Hybrid Optimization: Accelerating Convergence with Robustness in Optimization-Based Feedback Control

Ricardo Sanfelice
Department Electrical and Computer Engineering
University of California

CoE Review @ Zoom - October 30, 2020

Outline of Recent Results

1. **Estimation**
   - Finite-time Parameter Estimation via Hybrid Methods
     ACC 21a, ACC 21b, ACC 21c (all submitted), + *CoE collab*
   - Observers for Hybrid Systems *ACC 20, CDC 19, CDC 20, Automatica (submitted)*
2. **Safety**
   - Reachable maps for hybrid systems and regularity
     HSCC 20, TAC 19, NAHS 20, HSCC 20, CDC 20 (submitted)
   - (Necessary and Sufficient) Safety Certificates, with Events *ACC 21a, ACC 21d (submitted), TAC 20* + *CoE collab*
3. **Optimization**
   - High Performance Optimization via Uniting Control
     ACC 19, MTNS 20, ACC 20e (submitted) + AFRL/RV collab.
   - Model Predictive Control for Hybrid Systems *ACC 20, CDC 20, IFAC WC 20 Workshop*

Hybrid Optimal Control Problem

Problem ($\star$)

Given an initial condition x_0,

$$\begin{aligned} \underset{(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0)}{\text{minimize}} \quad & \mathcal{J}(x,u) \\ \text{subject to} \quad & (T,J)\in\mathcal{T} \\ & x(T,J)\in X, \end{aligned}$$

where (T,J) is the terminal time of (x,u).

Hybrid Optimal Control Problem

Problem ($\star$)

Given an initial condition x_0,

$$\begin{aligned} \underset{(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0)}{\text{minimize}} \quad & \mathcal{J}(x,u) \\ \text{subject to} \quad & (T,J)\in\mathcal{T} \\ & x(T,J)\in X, \end{aligned}$$

where (T,J) is the terminal time of (x,u).

- The *feasible set* $\mathcal{X}$, set of all x_0 with *feasible* $(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0)$.
- The *value function* $\mathcal{J}^* : \mathcal{X}\to\mathbb{R}_{\geq 0}$, defined as

$$\mathcal{J}^*(x_0) := \inf_{\substack{(x,u)\in\mathcal{S}_{\mathcal{H}_P}(x_0)\\ (T,J)\in\mathcal{T}\\ x(T,J)\in X}} \mathcal{J}(x,u) \quad \forall x_0\in\mathcal{X}.$$

Hybrid Optimization

- Solve optimization problems using accelerated methods with guaranteed performance

  Nesterov's algorithm: $\ddot{\xi} + a\dot{\xi} + b\nabla L(\xi + c\dot{\xi}) = 0$

  Heavy ball algorithm: $\ddot{\xi} + a\dot{\xi} + b\nabla L(\xi) = 0$

  a, b, and c are constants, L is cost function

  - Fast convergence to minimizers leads to overshoot/oscillations
  - Constraints typically encoded as soft constraints

- Solve MPC problems (CT, DT, and hybrid) using accelerated methods

Hybrid Optimization

- Solve optimization problems using accelerated methods with guaranteed performance

  Nesterov's algorithm: $\ddot{\xi} + a\dot{\xi} + b\nabla L(\xi + c\dot{\xi}) = 0$

  Heavy ball algorithm: $\ddot{\xi} + a\dot{\xi} + b\nabla L(\xi) = 0$

  a, b, and c are constants, L is cost function

  - Fast convergence to minimizers leads to overshoot/oscillations
  - Constraints typically encoded as soft constraints

- Solve MPC problems (CT, DT, and hybrid) using accelerated methods

Research collaboration with AFRL/RV (Christopher Petersen and Sean Phillips) on optimization and hybrid systems, with applications to orbital maneuvering

Accelerating Convergence with Robustness in Optimization-Based Feedback Control

Dawn M. Hustig-Schultz
and Ricardo G. Sanfelice

Department of Electrical and Computer Engineering
Hybrid Systems Laboratory
University of California, Santa Cruz

Assured Autonomy in Contest Environments

—

Fall 2020 Review

—

October 30, 2020

Optimization Algorithms as ODEs

Accelerated gradient methods have an added "velocity" term

$$\ddot{\xi} + 2d\dot{\xi} + \frac{1}{M\zeta^2}\nabla L(\xi + \beta\dot{\xi}) = 0$$

Accelerated gradient methods have an added "velocity" term

$$\ddot{\xi} + 2d\dot{\xi} + \frac{1}{M\zeta^2}\nabla L(\xi + \beta\dot{\xi}) = 0$$

From this ODE, derive a plant, where $\xi := z_1$ and $\dot{\xi} = z_2$ and output $y = h(z)$.

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ u \end{bmatrix} \quad (z, u) \in \mathbb{R}^{2n} \times \mathbb{R}^n$$

Optimization Algorithms as ODEs

Accelerated gradient methods have an added "velocity" term

$$\ddot{\xi} + 2d\dot{\xi} + \frac{1}{M\zeta^2}\nabla L(\xi + \beta\dot{\xi}) = 0$$

From this ODE, derive a plant, where $\xi := z_1$ and $\dot{\xi} = z_2$ and output $y = h(z)$.

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ u \end{bmatrix} \qquad (z, u) \in \mathbb{R}^{2n} \times \mathbb{R}^n$$

And derive a control algorithm, with Lipschitz constant $M > 0$,

$$\kappa(h(z)) = -2dz_2 - \frac{1}{M\zeta^2}\nabla L(z_1 + \beta z_2) \qquad (\star)$$

Optimization Algorithms as ODEs

Accelerated gradient methods have an added "velocity" term

$$\ddot{\xi} + 2d\dot{\xi} + \frac{1}{M\zeta^2}\nabla L(\xi + \beta\dot{\xi}) = 0$$

From this ODE, derive a plant, where $\xi := z_1$ and $\dot{\xi} = z_2$ and output $y = h(z)$.

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ u \end{bmatrix} \qquad (z, u) \in \mathbb{R}^{2n} \times \mathbb{R}^n$$

And derive a control algorithm, with Lipschitz constant $M > 0$,

$$\kappa(h(z)) = -2dz_2 - \frac{1}{M\zeta^2}\nabla L(z_1 + \beta z_2) \qquad (\star)$$

A static state-feedback law is given by $(\star)$

Optimization Algorithms as ODEs

Accelerated gradient methods have an added "velocity" term

$$\ddot{\xi} + 2d\dot{\xi} + \frac{1}{M\zeta^2}\nabla L(\xi + \beta\dot{\xi}) = 0$$

From this ODE, derive a plant, where $\xi := z_1$ and $\dot{\xi} = z_2$ and output $y = h(z)$.

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ u \end{bmatrix} \qquad (z, u) \in \mathbb{R}^{2n} \times \mathbb{R}^n$$

And derive a control algorithm, with Lipschitz constant $M > 0$,

$$\kappa(h(z)) = -2dz_2 - \frac{1}{M\zeta^2}\nabla L(z_1 + \beta z_2) \qquad (\star)$$

A static state-feedback law is given by $(\star)$

- This optimization algorithm is an *accelerated gradient method* called the *Nesterov's accelerated gradient descent* [Nesterov 83].
- It models a mass-spring-damper with a curvature-dependent damping term [Muehlebach and Jordan 19]. The constant $\zeta > 0$ rescales solutions in time, and d and β take different forms depending on the convexity of L.

Optimization Algorithms as ODEs

Another accelerated optimization algorithm adds a "velocity" term to classical gradient descent

$$\ddot{\xi} + \lambda\dot{\xi} + \gamma\nabla L(\xi) = 0$$

For $\xi := z_1$ and $\dot{\xi} = z_2$, we derive the same plant

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ u \end{bmatrix} \qquad (z, u) \in \mathbb{R}^{2n} \times \mathbb{R}^n$$

with output $y = h(z)$.

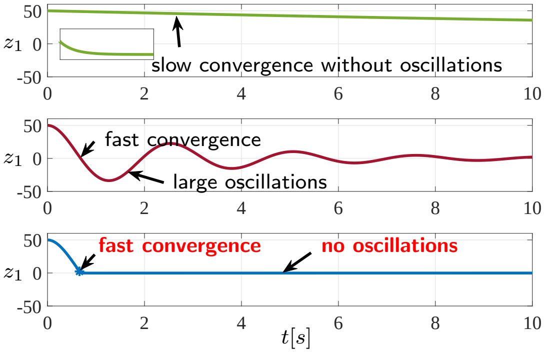
But with a different static state-feedback for the control algorithm

$$\kappa(h(z)) = -\lambda z_2 - \gamma\nabla L(z_1) \qquad (\star)$$

- This accelerated gradient method is called the *heavy ball method* [Polyak 63].
- It models the dynamics of a particle sliding on a profile defined by L (the *objective function*), where $\lambda > 0$ represents *friction* and $\gamma > 0$ represents *gravity*.

Unconstrained Hybrid Optimization Algorithms for Performance Improvement

Motivation



Heavy ball:

- Large λ: slow
- Small λ: fast, but large oscillations

Nesterov:

- Fast, but large oscillations
- Hard to guarantee uniform global asymptotic stability (UGAS)

A Logic-based algorithm, preserving the rates of the individual algorithms, is needed to ensure fast convergence and UGAS of the set of minimizers.

Uniting Approach

Nesterov's algorithm is used as global feedback to the plant, and heavy ball (with large λ) is used as local feedback

$$\kappa_0(h_0(z)) := -\lambda z_2 - \gamma \nabla L(z_1), \quad \kappa_1(h_1(z)) := -2dz_2 - \frac{1}{M}\nabla L(z_1 + \beta z_2)$$

where $\zeta = 1$ and where d and β are defined, for condition number $\kappa := \frac{M}{\mu}$ and strong convexity constant $\mu > 0$, as

$$d := \frac{1}{\sqrt{\kappa}+1}, \quad \beta := \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$$

and with h defined for the individual optimization algorithms as

$$h_0(z) := \begin{bmatrix} z_2 \\ \nabla L(z_1) \end{bmatrix}, \quad h_1(z) := \begin{bmatrix} z_2 \\ \nabla L(z_1 + \beta z_2) \end{bmatrix}$$

Optimization parameters can be designed using both of the following:

- Lyapunov functions of the form (with $a > 0$ properly chosen)

$$V_0(z) := \gamma_q \left(L(z_1) - L^*\right) + \frac{1}{2}\left|z_2\right|^2$$

$$V_1(z) := \frac{1}{2}\left|a(z_1 - z_1^*) + z_2\right|^2 + \frac{1}{M}\left(L(z_1) - L^*\right)$$

- The hybrid systems tools in [Goebel, Sanfelice, Teel, 12 PUP].

Nesterov's algorithm is used as global feedback to the plant, and heavy ball (with large λ) is used as local feedback

$$\kappa_0(h_0(z)) := -\lambda z_2 - \gamma \nabla L(z_1), \quad \kappa_1(h_1(z)) := -2dz_2 - \frac{1}{M}\nabla L(z_1 + \beta z_2)$$

where $\zeta = 1$ and where d and β are defined, for condition number $\kappa := \frac{M}{\mu}$ and strong convexity constant $\mu > 0$, as

$$d := \frac{1}{\sqrt{\kappa}+1}, \quad \beta := \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$$

and with h defined for the individual optimization algorithms as

Assumption 1 (Strong convexity of L)

L is strongly convex with $\mu > 0$, namely, for all $u_1, z_1 \in \mathbb{R}^n$,

1. $\nabla^2 L(z_1) \geq \mu I$;
2. $L(u_1) \geq L(z_1) + \langle \nabla L(z_1), u_1 - z_1 \rangle + \frac{\mu}{2} |u_1 - z_1|^2$.

Assumption 2 (Lipschitz continuity of L)

∇L is Lipschitz continuous with constant $M > 0$, namely,

$$|\nabla L(z_1) - \nabla L(u_1)| \leq M |z_1 - u_1|$$

for all $u_1, z_1 \in \mathbb{R}^n$.

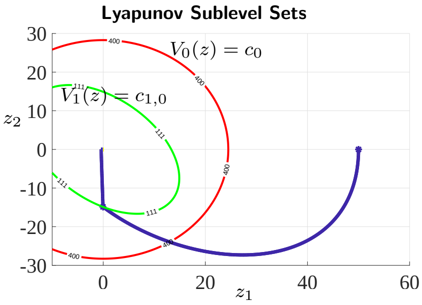
Switching Rules

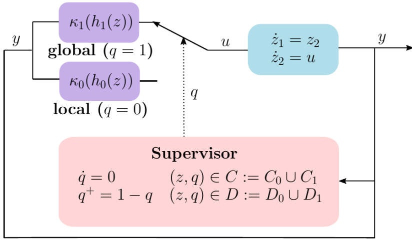
The algorithm is modeled as a hybrid system $\mathcal{H}$ with state $(z,q) \in \mathbb{R}^{2n} \times Q := \{0,1\}$, data (C,F,D,G), and $c_{1,0} \in (0,c_0)$, $c_0 > 0$:

$$F(x) := \begin{bmatrix} z_2 \\ \kappa_q(h_q(z)) \\ 0 \end{bmatrix} \quad \forall x \in C := C_0 \cup C_1$$

$$G(x) := \begin{bmatrix} z \\ 1-q \end{bmatrix} \quad \forall x \in D := D_0 \cup D_1$$

$$C_0 := \left\{z \in \mathbb{R}^2 : V_0(z) \le c_0\right\} \times \{0\}$$
$$C_1 := \left\{z \in \mathbb{R}^2 : V_1(z) \ge c_{1,0}\right\} \times \{1\}$$
$$D_0 := \left\{z \in \mathbb{R}^2 : V_0(z) \ge c_0\right\} \times \{0\}$$
$$D_1 := \left\{z \in \mathbb{R}^2 : V_1(z) \le c_{1,0}\right\} \times \{1\}$$





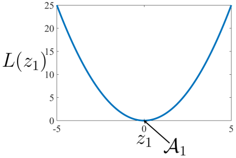
UGAS and Convergence Rate

Uniform global asymptotic stability for (UGAS) $\mathcal{H}$: all solutions that start close to the set of interest $\mathcal{A}$ for the state (z_1, z_2, q) stay close to, and converge to $\mathcal{A}$.

To construct this set, we want:

- The state z_1 to be in the set of minimizers $\mathcal{A}_1 := \{z_1 \in \mathbb{R}^n \ : \ \nabla L(z_1) = 0 \}$;
- The state z_2 to be zero;
- The state $q = 0$ (local algorithm active)

This yields the set of interest $\mathcal{A} := \mathcal{A}_1 \times \{0\} \times \{0\}$.



Convergence rate describes how fast, in the worst case, the value of the objective function L approaches L^*. For example, the bound

$$L(z_1(t)) - L^* \leq (L(z_1(0)) - L^*) \exp(-mt)$$

has the rate $\exp(-mt)$, with $m > 0$.

Theorem (UGAS of $\mathcal{A}$ and convergence rate for $\mathcal{H}$)

Let L satisfy Assumptions 1 and 2. Additionally, let $\lambda > 0$, $\gamma > 0$, and $c_{1,0} \in (0, c_0)$, $c_0 > 0$. Then, the set $\mathcal{A}$ is uniformly globally asymptotically stable for $\mathcal{H}$. Furthermore, each maximal solution $(t,j) \mapsto x(t,j) = (z_1(t,j), z_2(t,j), q(t,j))$ of the hybrid closed-loop algorithm $\mathcal{H}$ starting from C_1[a] satisfies

$$L(z_1(t,j)) - L^* \leq (L(z_1(0,0)) - L^*)\exp(-at)$$

when $q(t,j) = 1$ and

$$L(z_1(t_1,1)) - L^* \leq (L(z_1(0,0)) - L^*)\exp(-2\mu t)$$

when $q(t,j) = 0$, for all $(t,j) \in \operatorname{dom} x$. The constant $\mu > 0$ is the strong convexity constant and $a > 0$ is defined, for $\kappa := \frac{M}{\mu} \geq 1$, as

$$a := d + \frac{\beta}{2\kappa} = \frac{1}{\sqrt{\kappa}} - \frac{1}{2\kappa}.$$

[a] For solutions not in C_0, a similar bound, keeping track of the two jumps, can be written.

Outline of Proof

- The heavy ball algorithm ($\mathcal{H}_0$) satisfies
$$\dot{V}_0 = \langle \nabla V_0(z), F_p(z, \kappa_q(h(z))) \rangle = -\lambda \left| z_2 \right|^2 \leq 0$$
for all $\lambda > 0$, and $\gamma > 0$, where F_P is the plant and h_0 is the output;
- The largest weakly invariant set for $\mathcal{H}_0$ contained in
$$\left\{ z \in \mathbb{R}^{2n} : \dot{V}_q(z) = 0 \right\} \cap \left\{ z \in \mathbb{R}^{2n} : V_q(z) = r_q \right\}$$
is when $r_q = 0$, which is equal to $\mathcal{A}_1 \times \{0\}$;
- Nesterov's algorithm ($\mathcal{H}_1$) satisfies $\dot{V}_1(z) \leq -aV_1(z)$;
- By an invariance principle (for $\mathcal{H}_0$), since every maximal solution is complete, $\mathcal{H}_0$ and $\mathcal{H}_1$ have $\mathcal{A}_1 \times \{0\}$ uniformly globally asymptotically stable;
- Uniform global asymptotic stability of $\mathcal{A}$ for $\mathcal{H}$ follows from the construction of G and D.
- Convergence rate for $\mathcal{H}_0$ follows from strong convexity of L, and convergence rate for $\mathcal{H}_1$ follows from Grönwall's Lemma.

Hybrid Algorithm for Comparison

In [Poveda and Li CDC19], a Nesterov-like reset algorithm for strongly convex L was proposed. Using an alternate state space representation $z_1 := \xi$ and $z_2 := \xi + \frac{\tau}{2}\dot{\xi}$, the HAND-2 algorithm has state $(z, \tau) \in \mathbb{R}^{2n+1}$ and data (C, F, D, G):

$$F(z,\tau) := \begin{bmatrix} \frac{2}{\tau}(z_2 - z_1) \\ -2c\tau\nabla L(z_1) \\ 1 \end{bmatrix} \quad \forall (z,\tau) \in C \qquad \begin{aligned} C &:= \left\{ (z,\tau) \in \mathbb{R}^{2n+1} \ : \ \tau \in [T_{\min}, T_{\max}] \right\} \\ D &:= \left\{ (z,\tau) \in \mathbb{R}^{2n+1} \ : \ \tau \geq T_{\max} \right\} \end{aligned}$$

$$G(z,\tau) := \begin{bmatrix} G_z(z,\tau) \\ T_{\min} \end{bmatrix} \qquad \forall (z,\tau) \in D$$

where $c > 0$, $G_z(z,\tau) := [z_1^\top z_1^\top]^\top$, and $0 < T_{\min} < T_{\max} < \infty$.
It is shown that each maximal solution $(t,j) \mapsto (z_1(t,j), z_2(t,j), \tau(t,j))$ of the HAND-2 algorithm satisfies

$$L(z_1(t,j)) - L^* \leq k_a \left| z_1(0,0) - z_1^* \right|^2 \exp\left(-\tilde{k}_b \tilde{\alpha}\,(t+j) \right)$$

for all $(t,j) \in \operatorname{dom}(z,\tau)$, where $k_a := 0.5 k_1 M$, $M > 0$, $k_1 := \frac{(c\mu)^{-1} + T_{\min}^2}{\Delta T^2}$, $\Delta T := T_{\max} - T_{\min}$, $0 < T_{\min} < T_{\max}$, $\frac{1}{c\mu} < T_{\max}^2 - T_{\min}^2$, $\tilde{k}_b := 1 - \frac{(c\mu)^{-1} + T_{\min}^2}{T_{\max}^2}$, and $j \geq \tilde{\alpha}(t+j) := \frac{\max\{t+j-\Delta T, 0\}}{\Delta T + 1}$.

Hybrid Algorithm for Comparison

In [Poveda and Li CDC19], a Nesterov-like reset algorithm for strongly convex L was proposed. Using an alternate state space representation $z_1 := \xi$ and $z_2 := \xi + \frac{\tau}{2}\dot{\xi}$, the HAND-2 algorithm has state $(z,\tau) \in \mathbb{R}^{2n+1}$ and data (C, F, D, G):

$$F(z,\tau) := \begin{bmatrix} \frac{2}{\tau}(z_2 - z_1) \\ -2c\tau\nabla L(z_1) \\ 1 \end{bmatrix} \quad \forall (z,\tau) \in C \qquad C := \left\{(z,\tau) \in \mathbb{R}^{2n+1} \ : \ \tau \in [T_{\min}, T_{\max}] \right\}$$

$$D := \left\{(z,\tau) \in \mathbb{R}^{2n+1} \ : \ \tau \geq T_{\max} \right\}$$

$$G(z,\tau) := \begin{bmatrix} G_z(z,\tau) \\ T_{\min} \end{bmatrix} \quad \forall (z,\tau) \in D$$

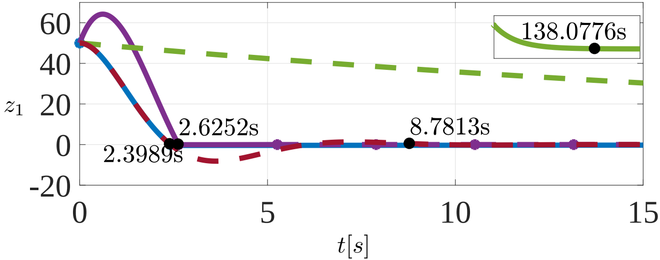
where $c > 0$, $G_z(z,\tau) := [z_1^\top z_1^\top]^\top$, and $0 < T_{\min} < T_{\max} < \infty$.
It is shown that each maximal solution $(t,j) \mapsto (z_1(t,j), z_2(t,j), \tau(t,j))$ of the HAND-2 algorithm satisfies

$$L(z_1(t,j)) - L^* \leq k_a \left|z_1(0,0) - z_1^*\right|^2 \exp\left(-\tilde{k}_b \tilde{\alpha}\,(t+j)\right)$$

This bound holds when: $z_1(0,0) = z_2(0,0)$ and $\tau(0,0) = T_{\min}$.

$\tilde{k}_b := 1 - \frac{(c\mu)^{-1}+T_{\min}^2}{T_{\max}^2}$, and $j \geq \tilde{\alpha}(t+j) := \frac{\max\{t+j-\Delta T, 0\}}{\Delta T + 1}$.

Numerical Example



- $L(z_1) := z_1^2$, with single minimum $\mathcal{A}_1 := \{0\}$.
- $\mathcal{H}_0$: $\lambda = 40$, $\gamma = \frac{2}{3}$, $\mathcal{H}_1$: $M = 2$, $\kappa = 1$.
- Sublevel set2: $c_0 = 400$, $c_{1,0} = 111$.
- HAND-2: $T_{\min} = 3$, $T_{\max} = 5.63$, $c = 0.25$.
- Initial conditions ($\mathcal{H}_0$, $\mathcal{H}_1$, and $\mathcal{H}$): $z_1(0,0) = 50$, $z_2(0,0) = 0$, $q = 0$.
- HAND-2 initial conditions: $z_1(0,0) = 50$, $z_2(0,0) = 50$, $\tau = T_{\min}$.

Conclusions

- The hybrid closed-loop system $\mathcal{H}$ is 8.6% faster than HAND-2, 72.7% faster than $\mathcal{H}_1$, and 98.3% faster than $\mathcal{H}_0$.
- The different initial velocity of $\mathcal{H}$ is key to its improved performance and lack of overshoot compared to HAND-2.

Constrained Optimization For Fast Solvers (A Work in Progress)

In collaboration with AFRL/RV, C. Petersen and S. Phillips

The Optimal Control Problem

For a plant $\psi_{k+1} := A_d\psi_k + B_d\nu_k$, $\psi_k \in \mathbb{R}^m$ and $\nu_k \in \mathbb{R}^s$:

$$\min_{x_k,u_k} \sum_{k=0}^{N} l(x_k,u_k) + \phi(x_N)$$

Horizon: N; Stage cost: $l(x_k,u_k)$; Terminal cost: $\phi(x_N)$

$$\text{s.t. } x_0 = \psi_k$$
$$g(x_k,u_k) = 0$$
$$x_N \in \mathbb{X}_N$$

(Equality Constraints)

$$h(u_k) \le 0$$

(Inequality constraints)

$\Rightarrow$

$$\min_{z_1} J(z_1)$$
$$\text{s.t. } Gz_1 - g = 0$$
$$Hz_1 - h \le 0$$

$$z_1 := \begin{bmatrix} x_0 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

For MPC, we want to

- Find the minimum $z_1^* \in \mathbb{R}^{m_z}$ of the cost function J $(m_{z_1} := N(m+s))$;
- Use $\nu_k := u_0^*$ for one step of the plant;
- Repeat.

Solving the OCP

- The Lagrangian for the reformulated problem, with weights $\eta \in \mathbb{R}^{m_\eta}$ and $\mu \in \mathbb{R}^{m_h}$, $m_\eta = Nm$, is

$$\mathcal{L}(z_1, \eta, \mu) := J(z_1) + \eta^\top (G z_1 - g) + \mu^\top h(z_1)$$

- The solution must satisfy the Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned} \nabla_{z_1} \mathcal{L}(z_1, \eta, \mu) &= 0 \\ g - G z_1 &= 0 \\ -h(z_1) + N_+(\mu) &\ni 0 \end{aligned}$$

$$N_+(\mu) = \begin{cases} \{ w \in \mathbb{R}^{m_h} \; : \; \langle w, y - \mu \rangle \leq 0 \; \forall y \geq 0 \}, & \text{if } \mu \geq 0 \\ \emptyset, & \text{if } \mu < 0. \end{cases}$$

OCP Solver

In [Nicotra, et. al. IEEE TAC19] both the OCP and the KKT conditions were solved with gradient flow. Let $\alpha > 0$ and $\dot{\xi} = \dot{z}_1$, $\xi = z_1$:

Unconstrained GF

$$\dot{\xi} = -\alpha \nabla L(\xi)$$

Constrained GF

$$\begin{bmatrix} \dot{z}_1 \\ \dot{\eta} \\ \dot{\mu} \end{bmatrix} = -\alpha \begin{bmatrix} \nabla_{z_1} \mathcal{L}(z_1, \eta, \mu) \\ g - G z_1 \\ -h(z_1) + P_N(h(z_1), \mu) \end{bmatrix}$$

$$P_N(h(z_1), \mu) = \underset{w \in N_+(\mu)}{\operatorname{argmin}} |w - h(z_1)|_2^2$$

- In [Nicotra, et. al. IEEE TAC19], this constrained ODE was used to control a continuous-time plant.
- We intend to use this ODE as a solver for MPC, for a discrete-time plant.
- A uniting algorithm might be possible, if we can tune local and global algorithms via scalar or time-varying α.

Application: Orbital Maneuvers

$$\dot{\psi} = A\psi = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \dot{\psi}_1 \\ \dot{\psi}_2 \end{bmatrix}$$

$$A_d := \exp(AT)$$

$$B_d := A_d \begin{bmatrix} 0_2 \\ I_2 \end{bmatrix}$$

$$\psi_{k+1} := A_d \psi_k + B_d \nu_k$$

Photo: www.wikipedia.com

Let $\beta > 0$, $\xi := z_1$, $\dot{\xi} = z_2$, $p_1 := [z_1^\top \ \eta_1^\top \ \mu_1^\top]^\top$, and $p_2 := [z_2^\top \ \eta_2^\top \ \mu_2^\top]^\top$:

Unconstrained HBF

$$\ddot{\xi} = -\beta\dot{\xi} - \alpha\nabla L(\xi)$$

Constrained HBF

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = \begin{bmatrix} p_2 \\ -\alpha T(p_1) - \beta p_2 \end{bmatrix}$$

$$T(p_1) := \begin{bmatrix} \nabla_{z_1}\mathcal{L}(z_1, \eta_1, \mu_1) \\ g - Gz_1 \\ -h(z_1) + P(h(z_1), \mu_1) \end{bmatrix}$$

Conclusion and Future Work

Uniting algorithm:

- In progress: derive and prove properties of switching rules not requiring knowledge of z_1^* or L^*.
- Applications to learning.

Solver:

- Derive convergence rate for [Nicotra, et. al. IEEE TAC19].
- Derive and prove properties of accelerated algorithms for constrained optimization (such as mirror descent), to use as a solver for MPC.
- Implement an aperiodic timer, for start of the optimization step.
- Implement early stopping mechanisms and time-varying parameters.

Nonstrongly Convex L

Denoting time as a state $\tau > 0$, the global feedback (Nesterov) and local feedback (heavy ball) to the plant are

$$\kappa_0(h_0(z)) := -\lambda z_2 - \gamma \nabla L(z_1), \quad \kappa_1(h_1(z)) := -2\bar{d}(\tau) z_2 - \frac{1}{M} \nabla L(z_1 + \bar{\beta}(\tau) z_2)$$

with $\bar{d}$ and $\bar{\beta}$ typically chosen as

$$\bar{d}(\tau) := \frac{3}{2(\tau+2)}, \quad \bar{\beta}(\tau) := \frac{\tau - 1}{\tau + 2}$$

The Lyapunov functions used to design the optimization parameters are

$$V_0(z) := \gamma_q \left(L(z_1) - L^*\right) + \frac{1}{2} |z_2|^2$$

$$V_1(z) := \frac{1}{2} \left| \bar{a}(\tau) \, |z_1|_{\mathcal{A}_1} + z_2 \right|^2 + \frac{1}{M} (L(z_1) - L^*)$$

where $\bar{a}(\tau)$ is

$$\bar{a}(\tau) = \frac{2}{\tau + 2}$$

The algorithm is modeled as a hybrid system $\mathcal{H}$ with state $(z, q, \tau) \in \mathbb{R}^{2n} \times Q \times \mathbb{R}_{\geq 0}$, $Q := \{0, 1\}$, data (C, F, D, G), and $c_{1,0} \in (0, c_0)$, $c_0 > 0$, as follows:

$$F(x) := \begin{bmatrix} \begin{bmatrix} z_2 \\ \kappa_q(h_q(z)) \end{bmatrix} \\ 0 \\ q \end{bmatrix} \qquad \forall x \in C := C_0 \cup C_1$$

$$G(x) := \begin{bmatrix} z \\ 1 - q \\ 0 \end{bmatrix} \qquad \forall x \in D := D_0 \cup D_1$$

$$C_0 := \left\{ z \in \mathbb{R}^2 : V_0(z) \leq c_0 \right\} \times \{0\} \times \mathbb{R}_{\geq 0}$$
$$C_1 := \left\{ z \in \mathbb{R}^2 : V_1(z) \geq c_{1,0} \right\} \times \{1\} \times \mathbb{R}_{\geq 0}$$
$$D_0 := \left\{ z \in \mathbb{R}^2 : V_0(z) \geq c_0 \right\} \times \{0\} \times \mathbb{R}_{\geq 0}$$
$$D_1 := \left\{ z \in \mathbb{R}^2 : V_1(z) \leq c_{1,0} \right\} \times \{1\} \times \mathbb{R}_{\geq 0}$$

Results for Nonstrongly Convex L

Assumption 3 (Nonstrongly convex L)

L is nonstrongly convex. Namely, for all $u_1, z_1 \in \mathbb{R}^n$,

$$L(u_1) \geq L(z_1) + \langle \nabla L(z_1), u_1 - z_1 \rangle$$

Assumption 4 (Properties of L with respect to $\mathcal{A}$)

1. $\mathcal{A}_1$ is compact and connected;
2. L is positive definite with respect to $\mathcal{A}_1$.

Results for Nonstrongly Convex L

Theorem (UGAS of $\mathcal{A}$ and convergence rate for $\mathcal{H}$)

Let L satisfy Assumptions 2, 3, and 4. Additionally, let $\lambda > 0$, $\gamma > 0$, and $c_{1,0} \in (0, c_0)$, $c_0 > 0$. Then, the set $\mathcal{A}$ is uniformly globally asymptotically stable for $\mathcal{H}$. Furthermore, each maximal solution $(t,j) \mapsto x(t,j) = (z_1(t,j), z_2(t,j), q(t,j))$ of the hybrid closed-loop algorithm $\mathcal{H}$ starting from C_1 satisfies

$$\frac{1}{M}(L(z_1(t,j)) - L^*) \leq \frac{9\bar{C}}{(t+2)^2}\left(|z_1(0,0)|^2_{\mathcal{A}_1} + |z_2(0,0)|^2\right)$$

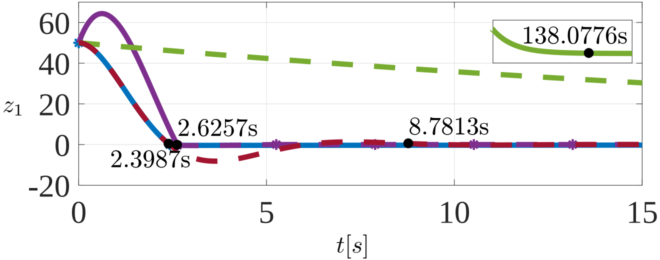
for all $t \geq 1$ when $q(t,j) = 1$ and

$$L(\overline{z}_1(T,1)) - L^* \leq \frac{V(z(0,0)) - V(z(T,j(T)))}{\lambda t}$$

when $q(t,j) = 0$, where $\bar{C} \in (0, 2\exp(M))$, $M > 0$ is the Lipschitz constant of ∇L, and $\overline{z}_1(T, j(T)) = \frac{1}{T}\int_0^T z_1(t, j(T))dt$[a], $(T,1) \in \operatorname{dom} x$.

[a]This equality is used to derive the bound for $q = 0$, via Jensen's inequality.

Numerical Example



- $L(z_1) := z_1^2$, with single minimum $\mathcal{A}_1 := \{0\}$.
- $\mathcal{H}_0$ parameters: $\lambda = 40$, $\gamma = \frac{2}{3}$, $\mathcal{H}_1$ parameter: $M = 2$.
- Sublevel set constants: $c_0 = 400$, $c_{1,0} = 111$.
- HAND-1 parameters [Poveda and Li CDC19]: $T_{\min} \approx 3.1047$, $T_{\max} = T_{\text{med}} + 1$, $c = 0.25$, $r = 51$, $\delta = 4236$.
- Initial conditions ($\mathcal{H}_0$, $\mathcal{H}_1$, and $\mathcal{H}$): $z_1(0,0) = 50$, $z_2(0,0) = 0$, $q = 0$.
- HAND-1 ICs: $z_1(0,0) = 50$, $z_2(0,0) = 50$, $\tau = T_{\min}$.

Conclusions

- The hybrid closed-loop system $\mathcal{H}$ is 8.6% faster than HAND-1, 72.7% faster than $\mathcal{H}_1$, and 98.3% faster than $\mathcal{H}_0$.
- The different initial velocity of $\mathcal{H}$ is key to its improved performance and lack of overshoot compared to HAND-1.