \leq x_0'(Q^{-1} - L)x_0 \\ & x_0 = x(t) \end{array}$$

performance and stability are decoupled

- SMPC approach:
  1. Solve LMI problem off-line to find stochastic Lyapunov fcn $V(x) = x'Q^{-1}x$
  2. Optimize stochastic performance based on scenario tree

Theorem: The closed-loop system is as. stable in the mean-square sense

- SMPC can be generalized to handle **input and state constraints**

Note: *recursive feasibility* guaranteed by backup solution $u(k) = Kx(k)$

A few sample applications of SMPC

- **Financial engineering:** dynamic hedging of portfolios replicating synthetic options

  (Bemporad, Bellucci, Gabbriellini, 2009)
  (Bemporad, Gabbriellini, Puglia, Bellucci, 2010)
  (Bemporad, Puglia, Gabbriellini, 2011)

- **Energy systems:** power dispatch in smart grids, optimal bidding on electricity markets

  (Patrinos, Trimboli, Bemporad 2011)
  (Puglia, Bernardini, Bemporad 2011)

- **Automotive control:** energy management in HEVs, adaptive cruise control (human-machine interaction)

  (Bichi, Ripaccioli, Di Cairano, Bernardini, Bemporad, Kolmanovsky, CDC 2010)

- **Networked control:** improve robustness against communication imperfections

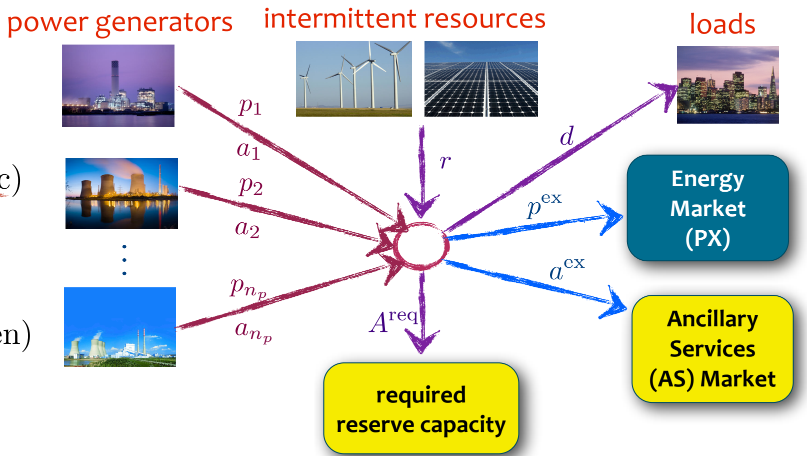
  (Bernardini, Donkers, Bemporad, Heemels, NECSYS 2010)

SMPC for real-time market-based power dispatch

- A Balance Responsible Party (BRP) is the only legal entity trading on the *energy (PX)* and *ancillary service (AS)* markets
- **Objective:** Minimize (expected) costs via efficient use of intermittent resources, and maximize (expected) profits by trading on PX and AS markets
- **Constraints:** Grid capacity constraints, rate limits, load balancing, AS balancing

p_i = power injection (decision)
a_i = reserve capacity (decision)
r = intermittent generation (stochastic)
d = demand (stochastic)
p^{ex} = power exchanged (decision)
p^{as} = power exchanged (decision)
A^{req} = required reserve for BRP (given)

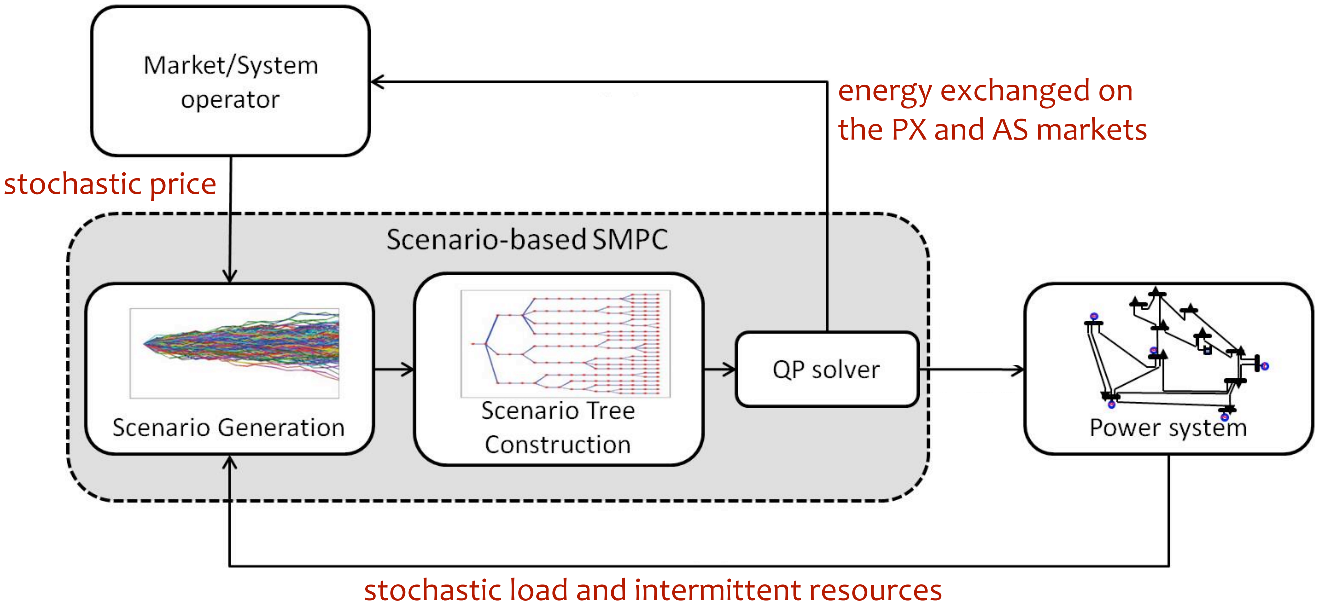
energy prices are also stochastic



SMPC for real-time market-based power dispatch

(Patrinos, Trimboli, Bemporad 2011)

- Stochastic MPC architecture



SMPC for market-based optimal power dispatch

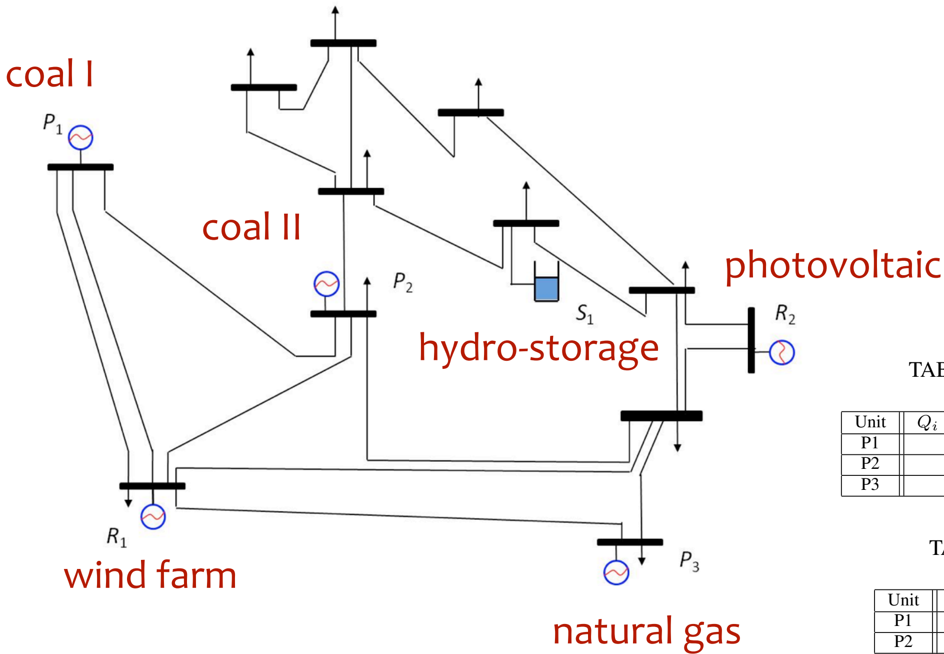


TABLE I: Generator Cost Data

Unit	Q_i (\$/MWh^2)	q_i (\$/MWh)	c_i (\$)
P1	0.009	30.375	398.025
P2	0.0225	73.35	292.275
P3	0.0488	61.488	489.952

TABLE II: Generator Data

Unit	$p_i^{\min}$	$p_i^{\max}$	$\Delta p_i^{\min}$	$\Delta p_i^{\min}$
P1	450	1100	-250	250
P2	50	500	-200	200
P3	50	100	-75	75

TABLE III: Storage Data

Unit	$x_i^{\min}$	$x_i^{\max}$	$\Delta x_i^{\min}$	$\Delta x_i^{\min}$	α_i	α_i^{c}	α_i^{d}
S1	15	300	-120	120	0.95	0.85	0.90
	$u_i^{c,\min} = u_i^{d,\min}$			$u_i^{c,\max} = u_i^{d,\max}$			
	0			300			

SMPC for market-based optimal power dispatch

- Numerical results

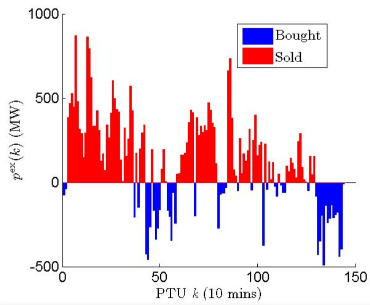
Exact knowledge future uncertainty

Time-varying expectations used for future uncertainty

SMPC: as tree density increases, performance gets better, although numerical complexity gets larger

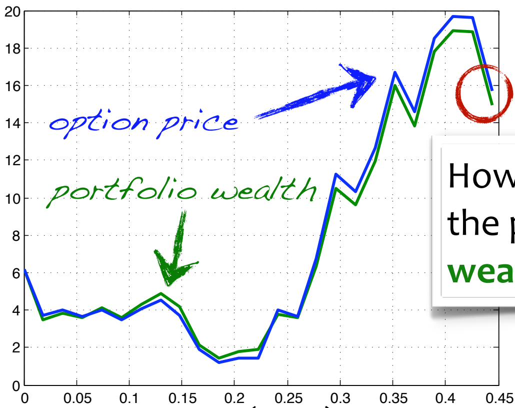
Algorithm	Storage Cost	No Storage Cost	Avg # of nodes
Prescient-OC	6427979	6879741	
CE-MPC	9778750	9819518	
SSMPC ($e_{\rm rel} = 0.1$)	7134582	7245962	350
SSMPC ($e_{\rm rel} = 0.2$)	7144011	7249401	335
SSMPC ($e_{\rm rel} = 0.3$)	7148494	7250207	172
SSMPC ($e_{\rm rel} = 0.4$)	7179848	7264505	87
SSMPC ($e_{\rm rel} = 0.5$)	7224912	7267497	50
SSMPC ($e_{\rm rel} = 0.6$)	7239985	7277410	38
SSMPC ($e_{\rm rel} = 0.7$)	7259491	7298023	31
SSMPC ($e_{\rm rel} = 0.8$)	7255246	7312092	26
SSMPC ($e_{\rm rel} = 0.9$)	7260424	7318643	22
SSMPC ($e_{\rm rel} = 1.0$)	7260424	7318642	20

power exchanged with grid

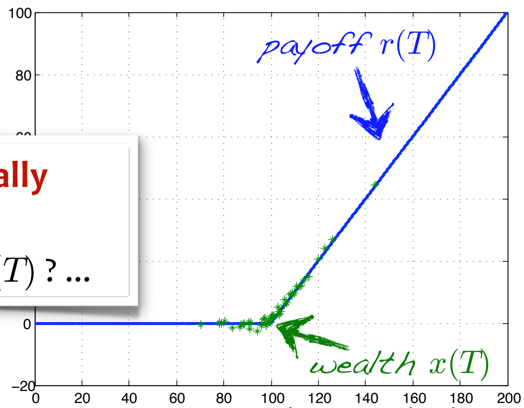


Dynamic hedging problem for financial options

- The financial institution sells a synthetic option to a customer and gets $x(0)$ (€)
- Such money $x(0)$ is used to create a portfolio $x(t)$ of n underlying assets (e.g., stocks) whose prices at time t are $w_1(t)$, $w_2(t)$, ..., $w_n(t)$
- At the expiration date T, the option is worth the payoff $r(T)$ = wealth (€) to be returned to the customer



How to **adjust dynamically** the portfolio so that **wealth** $x(T) =$ **payoff** $r(T)$? ...



.. for any price realization $w_i(t)$?

Portfolio dynamics

- Portfolio wealth at time t:

$$x(t) = u_0(t) + \sum_{i=1}^{n} w_i(t)u_i(t)$$

money in bank account (risk-free asset)

price of asset $\#i$ (stochastic process)

number of assets $\#i$

Example: $w_i(t)$ = **log-normal** model (used in Black-Scholes' theory)

$$dw_i = (\mu dt + \sigma dz_i)w_i$$

geometric Brownian motion

- Assets traded at **discrete-time** intervals under the *self-balancing constraint*:

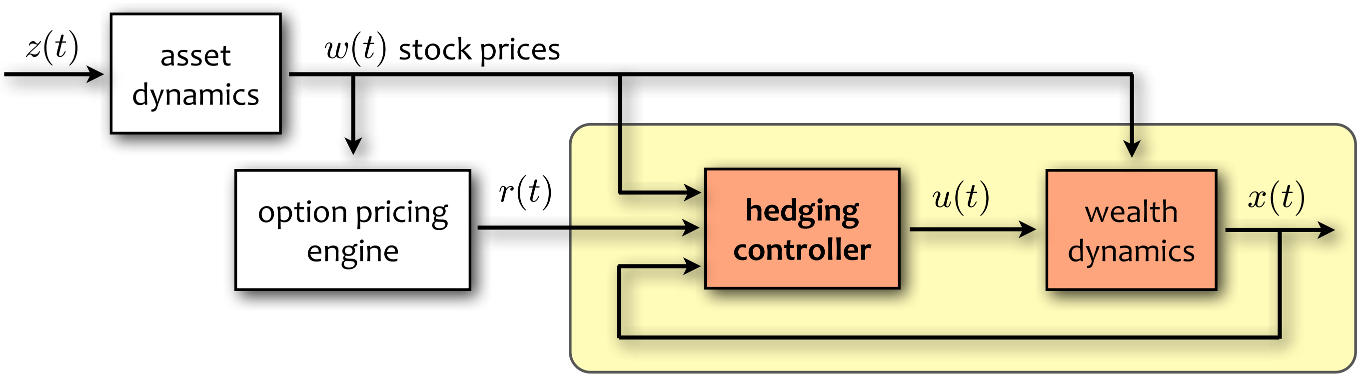
$$x(t+1) \;=\; (1+r)x(t) + \sum_{i=0}^{n} b_i(t)u_i(t)$$

r = interest rate

$$b_i(t) \triangleq w_i(t+1) - (1+r)w_i(t)$$

Option hedging = linear stochastic control

- Block diagram of dynamic option hedging problem:



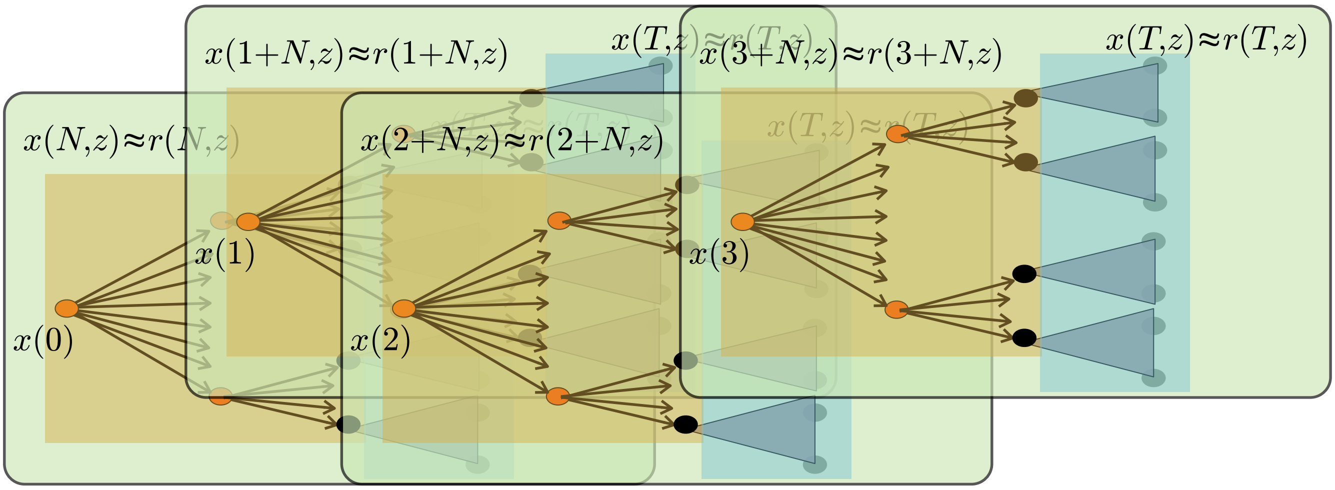
- Payoff function example: $r(T) = \max\{w(T) - K, 0\}$ **European call**

- **Control objective:** $x(T)$ should be as close as possible to $r(T)$, for any possible realization of the asset prices $w(t)$ *("tracking w/ disturbance rejection")*

SMPC for dynamic option hedging

- Stochastic finite-horizon optimal control problem:

$$
\begin{array}{ll}
\min\limits_{\{u(k,z)\}} & \text{Var}_z\left[x(t+N,z)-r(t+N,z)\right] \\
\text{s.t.} & x(k+1,z)=(1+r)x(k,z)+\sum\limits_{i=0}^{n} b_i(k,z)u_i(k,z),\ \ k=t,\ldots,t+N \\
 & x(t,z)=x(t)
\end{array}
$$



SMPC for dynamic option hedging

- Drawback: the longer the horizon N, the largest the number of scenarios !

- Special case: use $N{=}1$!

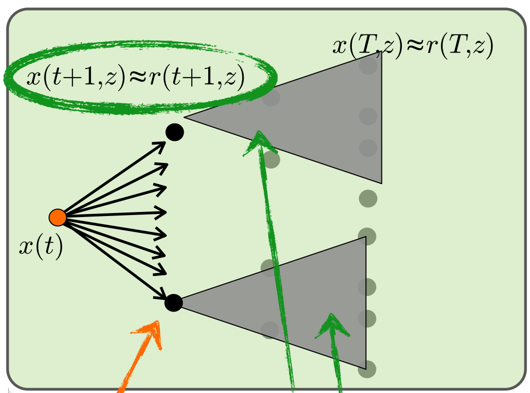
$$
\begin{array}{ll}
\min\limits_{u(t)} & \text{Var}_z\left[x(t+1,z)-r(t+1,z)\right] \\
\text{s.t.} & x(t+1,z)=(1+r)x(t)+\sum\limits_{i=0}^{n} b_i(t,z)u_i(t)
\end{array}
$$

minimum variance control !

✓ **Only one** vector $u(t)$ to optimize

✓ No further branching, so we can generate **a lot** of scenarios for z! (example: 1000)

◉ Need to compute target wealth $r(t{+}1,z)$ for all z



Perfect hedging assumption from time t+1 to T

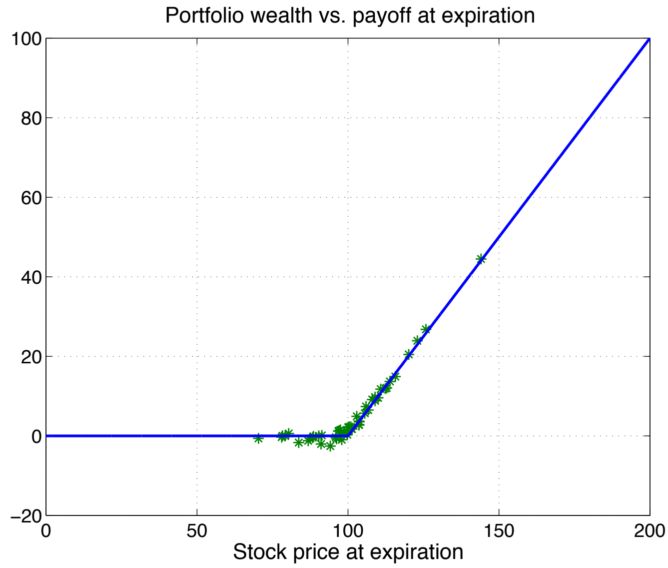
Optimize up to time t+1

On-line optimization: very simple least squares problem with n variables !

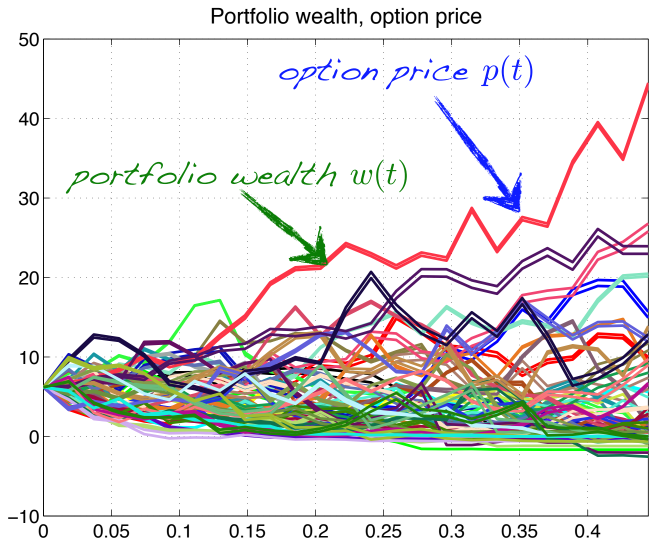
(n = number of traded assets)

Example: Hedging an European call

(Bemporad, Gabbriellini, Puglia, Bellucci, 2010)

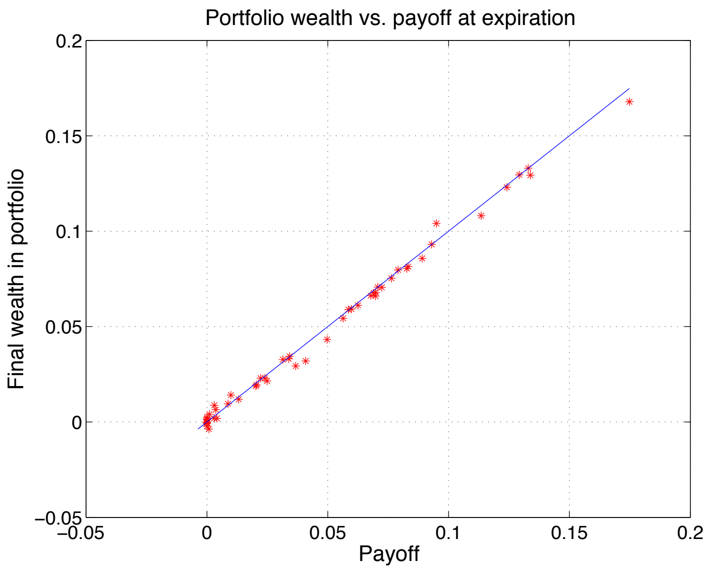


- Black-Scholes model (=log-normal)
- volatility$=0.2$, risk-free$=0.04$
- $T=24$ weeks ($\Delta t=1$ week)
- 50 simulations
- $M=100$ scenarios
- Pricing method: Monte Carlo sim.
- **SMPC**



- CPU time $=$ 7.52 ms per SMPC step (Matlab R2009 on this mac)

Example: Hedging an exotic option



- Black-Scholes model (=log-normal)
- volatility$=0.2$
- $T=24$ weeks (hedging every week)
- 50 simulations
- $M=100$ scenarios
- risk-free$=0.04$
- Pricing method: Monte Carlo sim.
- **SMPC: Trade underlying stock & European call with maturity $t+T$**

$$p(T) = \max\left\{0, C + \min_{i\in\{1,\ldots,N_{\text{fix}}\}} \frac{x(t_i)-x(t_{i-1})}{x(t_{i-1})}\right\}$$

"Napoleon cliquet" option

$t_i=0,8,16,24$ weeks

- CPU time $=$ 1625 ms per SMPC step (Matlab R2009 on this mac)

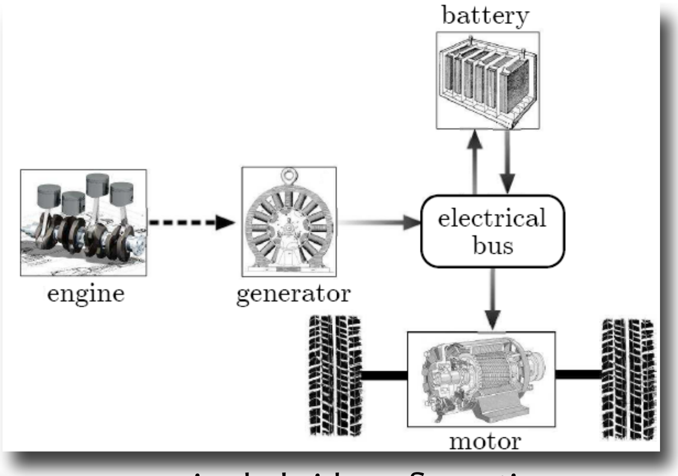
SMPC for hybrid electric vehicles

(Bichi, Ripaccioli, Di Cairano, Bernardini, Bemporad, Kolmanovsky, CDC 2010)

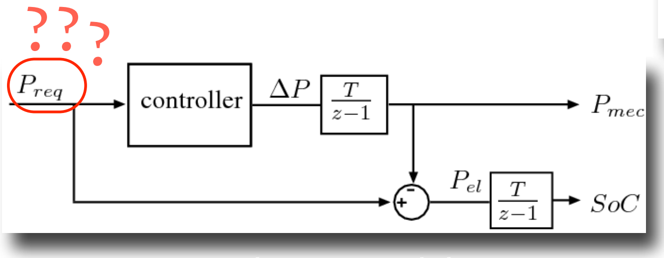
How to split **power** optimally among different power sources in **HEVs** (hybrid electric vehicles) to match power demanded by driver ?

Existing approaches

- Deterministic Dynamic Programming (DDP)
- Stochastic Dynamic Programming (SDP)
- Rule-based
- Game Theory
- Deterministic (hybrid) MPC



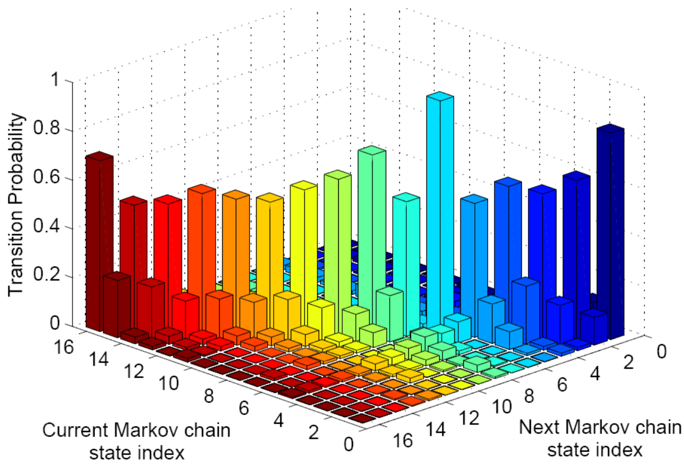
series hybrid configuration



Prediction model

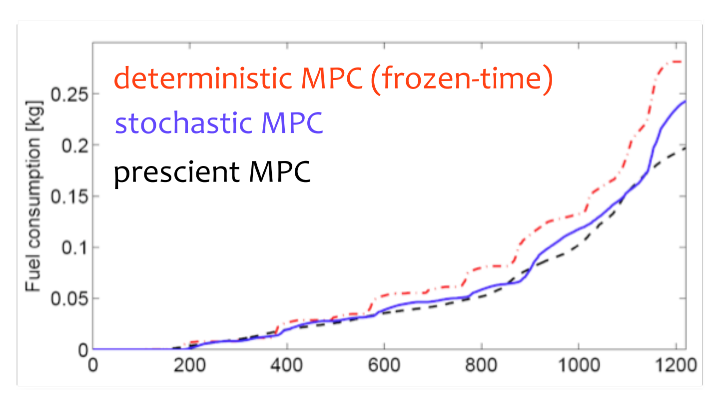
Stochastic model of power demand

- Power demanded by driver modeled as a **Markov chain**

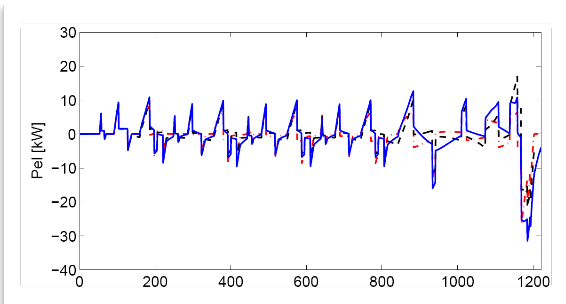


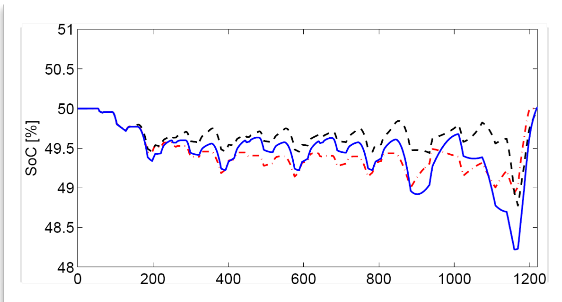
- requested power quantized in 16 levels
- Markov chain is modeling the probabilities of transition from one level to another
- transition probabilities estimated off-line on a collection of driving cycles (FTP, NEDC, 10-15 Mode)

Stochastic vs. prescient and deterministic MPC



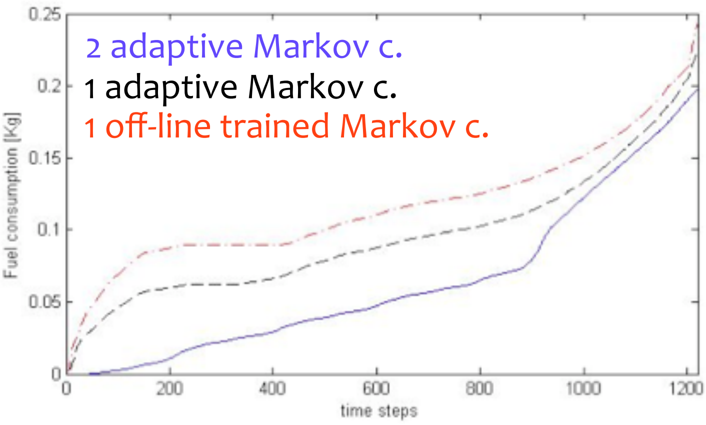
driver model	fuel cons. [kg]	fuel improv. [%]
frozen-time MPC	0.281	-
stochastic MPC	0.243	13.5
prescient MPC	0.197	29.8



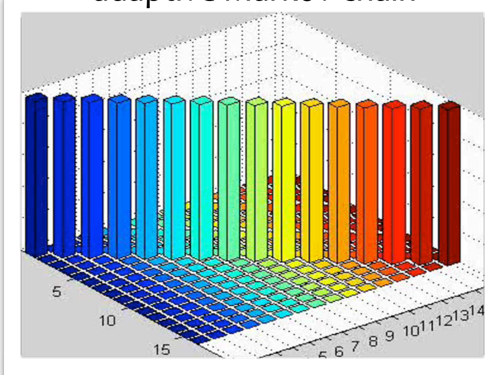


A single Markov chain is tuned off-line

SMPC results (NEDC driving cycle)



adaptive Markov chain



pretty close to having the crystal ball !

driver model	NEDC	FTP	10-15 mode
deterministic	0.281	0.533	0.125
1 Markov chain	0.244	0.323	0.091
2 Markov chain	0.224	0.323	0.089
2 MC adaptive	0.198	0.325	0.088
prescient	0.197	0.320	0.071

Conclusions and open research issues

- **MPC** is a very rich control methodology, with a variety of variants to handle different problems (large-scale, decentralized, networked, stochastic, ...) in different application domains (automotive, energy, water, aerospace, ...)

dynamical model + optimization algorithm = MPC

- **Explicit MPC** very good for small-size MPC problems based on LTI models that require fast sampling and simple control code. **Theory: mature**

- **Hybrid MPC** very versatile MPC framework for complex problems involving logic constraints. **Theory: mature**

- **Decentralized/distributed MPC**:
  - **Theory**: some contributions exist, **not yet mature**. A lot to gain from distrib. optim.

- **Stochastic MPC**:
  - **Theory**: some contributions exist, but large space for new approaches

The End