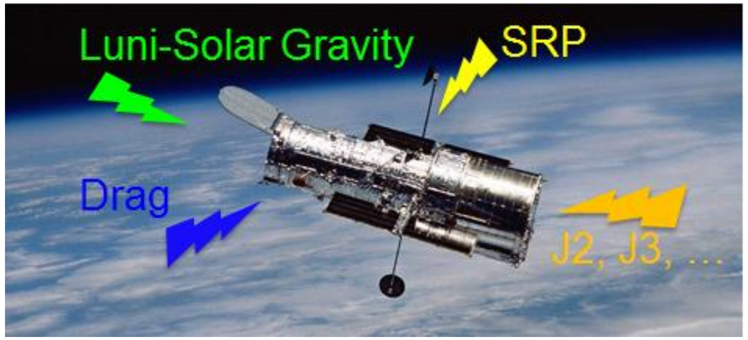
Drift Counteraction Optimal Control for Aerospace Applications

Ilya Kolmanovsky[1]
Dept. of Aerospace Engineering
University of Michigan

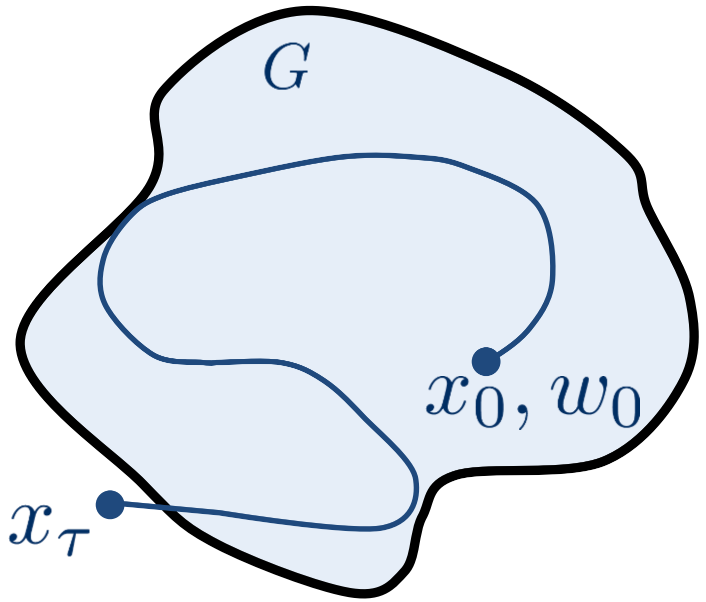


[1]Acknowledgements: Robert Zidek, Alberto Bemporad

Drift counteraction optimal control

- Large disturbances and/or dynamics causing drift
- State constraints
- Control constraints / constrained resources limiting ability to counteract drift
- Eventual constraint violation is inevitable
- **Goal:** Maximize time (or yield) till constraint violation occurs
- No set-points, only constraints!

$$x_{t+1} = f(x_t, u_t, w_t), \;\; u_t \in U$$



$$\tau = \tau(x_0, \{u_t\}, \{w_t\})$$

Deterministic DCOC

Nonlinear discrete-time model: $x_{t+1} = f(x_t, u_t),\ x_0 \in G$

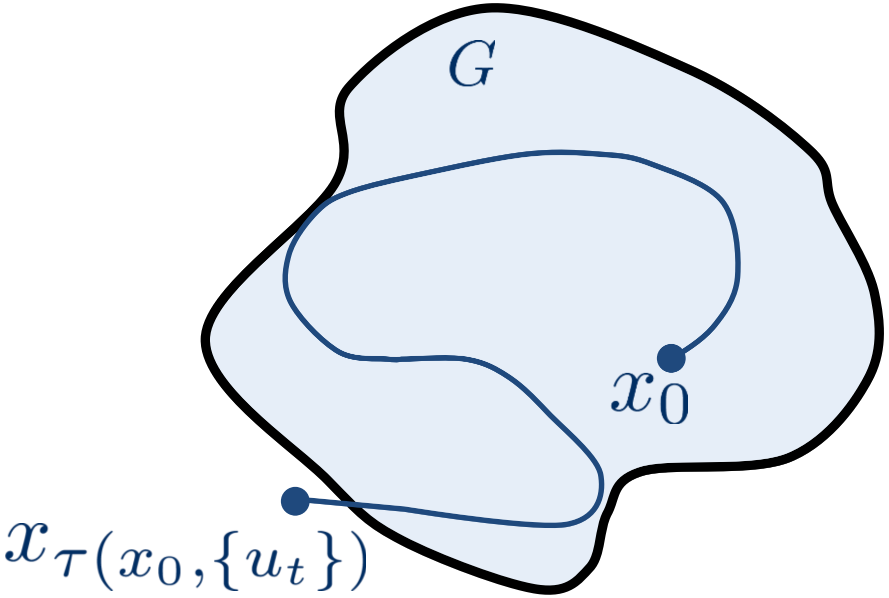
Control input: $u_t \in U$

Admissible sequence of control inputs: $\{u_t\} = \{u_0, u_1, \ldots\} \in U_{\text{seq}}$

Set of state constraints: G

First exit-time: $\tau(x_0, \{u_t\}) = \inf\{t \in \mathbb{Z}_+ : \ x_t \notin G\}$

Instantantaneus reward: $g(x,u) \geq 0$



DCOC problem

Maximize yield before constraint violation:

$$J(x_0, \{u_t\}) = \sum_{t=0}^{\tau(x_0,\{u_t\})-1} g(x_t, u_t) \to \max_{\{u_t\}\in U_{seq}}$$

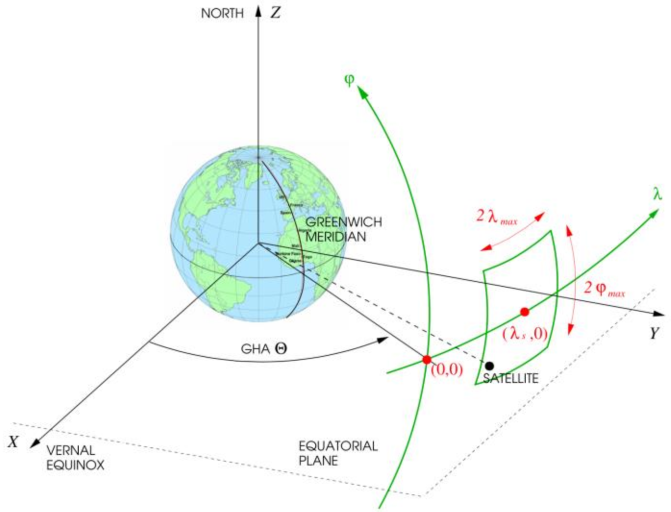
Special case: $g = 1 \ \Rightarrow$ first exit time maximization

GEO satellite station keeping

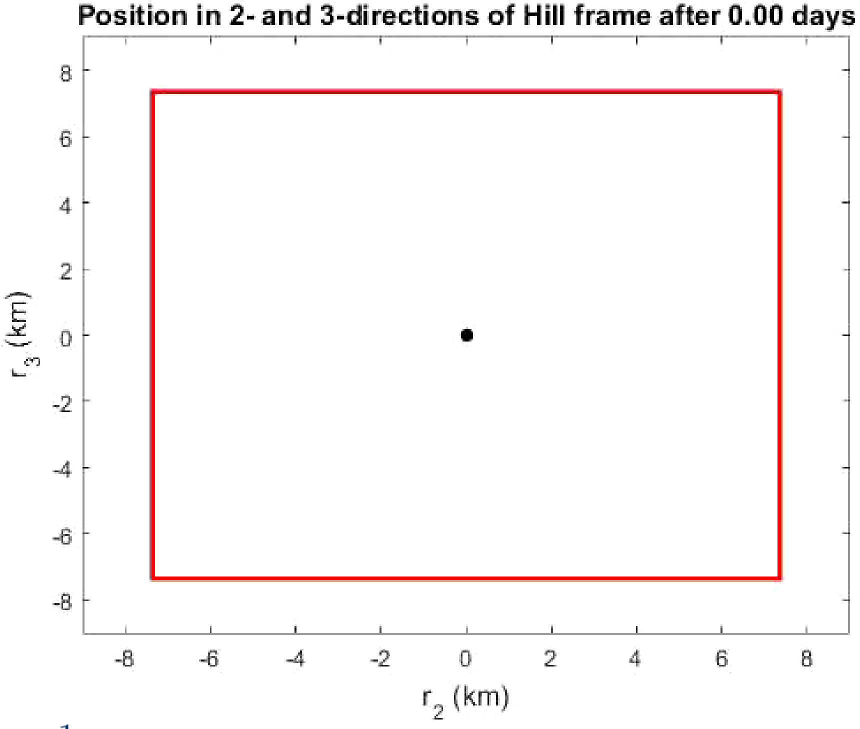
4000 kg satellite, six on/off thrusters (0.2 N each)

Perturbations: luni-solar gravity, solar radiation, $J_2, \cdots$

Nonlinear model with eight states

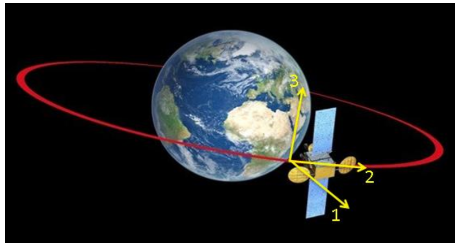


with DCOC



Station keeping requirements:

$$G = \Big\{ (x, m) \in \mathbb{R}^6 \times \mathbb{R}_{\geq 0} : \ |r_1|, |r_2|, |r_3| \leq 7.359 \text{ km}, \\ |v_1|, |v_2| \leq 0.55 \text{ m/sec}, \ |v_3| \leq 0.75 \text{ m/sec}, \\ m \geq m_{min} \Big\}$$



[1] No control ⇒ position constraints violated in less than a day

LEO satellite orbit decay reduction

3U Cube Sat, dry mass: 4 kg

On/off thruster in orbital track direction, 40 g fuel

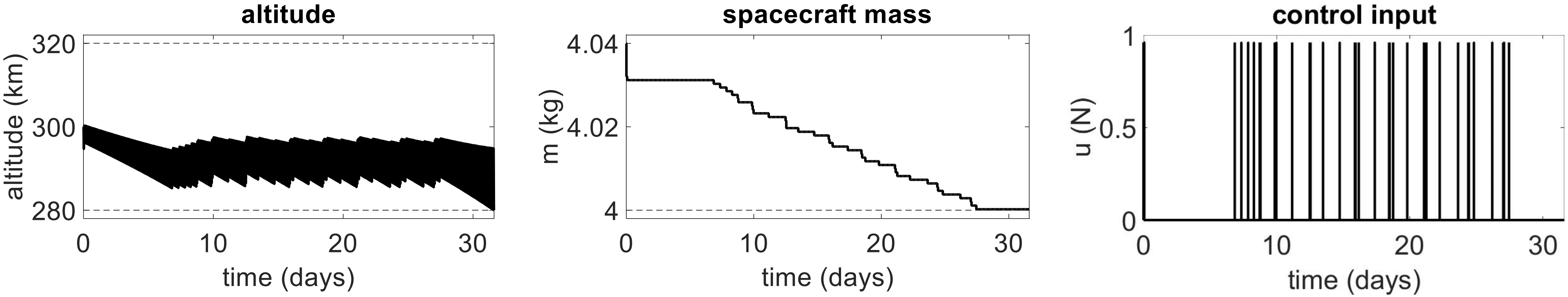
Perturbations: Air drag, J_2

5 state nonlinear model: $x_1 = r, x_2 = \theta, x_3 = \dot{\theta}, x_4 = \dot{\theta}, x_5 = m$

Orbit maintenance requirements:

$$G = \left\{ x \in \mathbb{R}^5 : \; 280 \leq r - R_e \leq 320 \text{ km}, m \geq 4 \text{ kg} \right\}$$

Solution[1]: Drift counteraction optimal control based on Approximate Dynamic Programming and Kriging



[1]No control ⇒ altitude constraints violated in ∼ 3 hours

Stochastic DCOC

Nonlinear discrete-time model: $x_{t+1} = f(x_t, u_t, w_t), \; x_0 \in G$

- Random measured disturbance: w_t
- Control policy: $u_t = \pi(x_t, w_t) \in U$
- First exit time: $\tau(x_0, w_0, \pi) = \inf\{t \in \mathbb{Z}_+ : \; x_t \notin G\}$

Instantaneous reward: $g(x, u) \geq 0$

Maximize expected reward before constraint violation:

$$\mathrm{E}\left\{ \sum_{t=0}^{\tau(x_0,w_0,\pi)-1} g(x_t, u_t) \,\middle|\, x_0, w_0, \pi \right\} \to \max_{\pi \in \Pi}$$

Glider range maximization

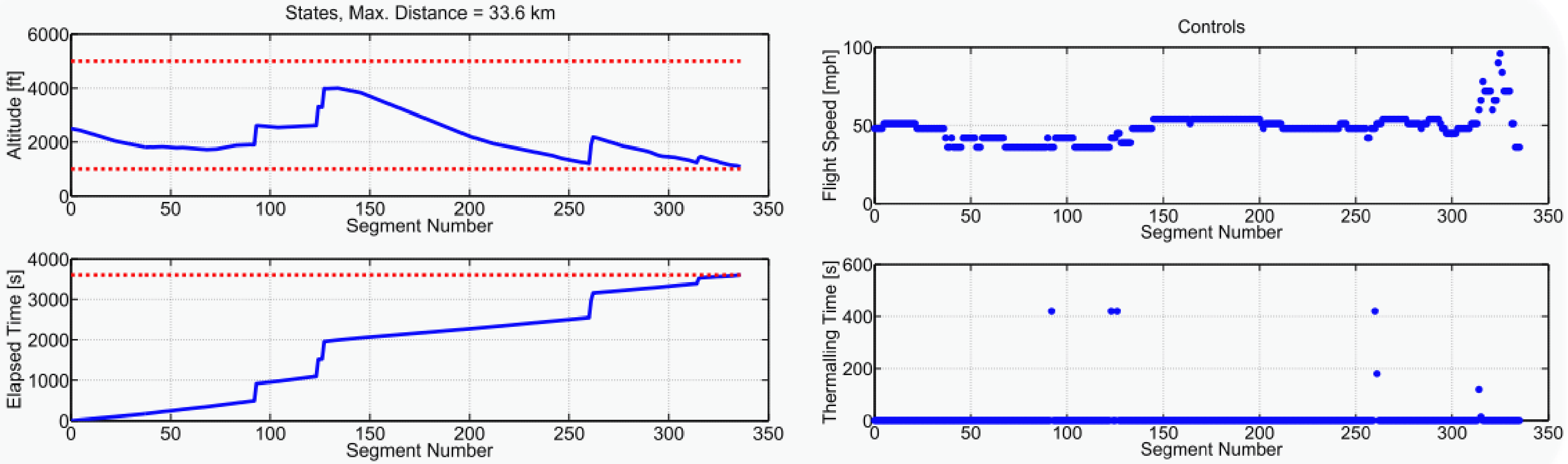
Maximize expected range subject to altitude and flight time constraints

Optimal decision policy for time to spend in a thermal and speed to fly outside thermal

Stochastic model for strengths of thermals and updrafts

Related problem: Maximize surveillance time of a moving ground vehicle

with DCOC



Menezes, Shah and **K.** (2018)

Related work

- There has been much work on exit-time/optimal stopping problems for both stochastic and deterministic continuous-time systems

Lions (1983), Fleming and Soner (2006), Barles and Rouy (1998), Bayraktar et. al. (2010), Gorodetsky et al. (2015), Buchdahn and Nie (2016), Kushner and Dupuis (2013), Barles and Perthane (1998), Blanc (1997), Cannarsa et al. (2000), Munos and Moore (2002), Rungger and Stursberg (2011), **…**

- There has been less work on discrete-time systems
- Much of previous work relied on different assumptions: discounted problems, minimizing non-negative cost, instead of maximizing non-negative cost, etc.
- Our focus is on computational improvements to dynamic programming, MPC formulations, and applications

Deterministic DCOC

Nonlinear discrete-time model: $x_{t+1} = f(x_t, u_t),\ x_0 \in G$

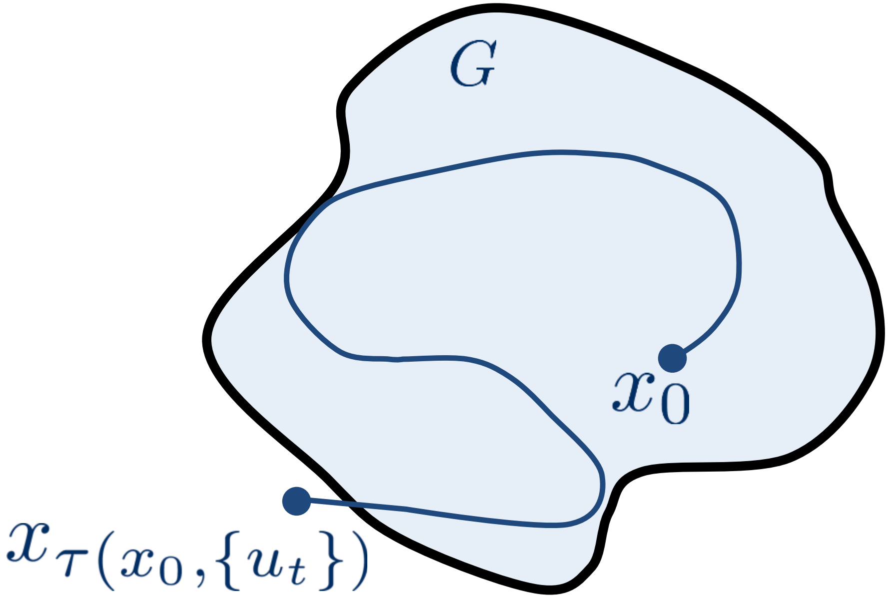
Control input: $u_t \in U$

Admissible sequence of control inputs: $\{u_t\} = \{u_0, u_1, ...\} \in U_{\text{seq}}$

Set of state constraints: G

First exit-time: $\tau(x_0, \{u_t\}) = \inf\{t \in \mathbb{Z}_+ : \ x_t \notin G\}$

Instantantaneus reward: $g(x, u) \geq 0$



DCOC problem

Maximize yield before constraint violation:

$$J(x_0, \{u_t\}) = \sum_{t=0}^{\tau(x_0,\{u_t\})-1} g(x_t, u_t) \rightarrow \max_{\{u_t\}\in U_{seq}}$$

Special case: $g = 1 \ \Rightarrow$ first exit time maximization

Exit time boundness assumption

Assumption 1: The first exit time is bounded from above:

$$\exists \bar{T} \in \mathbb{Z}_{>0} : \; \tau(x_0, \{u_t\}) \leq \bar{T} \text{ for all } x_0 \in G \text{ and } \{u_t\} \in U_{seq}$$

Suppose Assumption 1 holds and $g \equiv 1$ (time maximization problem). Then a solution to the DCOC problem exists.

Suppose Assumption 1 holds, U consists of a finite number of values and $g(x, u)$ is upper bounded on $G \times U$. Then a solution to the DCOC problem exists.

If Assumption 1 holds, G is compact, f is continuous and g is upper semicontinuous, then $\tau(x_0, \{u_t\})$, $J(x_0, \{u_t\})$ are upper semicontinuous w.r.t x_0

Dynamic programming solution

Dynamic Programming

Define $L^{\pi}V(x) = V(x) - V(f(x, \pi(x)))$

Find V and π^* such that

$$L^{\pi^*}V(x) = g(x, \pi^*(x)), \ \text{if } x \in G,$$
$$L^{\pi}V(x) \geq g(x, \pi(x)), \ \text{if } x \in G, \ \pi \neq \pi^*,$$
$$V(x) = 0, \ \text{if } x \notin G,$$

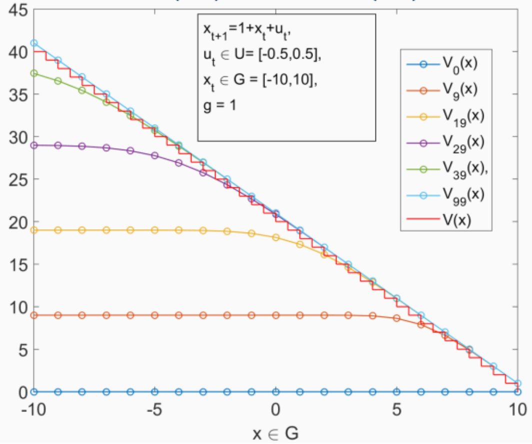
$$\pi^*(x) \in \Pi^*(x) = \arg\max_{u \in U}\{g(x,u) + V(f(x,u))\}$$

Value Iterations

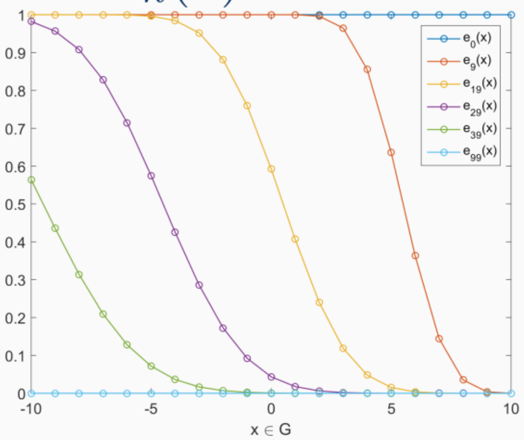
$$V_{n+1}(x) = V_n(x) + k e_n(x), \ \text{if } x \in G,$$
$$V_{n+1}(x) = 0, \ \text{if } x \notin G,$$
$$e_n(x) = \max_{u \in U}\{g(x,u) + V_n(f(x,u))\} - V_n(x)$$

- Convergence for constant $k \in (0, 2)$

$V_n(x) \rightarrow V(x)$



$e_n(x) \rightarrow 0$

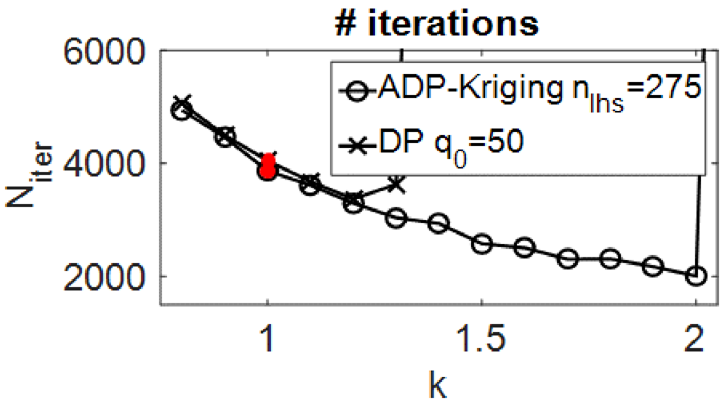


Computations

- In theory, $k = 1$ (conventional VI) maximizes convergence rate
- In a numerical setting, $k = 1$ may not be optimal

DCOC for LEO CubeSat

- $x = (r, v_r, \theta, v_\theta, m) \in \mathbb{R}^5$, $u \in \{0, F_t\} \subset \mathbb{R}^1$
- DP based on grid/mesh of size $q_0 \times q_0 \times 1 \times q_0 \times q_0$
- ADP based on Kriging interpolation of $12 + n_{\text{lhs}}$ points



$k = 1$, DP

q_0	30	50	70
t_τ (days)	2.6	23.7	31.1
t_{comp} (hours)	0.13	1.4	7.85

$k = 1.8$, ADP

n_{lhs}	250	300	350
t_τ (days)	30.9	31	31.4
t_{comp} (hours)	0.02	0.04	0.06

t_τ - exit time, t_{comp} - computing time

Adaptive proportional value iteration

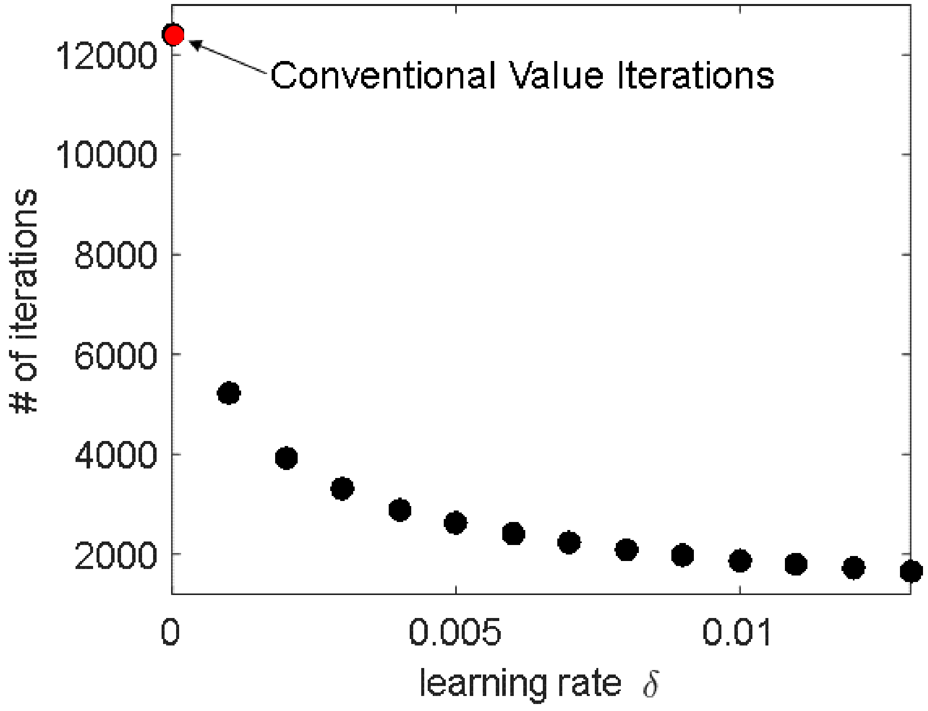
Error:

$$e_n(x) = \max_{u \in U}\{V_n(f(x,u)) + g(x,u)\} - V_n(x)$$

Adaptive proportional value iteration:

$$V_{n+1}(x) = V_n(x) + k_n(x)e_n(x), \ \text{if } x \in G$$
$$k_{n+1}(x) = k_n(x) + \delta e_n(x)$$
$$V_{n+1}(x) = 0, \ \text{if } x \notin G$$

GEO satellite station keeping



Modified value iteration algorithm

Exit time maximization problem, $g = 1$

Specify base control policy (here zero control): $\pi_0(x) = 0$ for all $x \in G$

Base trajectory corresponding to π_0:

$$S_0(x_0) = \left\{x_0, x_1 = f(x_0, 0), x_2 = f(x_1, 0), \cdots, x_{\tau(x_0,\pi_0)-1} = f(x_{\tau(x_0,\pi_0)-2}, 0)\right\}$$

Time instant corresponding to $x' \in S_0(x_0)$:

$$t_{x'}(x_0) = \inf\left\{t \in \mathbb{Z}_+ : x_{t-1} = x' \in S_0(x_0)\right\}$$

Modified value iterations, $V_n \to V$ pointwise

$$V_n(x) = \max_{(x^*,u)\in S_0(x)\times U} \left\{V_{n-1}(f(x^*, u)) + t_{x^*}(x)\right\}, \text{ if } x \in G$$
$$V_n(x) = 0, \text{ if } x \notin G$$
$$(g = 1)$$

More accurate, effect of interpolation errors is reduced.

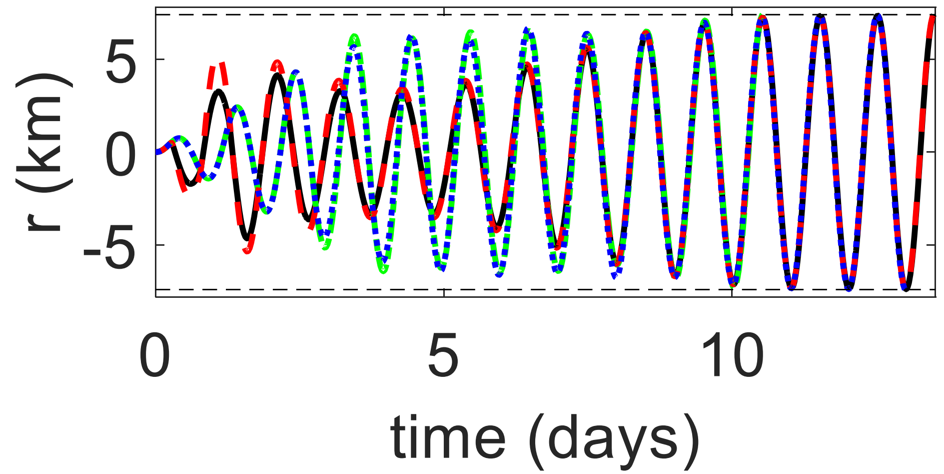
GEO North-South (out of orbital plane) station keeping

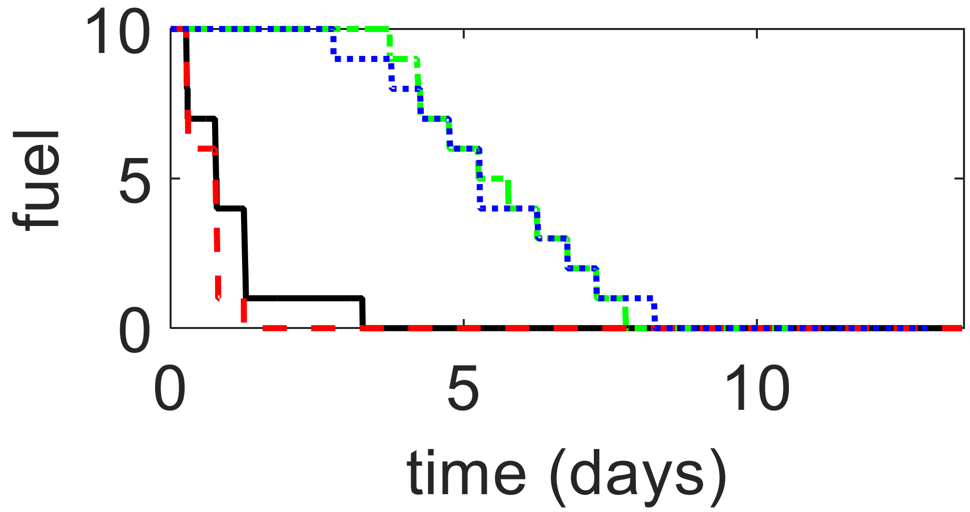
Discretizations $\tilde{G} \subseteq G$:

- Nominal grid: 31 million points
- Dense grid: 1.7 billion points

Case	Comp. time (C, desktop pc)	$\tau(x_0, \tilde{\pi}^*)$
new (nominal)	31 min	116,650
VI (nominal)	16 min	90,967
new (dense)	18.5 hours	116,596
VI (dense)	18 hours	112,356

"new" = modified, "VI" = conventional





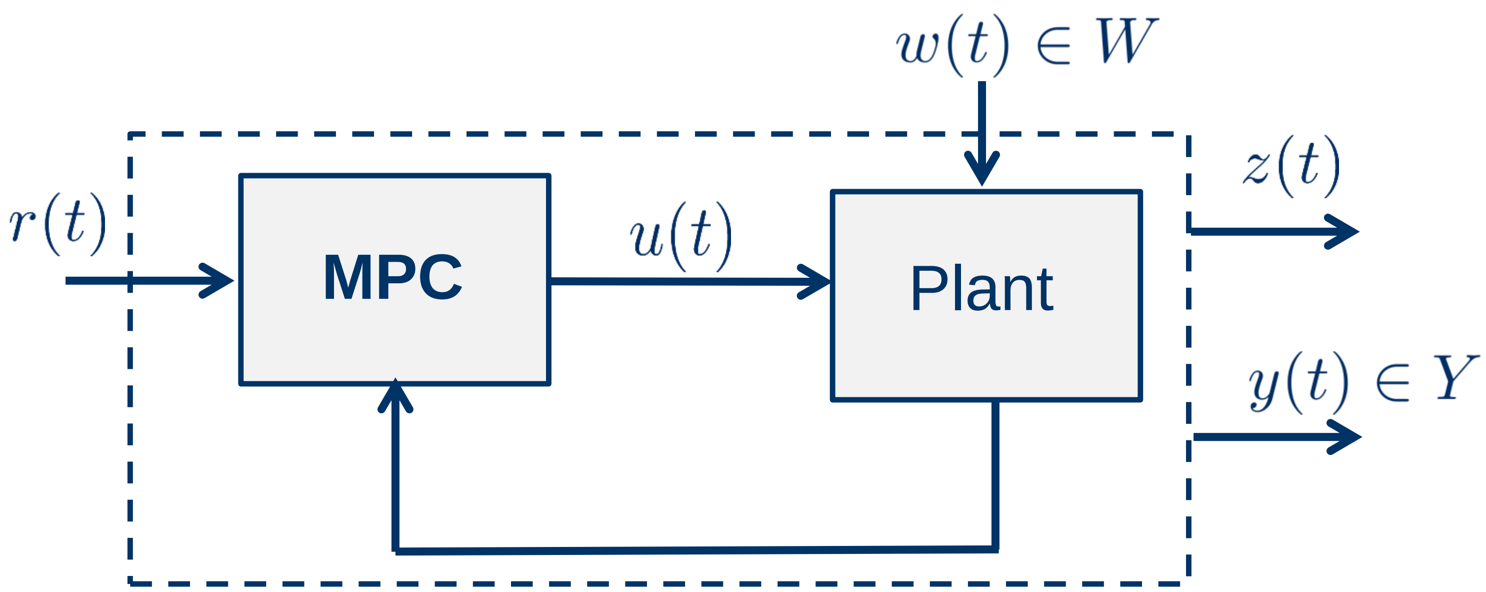
Handling larger dimensional problems

How to handle larger dimensional problems?

Model Predictive Control?

What are suitable MPC formulations?

Model Predictive Control (MPC)



$$F(x_N - x_e(r)) + \sum_{k=0}^{N-1} L(z_k - r, u_k)$$

$$\rightarrow \min_{u_0,\cdots,u_{N-1}}$$

subject to

$$x_{k+1} = f(x_k, u_k),\ \ k = 0, \cdots, N-1$$

$$x_0 = x(t),\ \ r = r(t),$$

$$z_k = g(x_k),$$

$$y_k = \begin{bmatrix} u_k \\ h(x_k) \end{bmatrix} \in Y,$$

$$x_N \in X_f(r) \subset X$$

$$u(t) = u_{MPC}(x(t), r(t)) = u_0^*$$

MPC is a feedback law defined by the first element of the optimal control sequence obtained as a solution to a constrained optimal control problem

Nonlinear and linear programming approaches

Basic setting:

Nonlinear discrete-time model: $x_{t+1} = f_t(x_t, u_t),\ x_0 \in G_0$

Control constraints: $u_t \in U_t$

Maximizing first exit-time: $\tau(x_0, \{u_t\}) = \inf\{t \in \mathbb{Z}_+ :\ x_t \notin G_t\} \to \max$

Assumption: G_t and U_t are polyhedral sets, can depend on time t

$$G_t = \{x \in \mathbb{R}^n : C_{\mathrm{s},t} x \leq b_{\mathrm{s},t}\}$$

Mixed integer nonlinear program (MINLP)

Assumption: $G_t = \{x \in \mathbb{R}^n : C_{\mathrm{s},t} x \leq b_{\mathrm{s},t}\}$

$$\min_{\{u_t\},\{\delta_{\tau_{\mathrm{lb}}},\dots,\delta_N\}} \sum_{t=\tau_{\mathrm{lb}}}^{N} \delta_t \quad \text{s.t.}$$

$$x_{t+1} = f_t(x_t, u_t)$$

$$\delta_{t-1} \leq \delta_t$$

$$\delta_t \in \{0,1\} \subset \mathbb{Z}, \quad t = \tau_{\mathrm{lb}}, \dots, N$$

$$C_{\mathrm{s},t} x_t \leq b_{\mathrm{s},t}, \quad t = 1, \dots, \tau_{\mathrm{lb}} - 1$$

$$C_{\mathrm{s},t} x_t \leq b_{\mathrm{s},t} + \mathbf{1} M \delta_t, \quad t = \tau_{\mathrm{lb}}, \dots, N$$

$$u_t \in U_t$$

N: time horizon
M: large scalar
δ_t: indicator variable for $x_t \notin G_t$
τ_{lb}: lower bound on optimal first exit-time

Theorem: The MINLP gives solution with the same exit time as the solution to the DCOC problem (if exists), if M and N are sufficiently large.

NLP formulation

Approximate solution: replace binary variables δ_t with real-valued variables ε_t
$\rightarrow$ **NLP** without integer variables (suboptimal solution)

$$\min_{\{u_t\},\{\varepsilon_{\tau_{\text{lb}}},\dots,\varepsilon_N\}} \sum_{t=\tau_{\text{lb}}}^{N} \varepsilon_t \quad \text{s.t.}$$

$$x_{t+1} = f_t(x_t, u_t)$$

$$0 \le \varepsilon_{t-1} \le \varepsilon_t$$

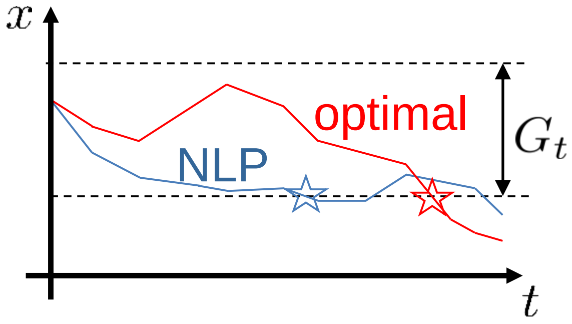
$$C_{\text{s},t} x_t \le b_{\text{s},t}, \quad t = 1, \dots, \tau_{\text{lb}} - 1$$

$$C_{\text{s},t} x_t \le b_{\text{s},t} + \mathbf{1}\varepsilon_t, \quad t = \tau_{\text{lb}}, \dots, N$$

$$u_t \in U_t$$

N: time horizon
ε_t: indicator variable for $x_t \notin G_t$
τ_{lb}: lower bound on optimal first exit-time

NLP solution is close to optimal solution (MINLP solution) if N and τ_{lb} are close to optimal first exit-time[1].



[1] Weighted versions may recover an exact solution

Reduction to LP

- Linear model approximation: $x_{t+1} = A_t x_t + B_t u_t + d_t$
- MILP (similar to MINLP) solves linear DCOC problem
- Approximate solution with **LP**

$$\min_{\{u_t\},\{\varepsilon_{\tau_{\text{lb}}},\dots,\varepsilon_N\}} \sum_{t=\tau_{\text{lb}}}^{N} \varepsilon_t \quad \text{s.t.}$$

$$x_{t+1} = A_t x_t + B_t u_t + d_t$$

$$0 \le \varepsilon_{t-1} \le \varepsilon_t$$

$$C_{\text{s},t} x_t \le b_{\text{s},t}, \quad t = 1, \dots, \tau_{\text{lb}} - 1$$

$$C_{\text{s},t} x_t \le b_{\text{s},t} + \mathbf{1}\varepsilon_t, \quad t = \tau_{\text{lb}}, \dots, N$$

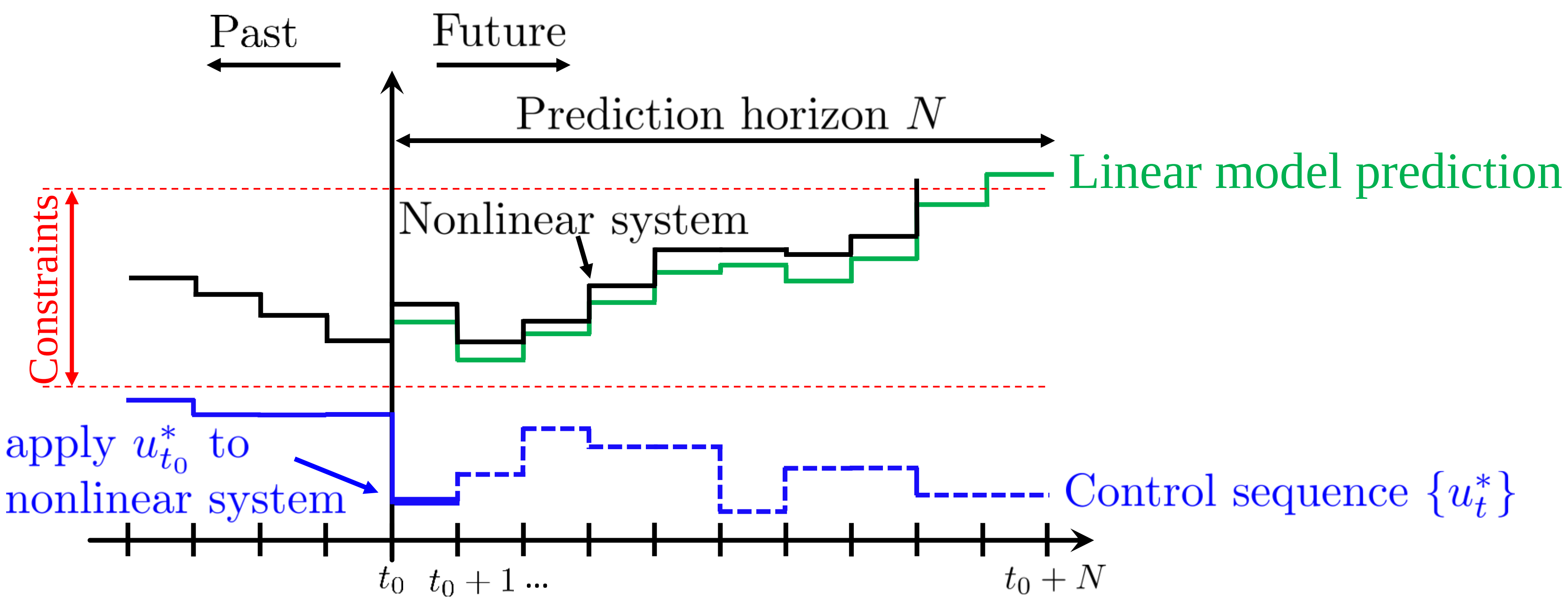
$$u_t \in U_t$$

Iterative Procedure: Increase horizon N until constraint violation

1: $\tau_{\text{lb}} \leftarrow$ Set initial lower bound
2: $N \leftarrow \tau_{\text{lb}} + N_{\text{add}}, \; N_{\text{add}} \in \mathbb{Z}_+$
3: $\{u_t\}, \{\varepsilon_{\tau_{\text{lb}}}, \dots, \varepsilon_N\} \leftarrow$ solution of LP
4: $\tau \leftarrow \max\{t \le N : \varepsilon_t = 0\} + 1$
5: **if** $\varepsilon_N = 0$ **then**
6: $\quad \tau_{\text{lb}} \leftarrow \tau$; go to Step 2
7: **end if**

MPC strategy

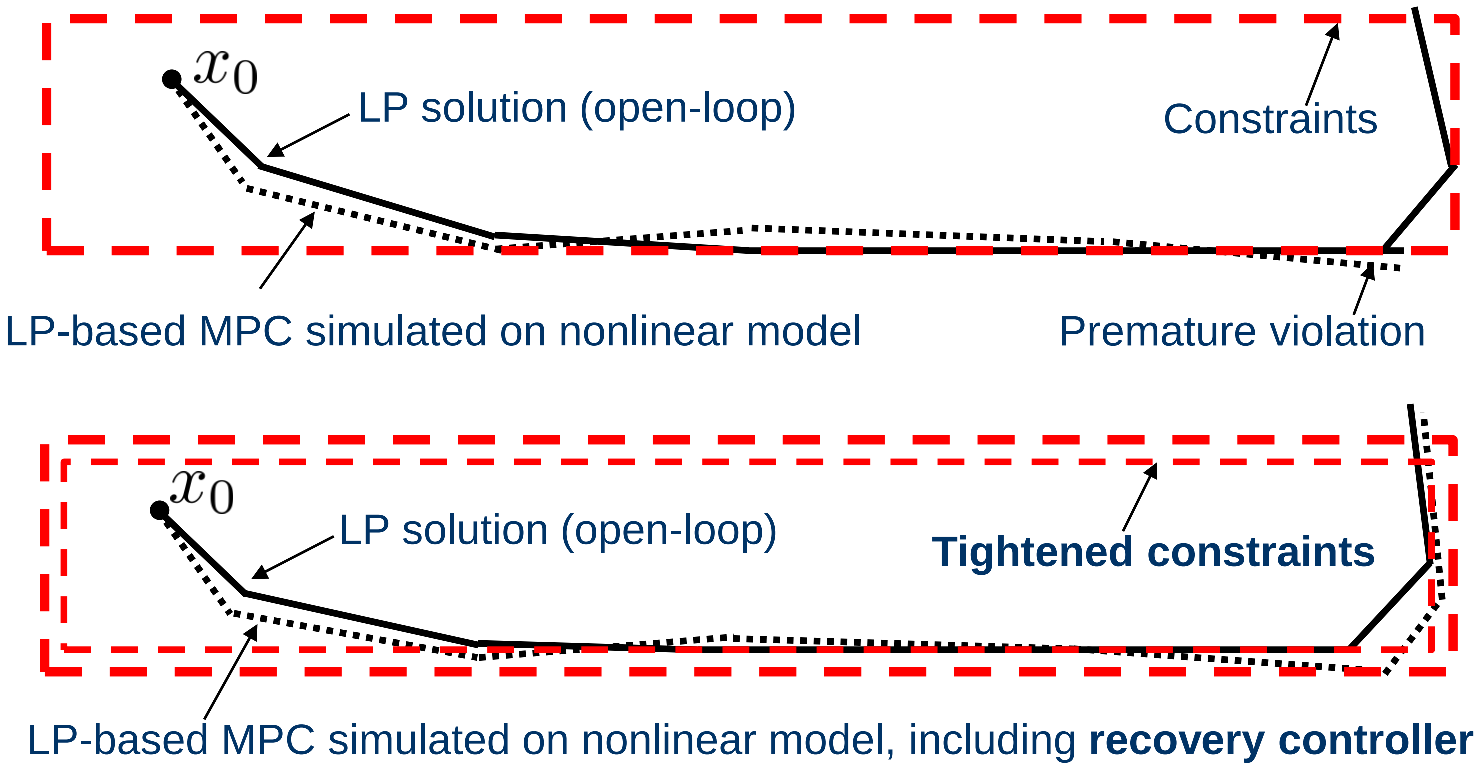
At each $t_0 \in \mathbb{Z}_{\geq 0}$, compute linear DCOC solution $\{u_t^*\} = \{u_{t_0}^*, u_{t_0+1}^*, \dots\}$ based on x_{t_0} with iterative procedure and apply $u_{t_0}^*$ to system.



Recovery control

Avoid premature constraint violation due to unmodelled effects

- Tighten constraints for LP solution
- Employ recovery controller (when tightened constraints are violated), also LP-based



Recovery controller LP

$$\min_{\{x_t\},\{u_t\},\{\varepsilon_t\}} \sum_{t=1}^{N_{recover}} \varepsilon_t$$

subject to

$$x_{t+1} = A_t x_t + B_t u_t + d_t, \quad x_0 \notin \tilde{G}_0$$

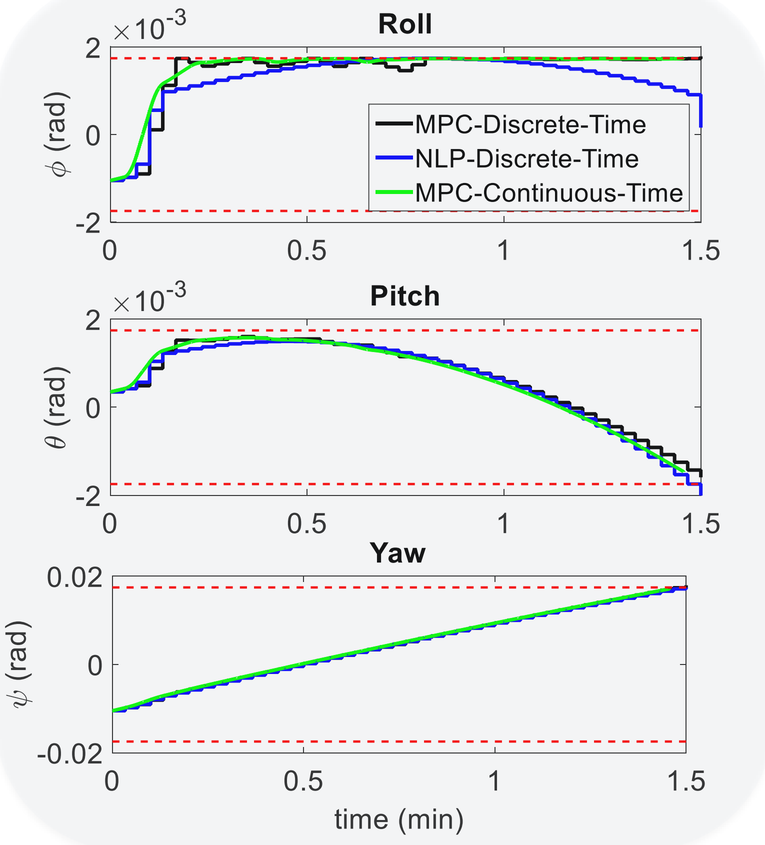
$$0 \leq \varepsilon_t$$

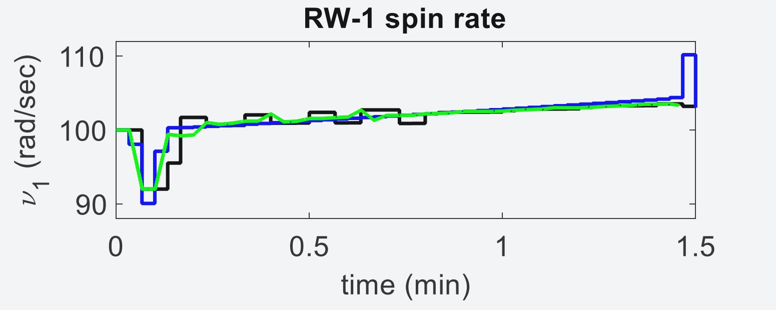
$$\tilde{C}_{s,t} x_t \leq b_{s,t} + \mathbf{1}\varepsilon_t$$

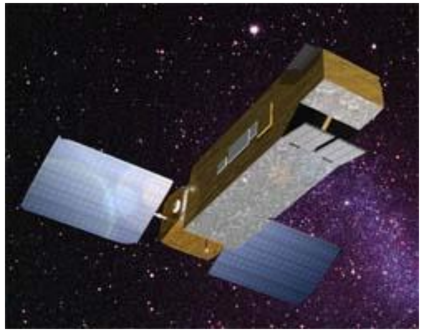
$$u_t \in U_t$$

Spacecraft pointing control with a single reaction wheel

- Nonlinear model, 7 states, 1 control. Reaction wheel spin axis $\bar{g}_1 = \frac{1}{\sqrt{3}}(1,\ 1,\ 1)^{\mathrm{T}}$.
- Attitude drift due to nonzero angular momentum and solar radiation pressure
- Linear model approximation (for MPC). Sampling time $\Delta t = 2$ sec







Method – Sim. Model	**Exit-Time** (min)	**Computation Time**, MATLAB (sec)
MPC - Discrete-Time	1.5	0.01 (average), 0.08 (worst-case)
NLP - Discrete-Time	1.5	50
MPC - Continuous-Time	1.46	0.01 (average), 0.08 (worst-case)

Stochastic DCOC based on linear models

Linear discrete-time model: $x_{t+1} = A_t x_t + B_t u_t + w_t, \; x_0 \in G_0$

Random disturbance $w_t \in W = \{w^i : i \in I\}$, modeled by a Markov Chain with a finite number of states and transition probabilities $\mathbb{P}(w_{t+1} = w^j | w_t = w^i) \in [0, 1]$

Control policy: $u_t = \pi(x_t, w_t, t) \in U_t$

First exit-time: $\tau(x_0, w_0, \pi) = \inf\{t \in \mathbb{Z}_+ : x_t \notin G_t\}$ (random, depends on $\{w_t\}$)

Expected first exit-time: $\bar{\tau}(x_0, w_0, \pi) = \mathbb{E}\{\tau(x_0, w_0, \pi)\}$

DCOC problem: $\boxed{\max_{\pi \in \Pi} \bar{\tau}(x_0, w_0, \pi)}$ (Π is the set of admissible control policies)

Solution existence

Assumption 1: There exists $\bar{T} > 0$ and $\bar{w} \in W$ s.t.

- $\bar{w}$ overpowers any control policy $\pi \in \Pi$, i.e.,
  $\boldsymbol{x_{t+1} = A_t x_t + B_t \pi(x_t, \bar{w}, t) + \bar{w}}$ violates constraints in at most $\bar{T}$ steps
- $P_W(\bar{w}|\bar{w}) > 0$ and $\bar{w}$ is accessible from each $w \in W$.

Theorem: If Assumption 1 holds, $\bar{\tau}(x, w, \pi)$ is bounded from above.

Theorem: If Assumption 1 holds and U_t is finite for all $t \in \mathbb{Z}_{\geq 0}$, a solution to the DCOC problem exists.

DP solution

Define $f(x,u,w) = Ax + Bu + w$ (time-invariant case)

$$\bar{V}^+(x,w,u) = \sum_{w^i \in W} \left[V(f(x,u,w), w^i)\mathbb{P}(w^i|w) \right], \; L^\pi V(x,w) = V(x,w) - \bar{V}^+(x,w,\pi(x,w))$$

Dynamic Programming

Find V and π^* such that

$$\begin{aligned} &L^{\pi^*} V(x,w) = 1, \text{ if } x \in G, \\ &L^\pi V(x,w) \geq 1, \text{ if } x \in G, \; \pi \neq \pi^*, \\ &V(x,w) = 0, \text{ if } x \notin G, \end{aligned}$$

$$\pi^*(x,w) \in \Pi^*(x,w) = \arg\max_{u \in U} \{\bar{V}^+(x,u,w)\}$$

Value Iterations

$$\begin{aligned} &V_{n+1}(x,w) = V_n(x,w) + k e_n(x,w), \text{ if } x \in G, \\ &V_{n+1}(x,w) = 0, \text{ if } x \notin G, \\ &e_n(x,w) = \max_{u \in U} \left\{ 1 + \bar{V}_n^+(x,w,u) \right\} - V_n(x,w) \end{aligned}$$

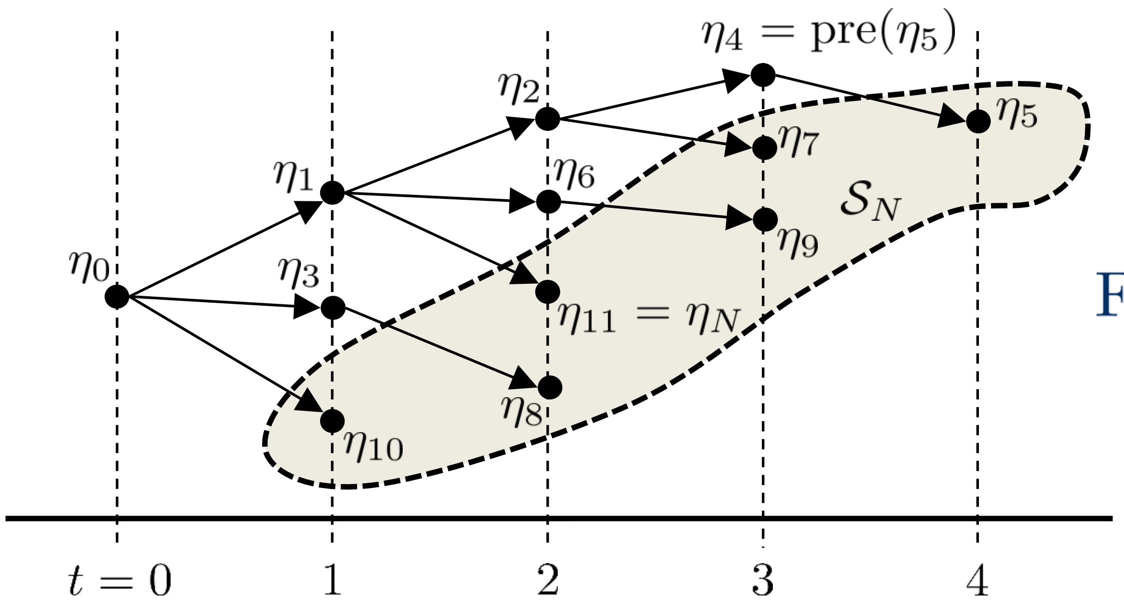
- Convergence for constant $k \in (0,2)$
- Adaptive proportional value iterations - faster
- Adaptive proportional value iterations with damping
- Approximate DP (Neural networks, Kriging)
- Higher order problems → **SMPC**

Scenario tree

Scenario tree $\mathcal{T}_N = \{\eta_0, \eta_1, ..., \eta_N\}$ encodes $|\mathcal{S}_N| \geq 1$ disturbance scenarios

$$\{w_t\}^\eta = \{w_t : t \in \mathbb{Z}_{[t_0, t^\eta]}\}^\eta = (w^{\eta_0}, ..., w^{\mathrm{pre}(\mathrm{pre}(\eta))}, w^{\mathrm{pre}(\eta)}, w^\eta),$$

for each leaf node $\eta \in \mathcal{S}_N$.



For example, $\{w_t\}^{\eta_9} = (w^{\eta_0}, w^{\eta_1}, w^{\eta_6}, w^{\eta_9})$

Scenario tree

- With each $\eta \in \mathcal{T}_N$, associate x^η, w^η, and t^η, where $x^{\eta_0} = x_0$, $w^{\eta_0} = w_0$, and $t^{\eta_0} = 0$ for root node.

- For each node $\eta \in \mathcal{T}_N$, assign a control u^η.

- Satisfy dynamics: $x^\eta = A_{t^{\mathrm{pre}(\eta)}} x^{\mathrm{pre}(\eta)} + B_{t^{\mathrm{pre}(\eta)}} u^{\mathrm{pre}(\eta)} + w^{\mathrm{pre}(\eta)}$.

- Probability of reaching node $\eta \in \mathcal{T}_N$, starting from root node η_0: $\rho^\eta = \rho^{\mathrm{pre}(\eta)} P_W(w^\eta | w^{\mathrm{pre}(\eta)}) \in [0, 1]$, where $\rho^{\eta_0} = 1$.

- Control inputs of tree $\mathcal{T}_N$: $\mathcal{U}_N = \{u^\eta \in U_{t^\eta} : \eta \in \mathcal{T}_N \setminus \mathcal{S}_N\}$

- Control policy defined by given $\mathcal{U}_N$: $\pi_{\mathcal{U}_N}$

Scenario tree generation

$\mathcal{T}_N \leftarrow \{\eta_0\};\ \mathcal{C} \leftarrow \emptyset;\ \rho^{\eta_0} \leftarrow 1$
$t^{\eta_0} \leftarrow 0;\ x^{\eta_0} \leftarrow x_0;\ w^{\eta_0} \leftarrow w_0$
$i \leftarrow 0$
while $i < N$ **do**
  for $j \in \{1, 2, ..., |W|\}$ **do**
    $w^{\eta_j^{\mathrm{succ}(\eta_i)}} \leftarrow w^j\ (w^j \in W)$
    $t^{\eta_j^{\mathrm{succ}(\eta_i)}} \leftarrow t^{\eta_i} + 1$
    $\rho^{\eta_j^{\mathrm{succ}(\eta_i)}} \leftarrow \rho^{\eta_i} P_W(w^j | w^{\eta_i})$
  end for
  $\mathcal{C} \leftarrow \mathcal{C} \cup \mathrm{succ}(\eta_i)$
  $\eta_{i+1} \leftarrow \mathrm{argmax}_{\eta \in \mathcal{C}}\, \rho^{\eta}$ (pick any maximizer)
  $\mathcal{T}_N \leftarrow \mathcal{T}_N \cup \{\eta_{i+1}\}$
  $\mathcal{C} \leftarrow \mathcal{C} \setminus \{\eta_{i+1}\}$
  $i \leftarrow i + 1$
end while

Generate most likely scenarios for a specified number of tree nodes N

Scenario tree – based optimization

- First exit-time corresponding to $\{w_t\}^\eta$, $\eta \in \mathcal{S}_N$:
  $\tau_N^\eta(x, \pi_N) = \min\{\min\{t \in \mathbb{Z}_{[0,t^\eta]} : x_t \notin G_t\} \cup \{t^\eta + 1\}\}$.

- Average first exit-time for $\mathcal{T}_N$ and control policy $\pi_N \in \Pi$:
  $\bar{\tau}_N(x, \pi_N) = \sum_{\eta \in \mathcal{S}_N} \tau_N^\eta(x, \pi_N)\rho^\eta$.

- Maximize average first exit-time of tree $\mathcal{T}_N$: $\boxed{\max_{\pi_N \in \Pi} \bar{\tau}_N(x, \pi_N)}$ *

Theorem: The solution to problem* is arbitrarily close (in terms of $\bar{\tau}$ performance) to the DCOC solution (if exists) for sufficiently large N.

MILP formulation

Assumption: G_t and U_t are polyhedral sets and $G_t = \{x \in \mathbb{R}^n : C_t x \le b_t\}$ for all $t \in \mathbb{Z}_{\ge 0}$.

$$\min_{\mathcal{U}_N, \{\delta^{\eta_0}, \dots, \delta^{\eta_N}\}} \sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_N^\eta} \delta^\eta \rho^\xi \quad \text{s.t.}$$

$$x^\eta = A_{t^{\text{pre}(\eta)}} x^{\text{pre}(\eta)} + B_{t^{\text{pre}(\eta)}} u^{\text{pre}(\eta)} + w^{\text{pre}(\eta)}, \text{ for all } \eta \in \mathcal{T}_N \setminus \{\eta_0\}$$

$$\delta^\eta \ge \delta^{\text{pre}(\eta)}, \text{ for all } \eta \in \mathcal{T}_N \setminus \{\eta_0\}$$

$$\delta^\eta \in \{0, 1\} \subset \mathbb{Z}, \text{ for all } \eta \in \mathcal{T}_N$$

$$C_{t^\eta} x^\eta \le b_{t^\eta} + \mathbf{1} M \delta^\eta, \text{ for all } \eta \in \mathcal{T}_N,$$

Theorem: If M is sufficiently large, the control policy $\pi_{\mathcal{U}_N^*}$ defined by the MILP solution $\mathcal{U}_N^*$ maximizes the average first exit-time of tree $\mathcal{T}_N$.

[1] $\mathcal{K}_N^\eta$ = set of leaf nodes whose associated disturbance scenarios contain node $\eta \in \mathcal{T}_N$

MILP formulation

Theorem: If M is sufficiently large, the control policy $\pi_{\mathcal{U}_N^*}$ defined by the MILP solution $\mathcal{U}_N^*$ is arbitrarily close (in terms of $\bar{\tau}$ performance) to a solution of the DCOC problem (if exists) for sufficiently large N.

Stochastic MPC (SMPC) implementation

Feedback to compensate for incomplete trees and/or unmodeled effects

- At each time instant $t_0 \in \mathbb{Z}_{\geq 0}$:
  - generate tree $\mathcal{T}_N$ based on x_{t_0} and w_{t_0}
  - compute MILP solution $\mathcal{U}_N^*$ for $\mathcal{T}_N$
- SMPC policy: $\pi_{\mathrm{SMPC},N}(x_{t_0}, w_{t_0}, t_0) = u^{n_0} \in \mathcal{U}_N^*$

Theorem: If M is sufficiently large, the SMPC policy $\pi_{\mathrm{SMPC},N}$ is arbitrarily close (in terms of $\bar{\tau}$ performance) to a solution of the DCOC problem (if exists) for sufficiently large N.

SMPC implementation

1: $t \leftarrow 0$
2: $x \leftarrow$ obtain current $x(t)$
3: $w \leftarrow$ obtain current $w(t)$
4: $\mathcal{T}_N \leftarrow$ generate tree based on $t^{\eta_0} \leftarrow t$, $x^{\eta_0} \leftarrow x$, and $w^{\eta_0} \leftarrow w$
5: $t_{\text{comp}} \leftarrow 0$
6: **while** computing solution of MILP **do**
7:     **if** $t_{\text{comp}} > t_{\max}$ **then**
8:         go to Step 13
9:     **end if**
10:     $t_{\text{comp}} \leftarrow$ update t_{comp}
11: **end while**
12:     $\mathcal{U}_N^* \leftarrow$ solution of MILP; go to Step 14
13:     $\mathcal{U}_N^* \leftarrow$ solution of LP without integer variables
14:     $u(t) \leftarrow$ apply control $u^{\eta_0} \in \mathcal{U}_N^*$ to the system
15:     $t \leftarrow t + 1$
16:     go to Step 2

Second order example with disturbance

$$x_{t+1} = A_t x_t + B_t u_t + w_t$$

$$x = [r_1,\ r_2]^{\mathrm{T}}$$

$$A_t = \begin{bmatrix} 1 & 0.1 \\ -0.1 & 1.2 \end{bmatrix}$$

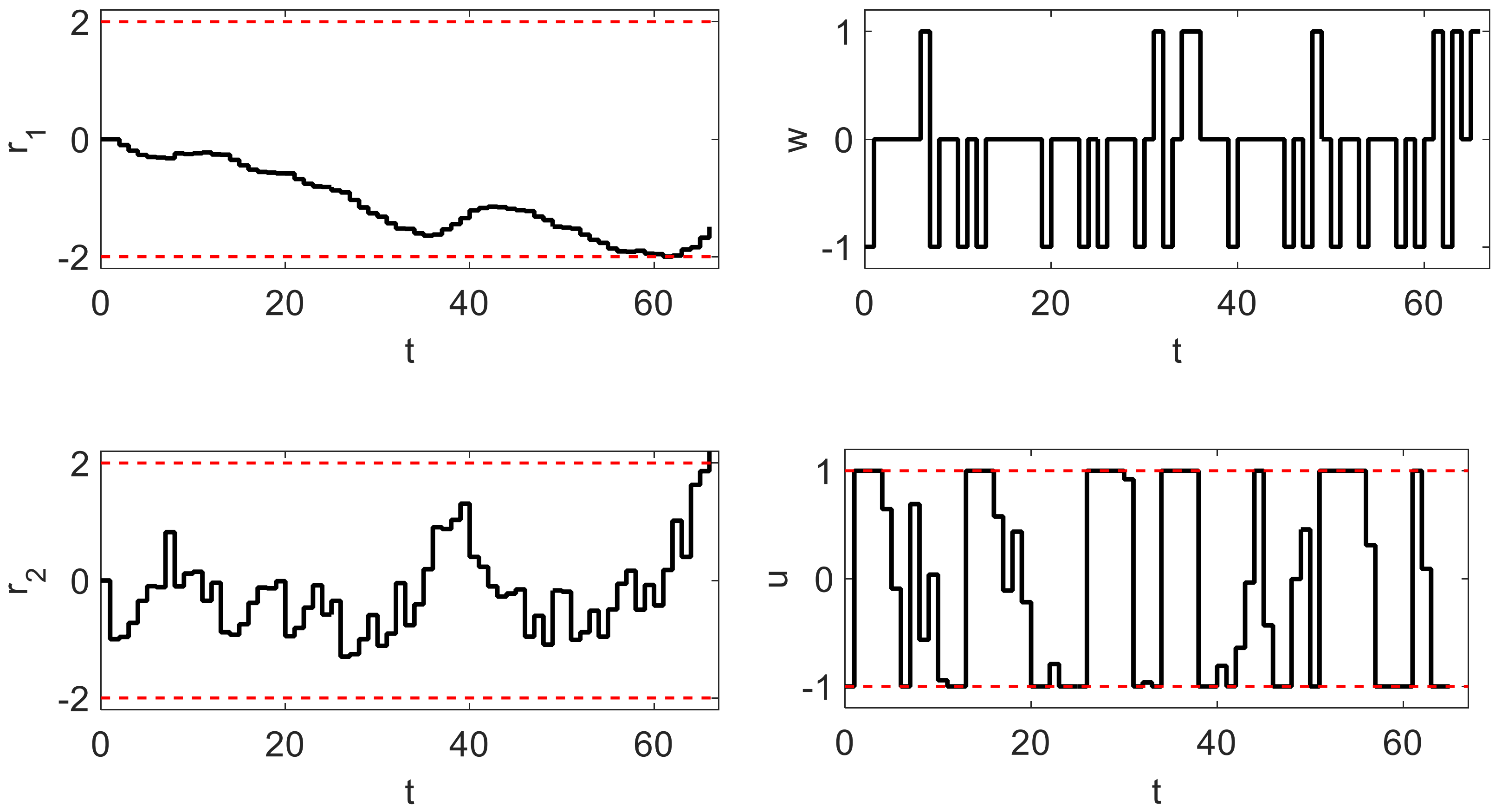
$$B_t = \begin{bmatrix} 0 \\ 0.5\sin(t/2) \end{bmatrix}$$

$$w_t \in \{-1, 0, 1\}$$

$$P_W = \begin{bmatrix} 0 & 0.8 & 0.2 \\ 0.3 & 0.5 & 0.2 \\ 0.35 & 0.4 & 0.25 \end{bmatrix}$$

$$G = [-2,\ 2] \times [-2,\ 2]$$

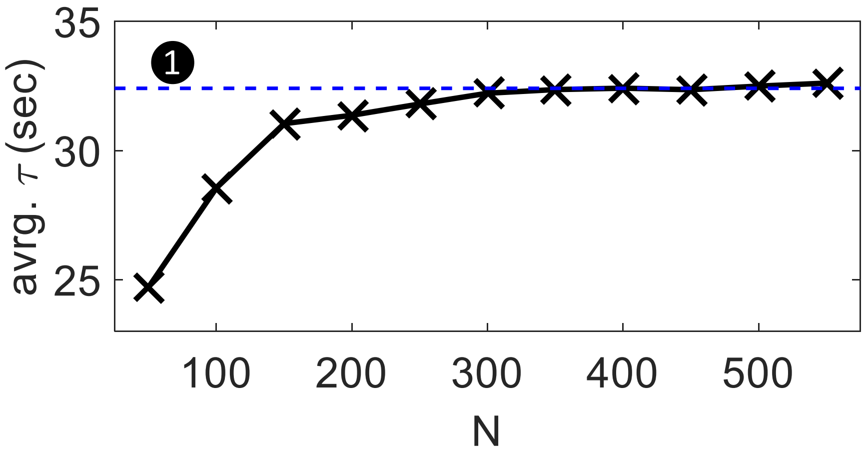
$$u_t \in [-1, 1] \quad x_0 = [0, 0]^{\mathrm{T}},\ w_0 = -1$$



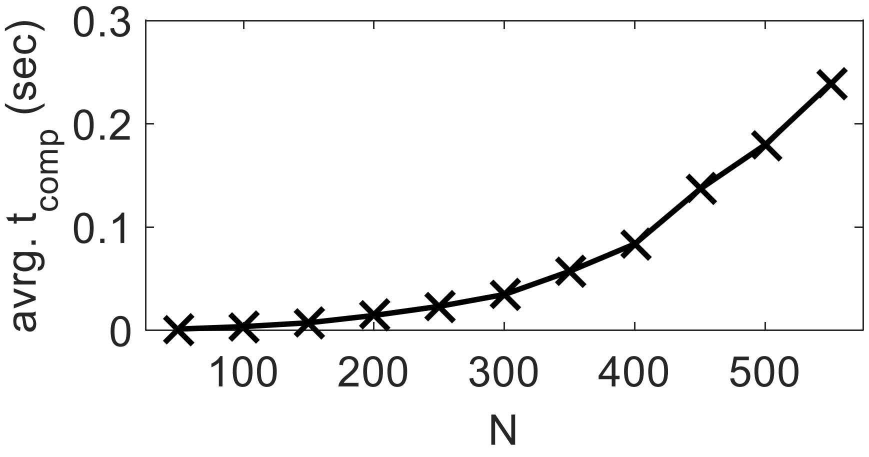
Second order example with disturbance

- Results for 1,000 random trajectories (samples)
- Hybrid Toolbox in MATLAB 2015a on Laptop
- Computation time limit for MILP: $t_{\max} = 10 \text{ sec}$

Average exit time vs number of tree nodes



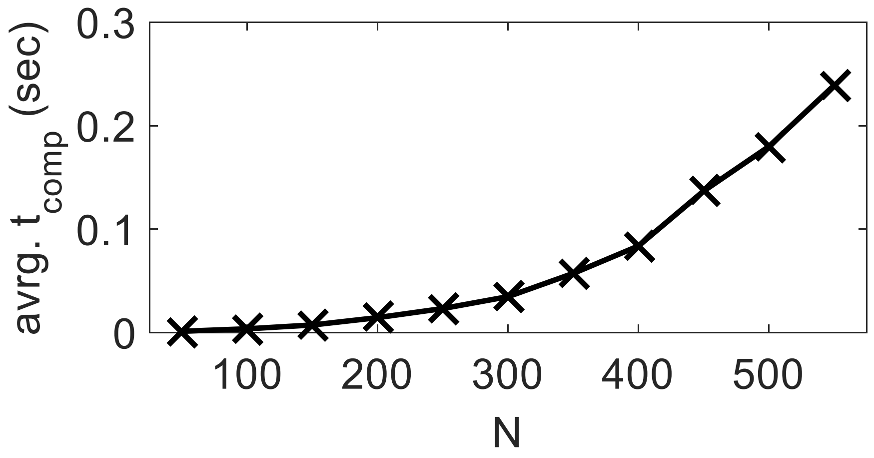
Average computation time (per single time instant) vs number of tree nodes



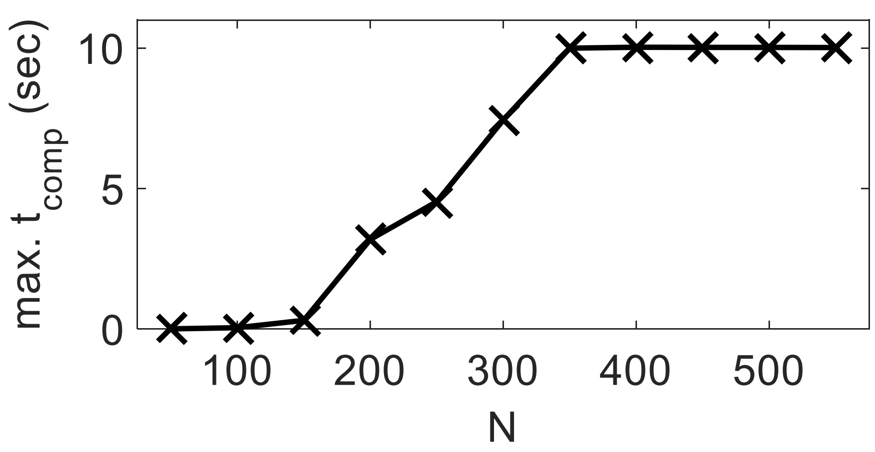
❶ DP solution for 60x60x250 grid: 32.41 sec, offline computation time: 1.63 hours (in C)

Second order example with disturbance

Average computation time (per single time instant) vs number of tree nodes



Worst-case computation time (per single time instant) vs number of tree nodes



Minimum-time MPC problem

Discrete-time model

$$x_{k+1} = f(x_k, u_k) = Ax_k + Bu_k + d,$$

Control constraints:

$$u_k \in U = \{u \in \mathbb{R}^{n_u} :\ \Gamma u \leq \gamma,\ \gamma \in \mathbb{R}^{n_\gamma}\},$$

Target set:

$$C = \{x \in \mathbb{R}^{n_x} :\ Hx \leq h,\ h \in \mathbb{R}^{n_h}\},$$

Time to reach the target set:

$$\tau(x_0, \{u_k\}) = \inf\{k :\ x_k = \phi_{\{u_k\}}(k, x_0) \in C,\ k \in \mathbb{Z}_{\geq 0}\}, \quad \phi \sim \text{solution map}$$

Minimum-time to reach target set:

$$\min_{\{u_k\} \subset U} \tau(x_0, \{u_k\}).$$

MILP reformulation of minimum-time MPC

Simplifying assumption: The terminal set is control-invariant

Mixed Integer Linear Programming problem:

$$\sum_{k=\tau_{\mathrm{lb}}(x_0)}^{N} \delta_k \to \min_{\{\delta_k\},\{u_k\}}$$

subject to

$$\begin{gathered}
x_{k+1} = Ax_k + Bu_k + d,\ k = 0, \cdots, N-1 \\
Hx_k \le h + M\mathbf{1}_{n_h}\delta_k,\ k = \tau_{\mathrm{lb}}(x_0), \cdots, N, \\
\delta_k \in \{0,1\},\ k = \tau_{\mathrm{lb}}(x_0), \cdots, N, \\
\Gamma u_k \le \gamma,\ k = 0, 1, \cdots, N-1, \\
\delta_{k+1} \le \delta_k,\ k = \tau_{\mathrm{lb}}(x_0), \cdots, N-1
\end{gathered}$$

MILP reformulation of minimum-time MPC problem

- The matrices H, h define the polyhedral target set C, M is a large parameter, $\mathbf{1}_{n_h}$, is a vector of ones, and δ_k is a binary decision variable used to relax the inequality constraint.
    - If $\delta_k = 0$, then the state lies inside the target set, C.
    - If the state falls outside the target set, then $\delta_k = 1$.
- The additional constraint $\delta_{k+1} \leq \delta_k$ ensures that once C is reached, the trajectory remains there for all future times.
- The control constraints are $\Gamma u_k \leq \gamma$
- $\tau_{lb}(x_0)$ is a lower bound on the time-to-go
- The minimum-time problem reduces to a Mixed-Integer Linear Program (MILP) and can be solved using standard numerical algorithms. Matlab: use intlinprog.m

MILP not requiring target set control invariance

$$\sum_{k=\tau_{\mathrm{lb}}(x_0)}^{N} \epsilon_k \to \min_{\{\epsilon_k\},\{\delta_k\},\{u_k\}}$$

subject to

$$\begin{gathered}
x_{k+1} = Ax_k + Bu_k + d,\ k = 0, \cdots, N-1 \\
Hx_k \le h + M\mathbf{1}_{n_h}(\delta_k + \epsilon_k),\ k = \tau_{\mathrm{lb}}(x_0), \cdots, N, \\
\delta_k \in \{0,1\},\ k = \tau_{\mathrm{lb}}(x_0), \cdots, N, \\
\epsilon_k \in \{0,1\},\ k = \tau_{\mathrm{lb}}(x_0), \cdots, N, \\
\delta_{k+1} + \epsilon_k = 1,\ k = \tau_{\mathrm{lb}}(x_0), \cdots, N, \\
\delta_{\tau_{lb}(x_0)} = 0, \\
\Gamma u_k \le \gamma,\ k = 0, 1, \cdots, N-1, \\
\delta_{k+1} \le \delta_k,\ k = \tau_{\mathrm{lb}}(x_0), \cdots, N-1
\end{gathered}$$

Flexible spacecraft EOMs

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{c(\theta)} \begin{bmatrix} c(\theta) & s(\phi)s(\theta) & c(\phi)s(\theta) \\ 0 & c(\phi)c(\theta) & -s(\phi)c(\theta) \\ 0 & s(\phi) & c(\phi) \end{bmatrix} \times \left(\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} + n \begin{bmatrix} c(\theta)s(\psi) \\ s(\phi)s(\theta) + c(\phi)c(\psi) \\ c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \end{bmatrix} \right)$$

$$J\dot{\omega} + \omega^{\times} J\omega + \Delta^{\mathrm{T}}\ddot{\eta} = \tau_{\mathrm{gg}} + u,$$

$$\ddot{\eta} + C_d\dot{\eta} + K\eta = -\Delta\dot{\omega},$$

$$\omega^{\times} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \quad \tau_{\mathrm{gg}} = \begin{bmatrix} -3n^2(J_2 - J_3)c_\phi s_\phi c_\theta^2 \\ 3n^2(J_3 - J_1)c_\phi c_\theta s_\theta \\ 3n^2(J_1 - J_2)s_\phi c_\theta s_\theta \end{bmatrix}$$

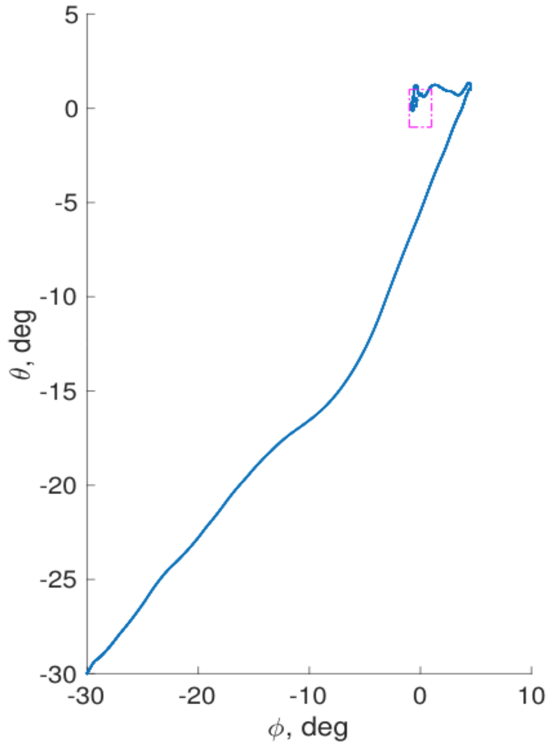
$$\Delta = \frac{-1}{\sqrt{10}} \begin{bmatrix} 6.45637 & 1.27814 & 2.15620 \\ -1.25619 & 0.91756 & -1.67264 \\ 1.11687 & 2.48901 & -0.83674 \\ 1.23637 & -2.6581 & -1.12503 \end{bmatrix} \mathrm{kg}^{1/2} \cdot \mathrm{m},$$

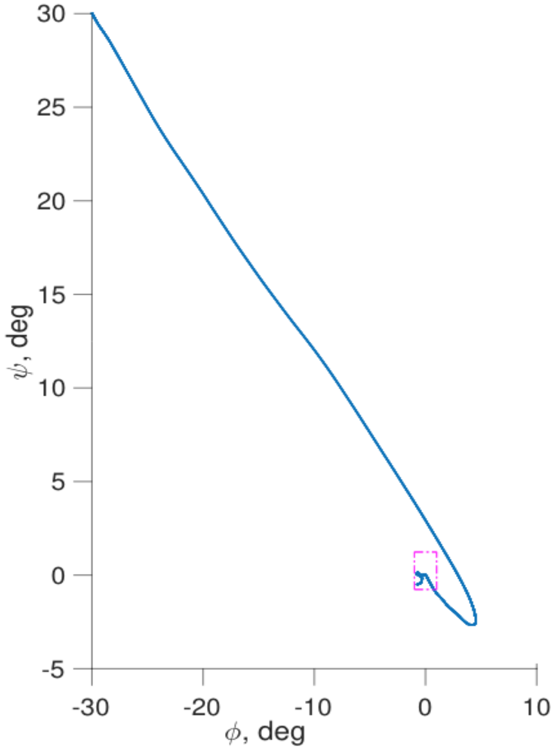
$$C_d = \mathrm{diag}(0.0086,\ 0.0190,\ 0.0487,\ 0.1275)\ \mathrm{sec}^{-1},\ \ K = \mathrm{diag}(0.59,\ 1.2184,\ 3.5093,\ 6.5004)\ \mathrm{sec}^{-2}$$

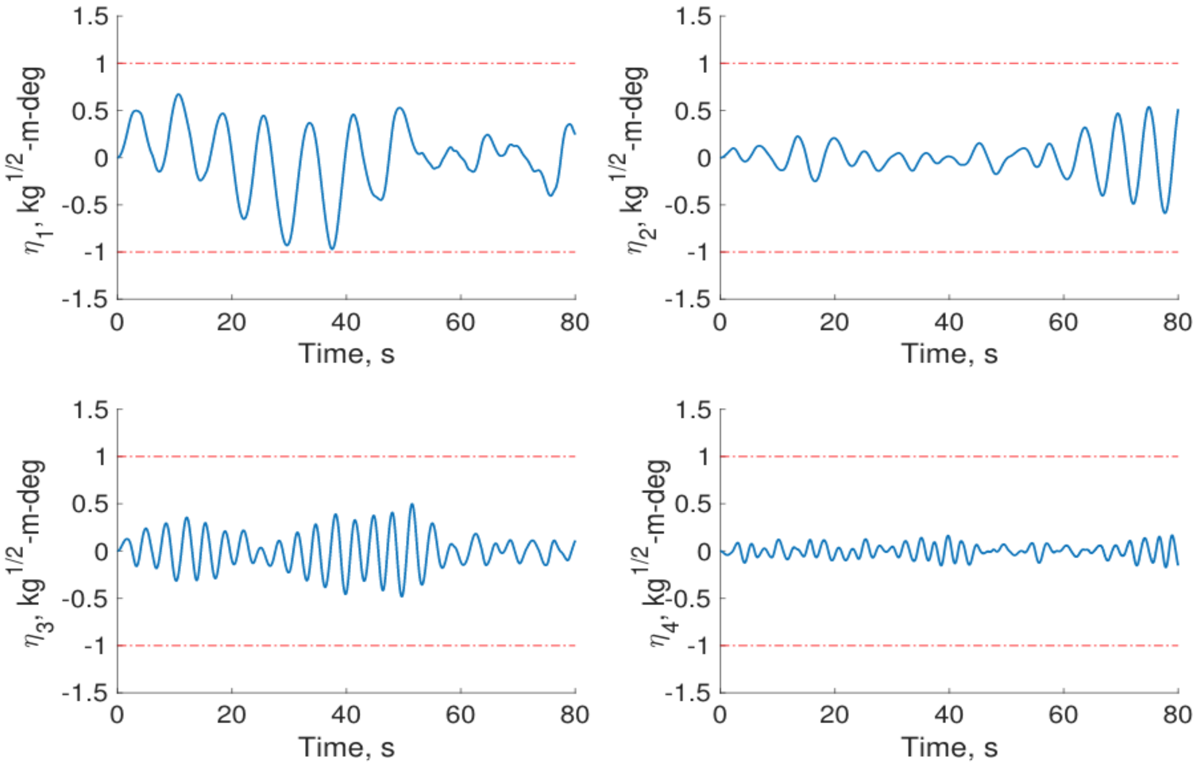
$$n = 2\pi/5400\ \mathrm{sec}^{-1},\quad J_1 = 150\ \mathrm{kg}\cdot\mathrm{m}^2,\ J_2 = 50\ \mathrm{kg}\cdot\mathrm{m}^2,\ J_3 = 170\ \mathrm{kg}\cdot\mathrm{m}^2$$

Simulation results

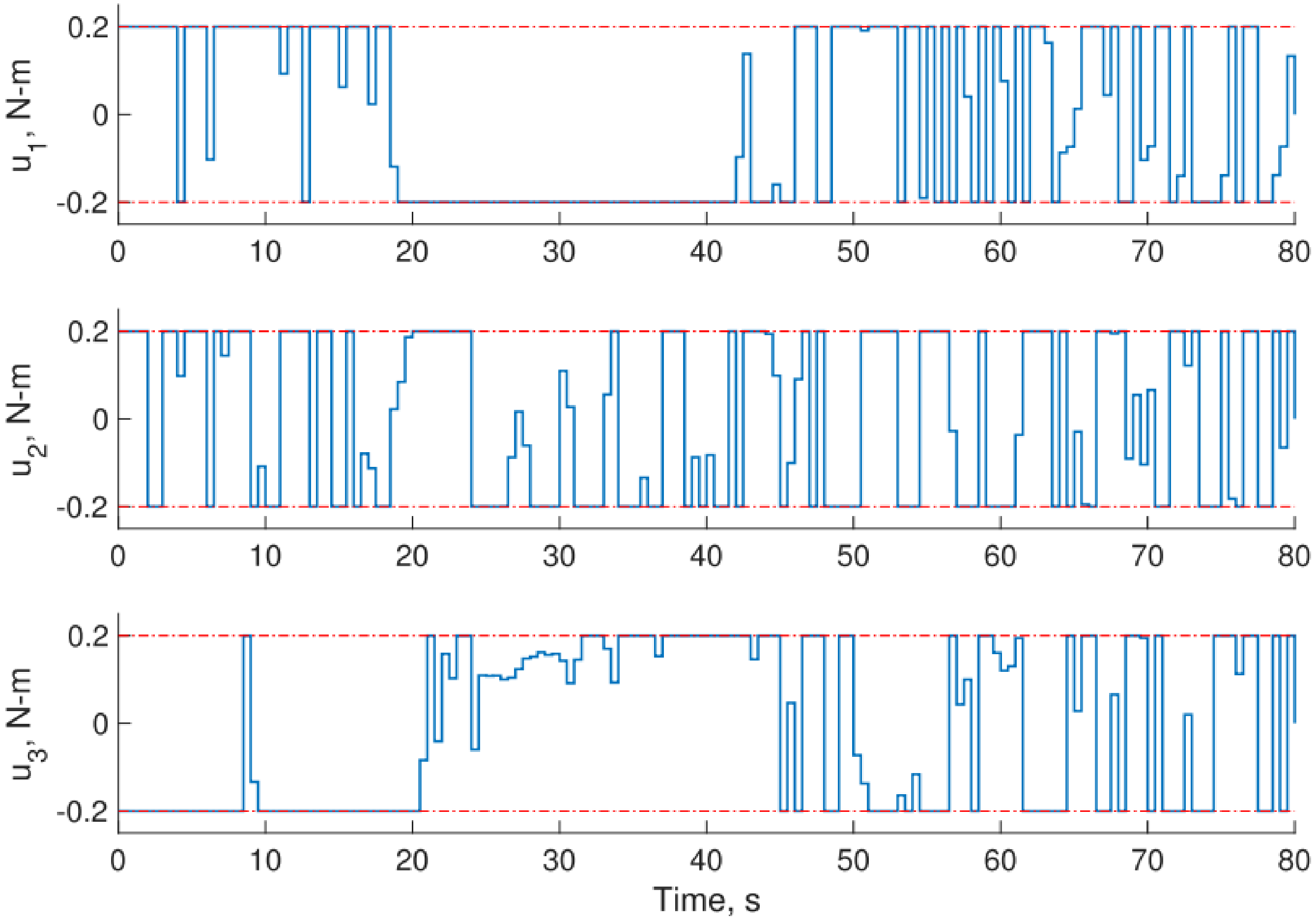
- In the simulation results, the controller is able to maintain the constraints on the flexible modes while executing the attitude change maneuver in minimum-time







Simulation results



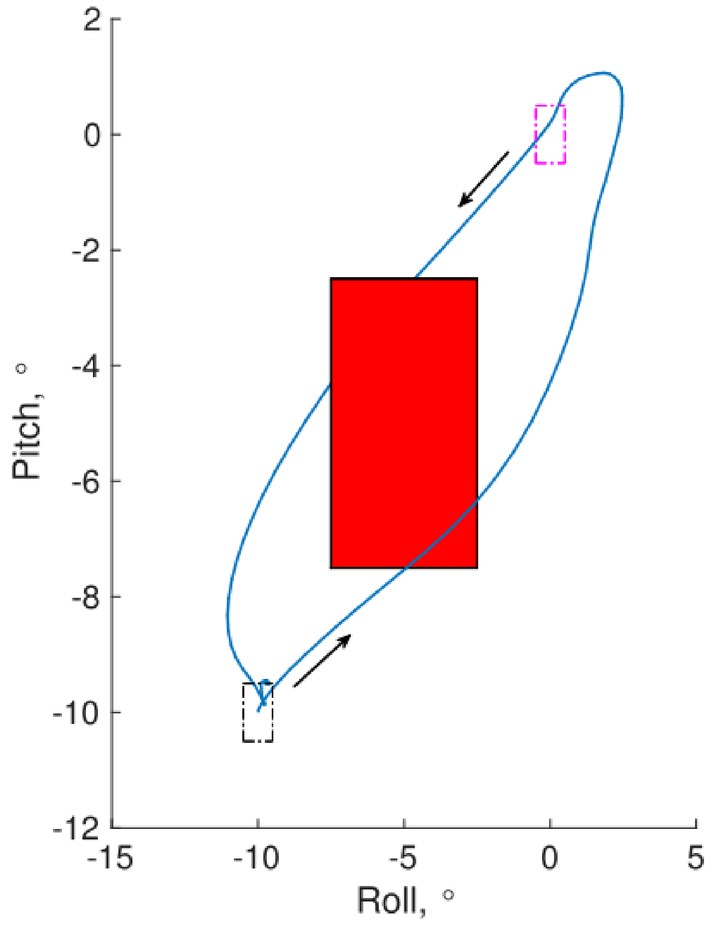
Exclusion Zone Avoidance

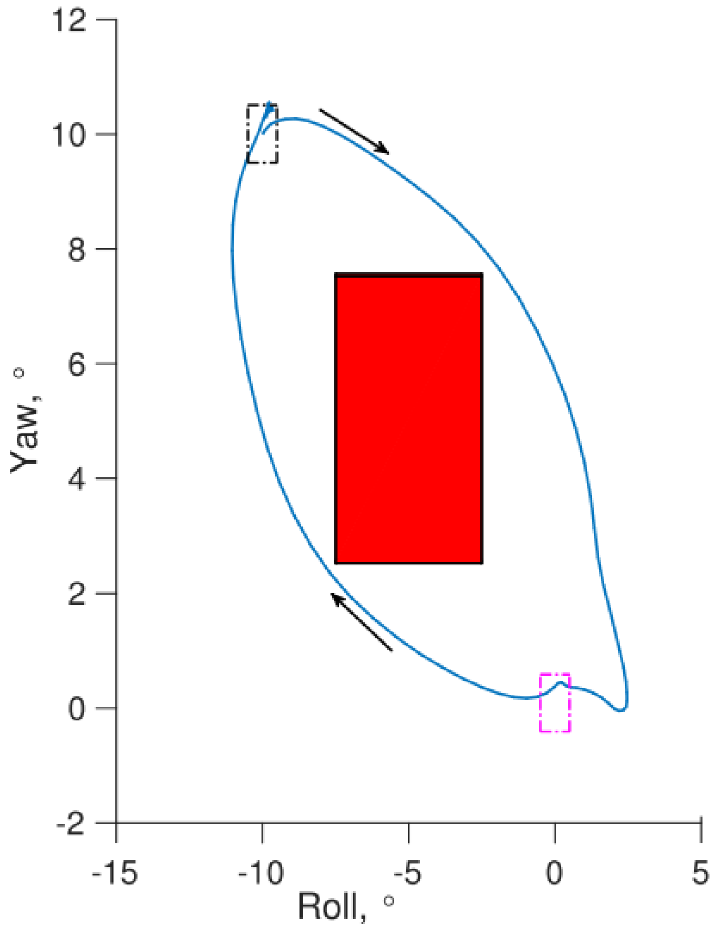
- The minimum-time MILP formulation can be augmented with additional binary parameters, ϵ, to represent exclusion zones.
- As an example, consider a rectangular exclusion zone for Euler angles, ϕ, θ, ψ. For $M > 0$ sufficiently large, the following inequality constraints can be added to MILP:

$$\phi_k \leq \phi_l + M\epsilon_{1,k}$$
$$-\phi_k \leq -\phi_u + M\epsilon_{2,k}$$
$$\theta_k \leq \theta_l + M\epsilon_{3,k}$$
$$-\theta_k \leq -\theta_u + M\epsilon_{4,k}$$
$$\psi_k \leq \psi_l + M\epsilon_{5,k}$$
$$-\psi_k \leq -\psi_u + M\epsilon_{6,k}$$
$$\sum_{i=1}^{6} \epsilon_{i,k} \leq 5$$
$$\epsilon_{i,k} \in \{0,1\},\ i = 1,2,3,4,5,6$$

Simulation results: Maneuvering rigid spacecraft with orientation exclusion zones

- The control moments are limited to +/- 0.1 Nm and random unmeasured disturbance torques sampled from the uniform distribution over the interval [-0.01, 0.01]





Summary

- MPC strategies can be devised for drift counteraction problems which involve maximizing expected time (or yield) before constraint violation
- Dynamic programming solutions can become sub-optimal due to numerical approximations, and are impractical for high order systems
- Value iteration algorithm can be modified to obtain faster convergence
- Much room remains for future research on these problems

References

Zidek, R., Kolmanovsky, I.V., and Bemporad, A., "Spacecraft drift counteraction optimal control: Open-loop and receding horizon solutions," AIAA Journal of Guidance, Control and Dynamics, vol. 41, no. 9, pp. 1859-1872, September 2018

Zidek, R., and Kolmanovsky, I.V., "Drift counteraction optimal control for deterministic systems and enhancing convergence of value iteration," *Automatica,* vol. 83, pp. 108-115, September, 2017.

Zidek, R., Kolmanovsky, I.V., and Bemporad, A., "Stochastic MPC approach to drift counteraction," Proceedings of 2018 Annual American Control Conference (ACC), June 27–29, 2018. Wisconsin Center, Milwaukee, USA, pp. 721-727.

Zidek, R., and Kolmanovsky, I., "A new algorithm for a class of deterministic drift counteraction optimal control problems," *Proceedings of 2017 American Control Conference*, Seattle, pp. 623-629.

Sutherland, R., Kolmanovsky, I.V., Girard, A., Leve, F.A., Petersen, C.D., "Minimum-time model predictive spacecraft attitude control for waypoint following and exclusion zone avoidance," Proceedings of AIAA SciTech Forum, 7-11 January 2019, San Diego,

BACKUP

Deterministic DCOC (alternative formulation)

Nonlinear discrete-time model: $x_{t+1} = f(x_t, u_t), \; x_0 \in G$

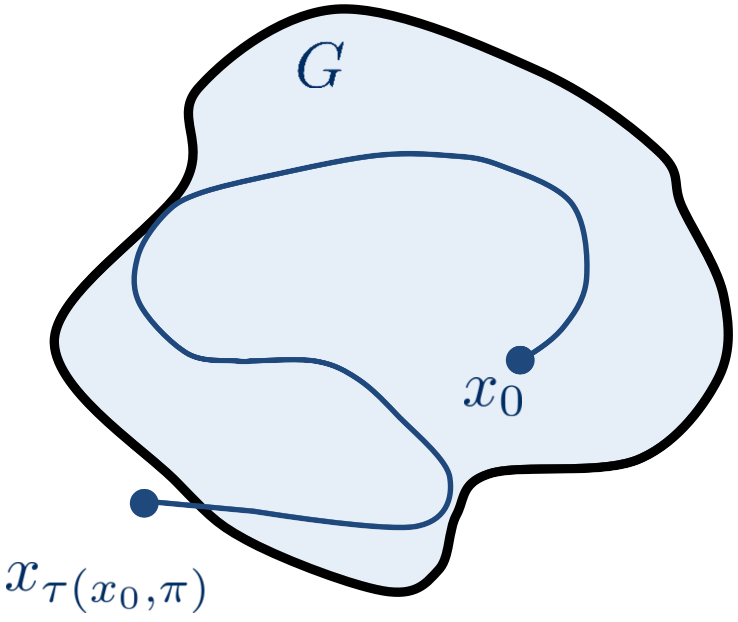
Control policy: $u_t = \pi(x_t) \in U$

First exit-time: $\tau(x_0, \pi) = \inf\{t \in \mathbb{Z}_+ : \; x_t \notin G\}$

Instantaneous reward: $g(x, u) \geq 0$

Maximize reward before constraint violation

$$\sum_{t=0}^{\tau(x_0,\pi)-1} g(x_t, u_t) \to \max_{\pi \in \Pi}$$



Most frequently: $g(x, u) = 1 \Rightarrow$ maximize time before constraint violation

Π = set of admissible control policies