Towards Fast Mixed-integer Quadratic Programming Algorithms for Real-time Loco-manipulation Control

Xuan Lin
RoMeLa(Robotics and Mechanisms Laboratory)

ROBOTICS & MECHANISMS LABORATORY

Table of Content

SCALER: Versatile Multi-Limbed Robot for Free-Climbing in Extreme Terrains

Yusuke Tanaka[1], Yuki Shirai[1], Alexander Schperberg[1], Xuan Lin[1], and Dennis Hong[1]

Generalized Benders Decomposition with Continual Learning for Hybrid Model Predictive Control in Dynamic Environment

Xuan Lin[†]

SCALER: Versatile Multi-Limbed Robot for Free-Climbing in Extreme Terrains

Climbing robots

Why interested in climbing?

- If legged robots can climb on the walls, it significantly extend their capability

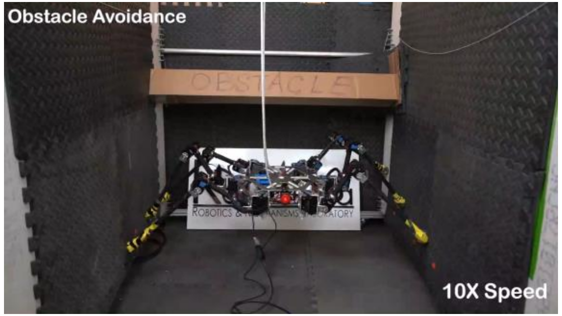
- Climbing requires dedicated grippers for various surfaces, and high-performance power grasps to move the robot itself

NASA LEMUR robot climbing up a cliff in Titus Canyon and scan for ancient fossils

Boston Dynamics RiSE robot performing an untethered climb of a multistory building

SiLVIA

SiLVIA (Six-Legged Vehicle with Intelligent Articulation)



Investigate the compliance model and its usage in motion planning [1]

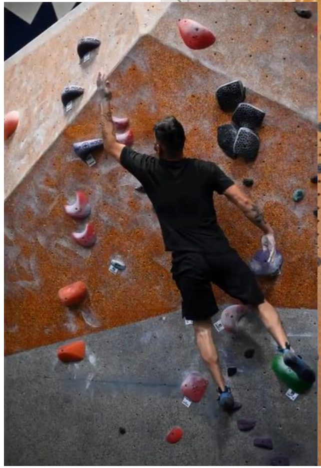
- IROS best paper on search and rescue robots

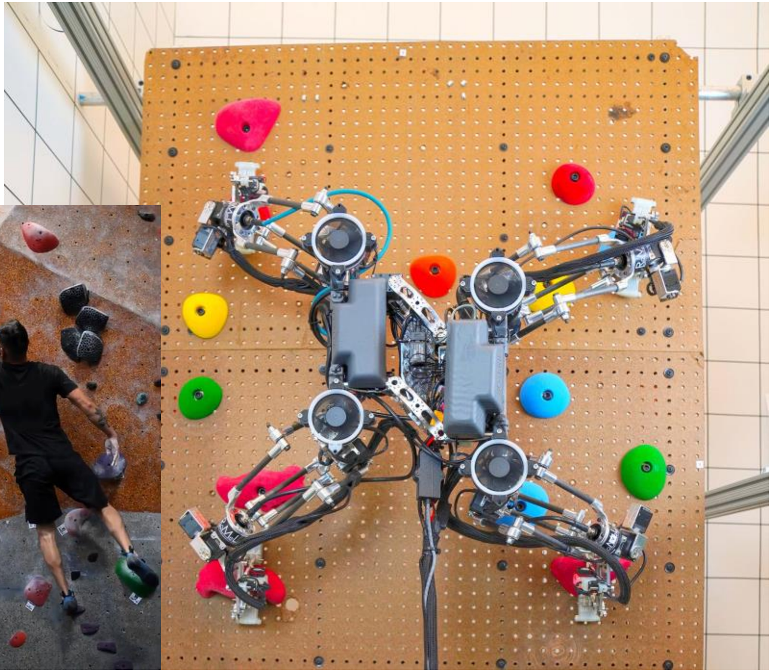
[1] Lin, Xuan, Jingwen Zhang, Junjie Shen, Gabriel Fernandez, and Dennis W. Hong. "Optimization based motion planning for multi-limbed vertical climbing robots." In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1918-1925. IEEE, 2019.

SCALER

SCALER: Spine-enhanced Climbing Autonomous Limbed Exploration Robot

- 4-legged, walking 13+6 DoF, climbing 29+6 DoF
- Position control servo motors in pairs
- Max torque 9 Nm
- Leg length walking 0.4 m, climbing 0.55 m
- Total weight walking 6.3 kg, climbing 8.8 kg
- IMU, encoder, FT sensor, vision
- Extra DoF inside body to enlarge workspace





SCALER

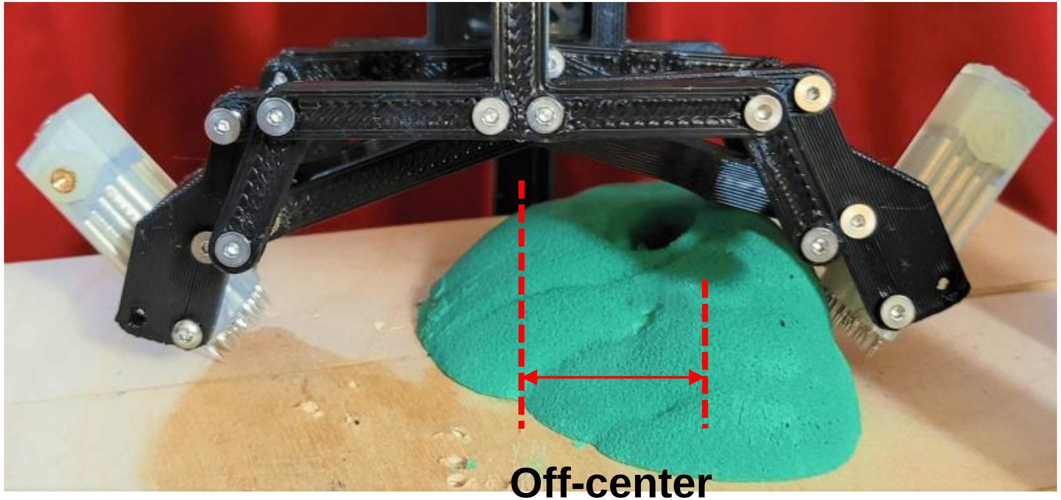
SCALER uses Grasping Onto Any Terrain (GOAT) gripper

Features of GOAT gripper

- Spine array or C shaped fingers
- Whipple tree mechanism for off-axis grasping
- Electrically or pneumatically actuated
- Force control

SCALER

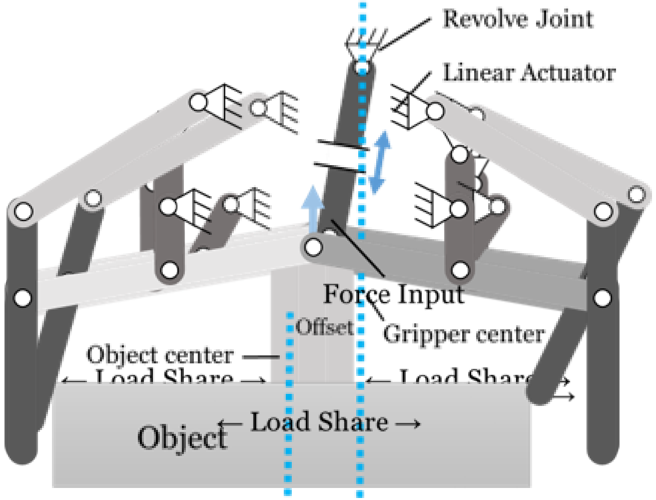
GOAT gripper needs to adapt to irregular objects



SCALER

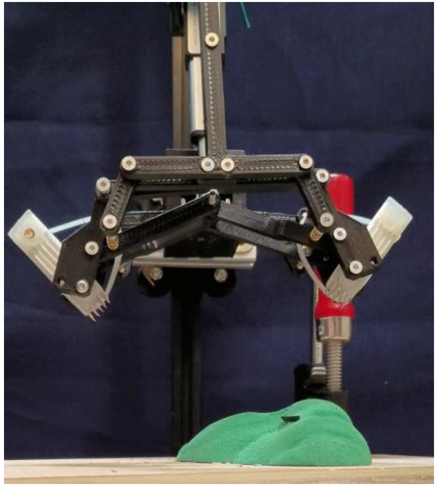
Whippletree mechanism that passively adapt to irregular objects

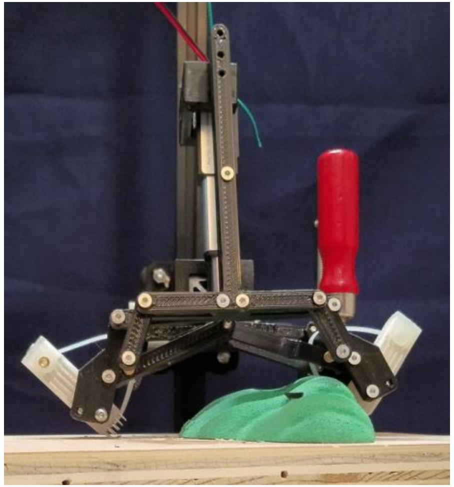
Use bars and pins instead of restoring springs and tendons

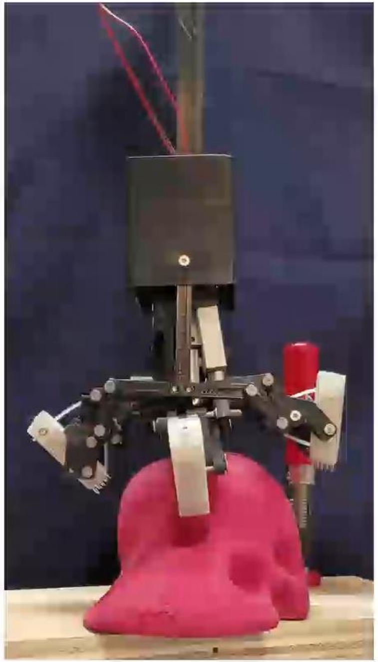


SCALER

Whippletree mechanism experiments





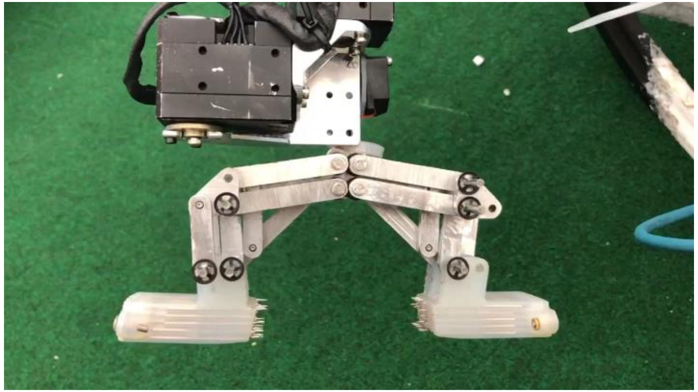


SCALER

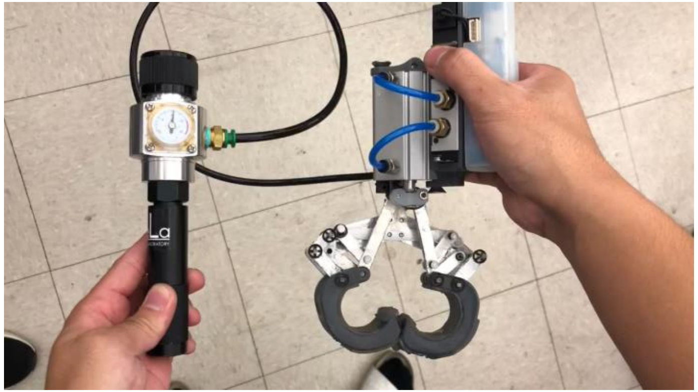
Actuation of GOAT gripper

- DC Linear actuator: 100-200N at 0.15Hz, force and position control

- CO_2 Pneumatic actuator: 200-300N at 5Hz, limited 100 grasps, no force or position control



DC Linear actuator

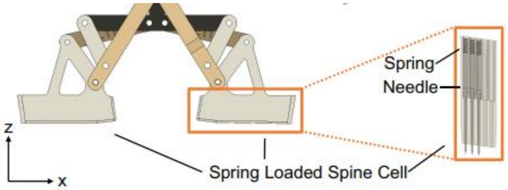


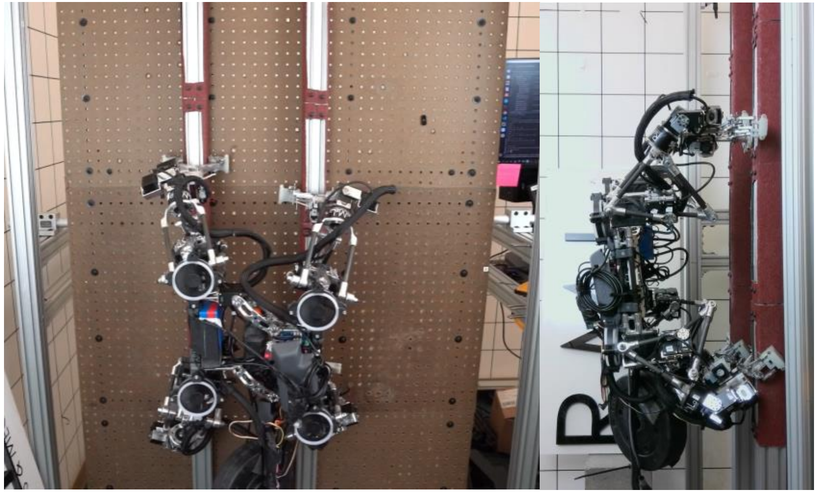
CO_2 Pneumatic actuator

SCALER

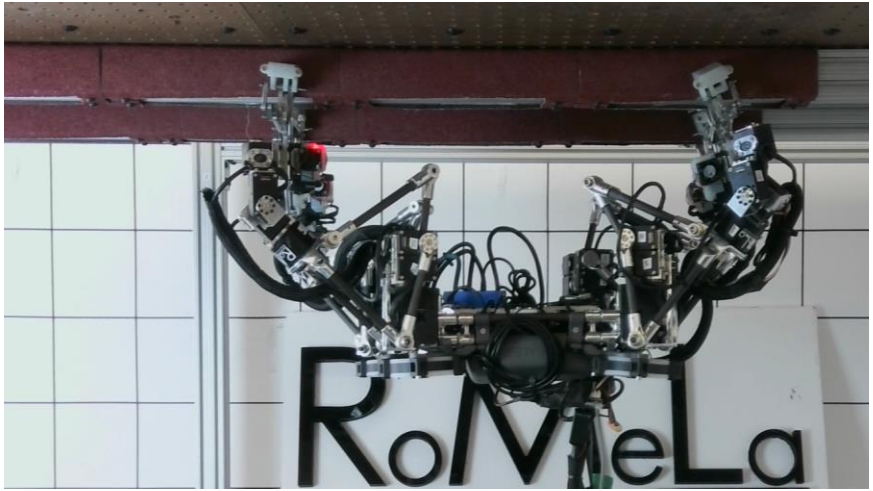
GOAT finger design

Spine for power grasp onto rough surfaces with large area





Vertical climbing with 3.4kg weight

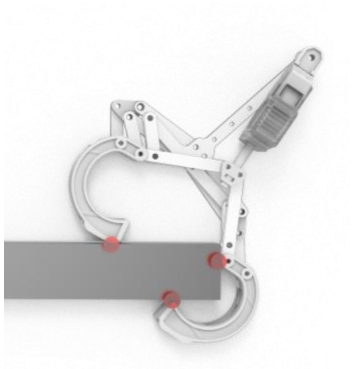


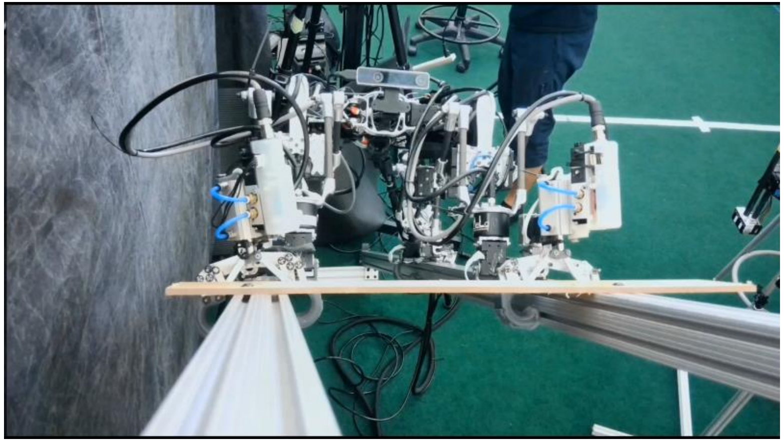
Inverted climbing

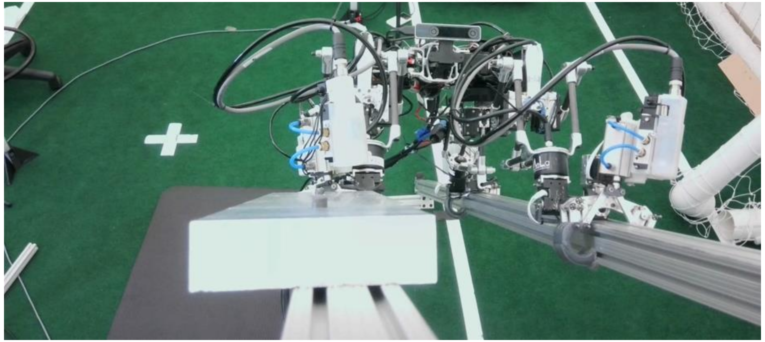
SCALER

GOAT finger design

C-shape fingers grasp onto thin obstacles





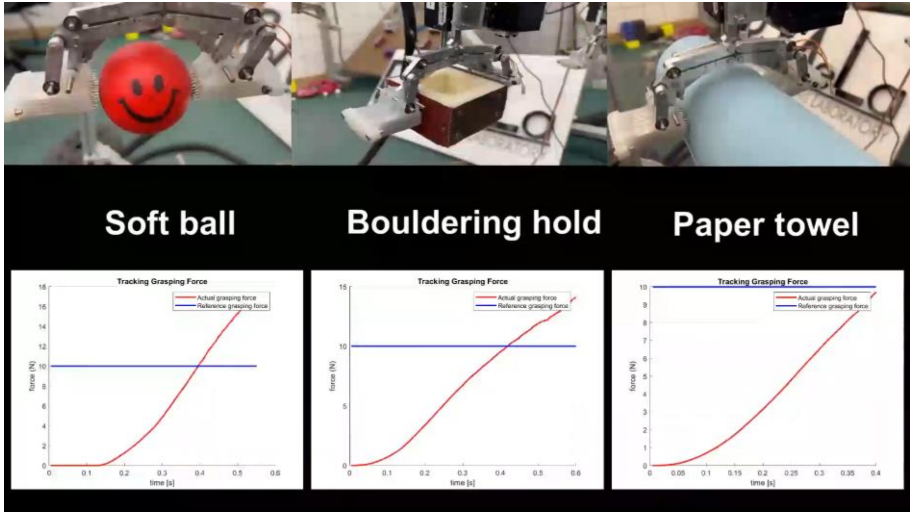


SCALER

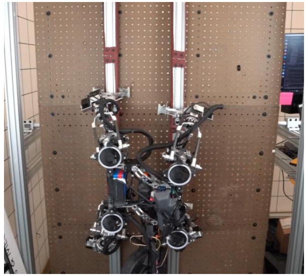
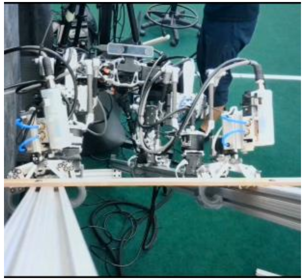
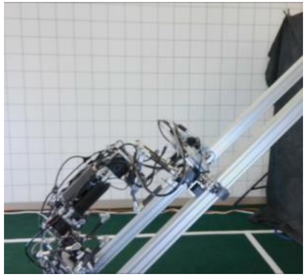
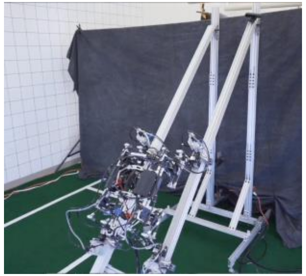
Admittance control – Please check out our paper presentation on Thursday 10:00-10:10, ThuO1T1.2

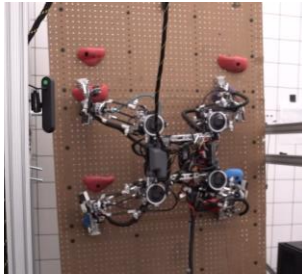
Title: Adaptive Force Controller for Contact-Rich Robotic Systems Using an Unscented Kalman Filter

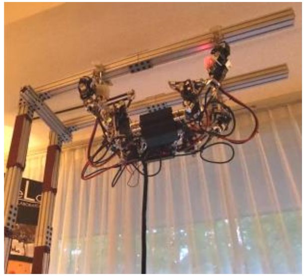
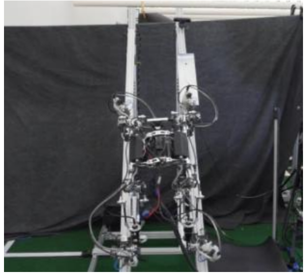
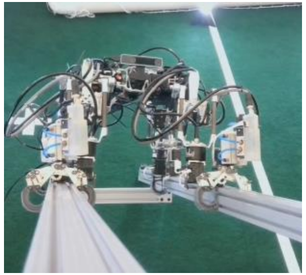
Authors: A Schperberg, Y Shirai, X Lin, Y Tanaka, D Hong



SCALER

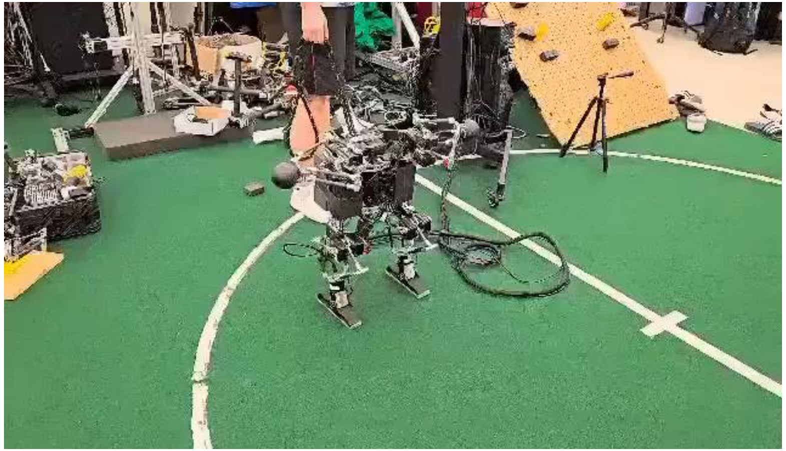








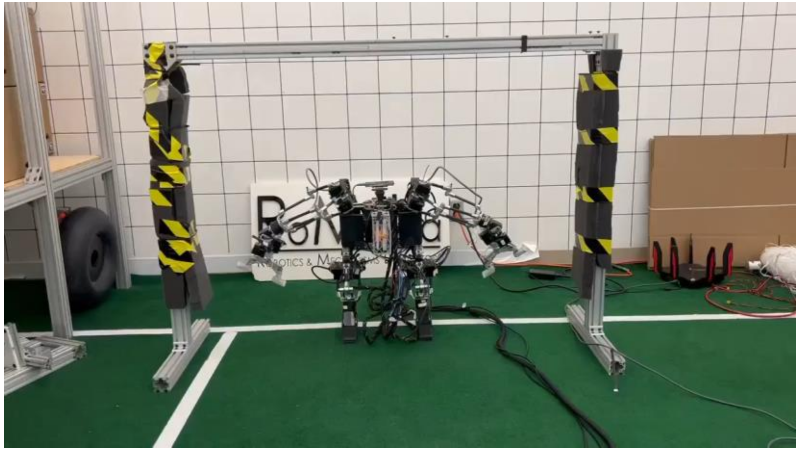




SCALER

For this conference …



Bipedal walking



Pull up

SCALER

The student team

Yusuke Tanaka

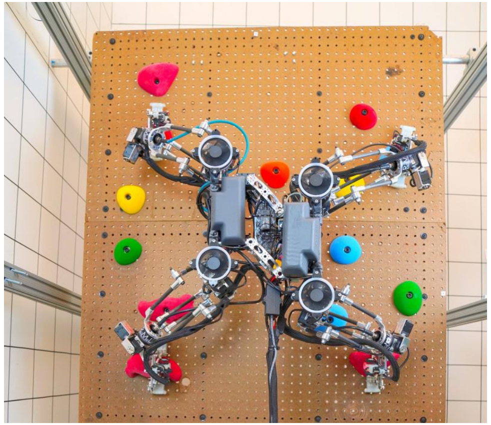
Yuki Shirai

Alexander Schperberg

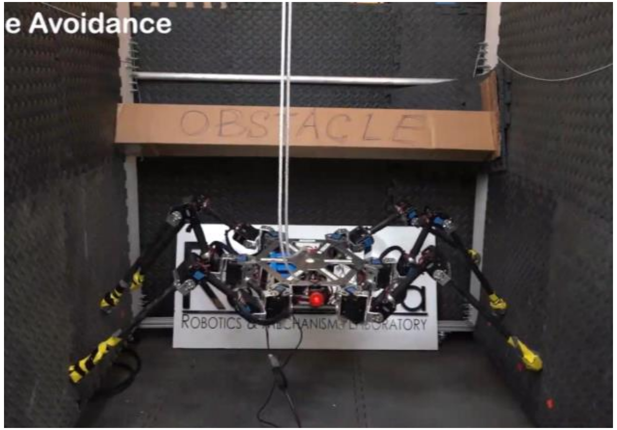
Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Many robotics applications such as hybrid MPC requires solving mixed-integer quadratic programming (MIQPs) very fast



Contact planning



Obstacle avoidance

The faster solver we have, the quicker that robot control can address model errors and disturbances

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

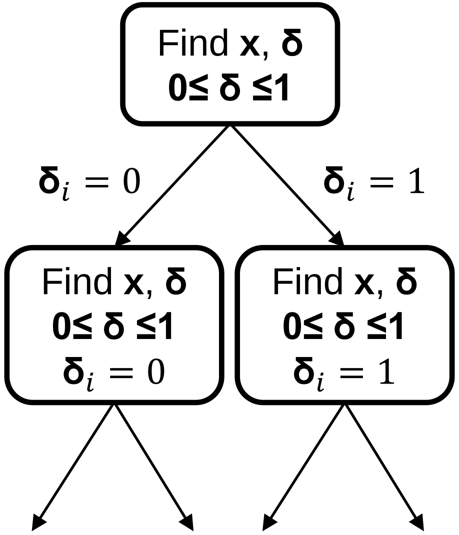
MIPs are NP-hard as in the worst case you search 2^B branches

Solving MIPs fast relies on choosing good binary branches to proceed

To achieve this:

- Commercialized solvers require heavy engineering: cutting planes, branching heuristics, presolve, multi-thread, …

- Recent work on ML methods [1-3] collect data and train classifiers for mode sequences offline



[1] Cauligi, Abhishek, Preston Culbertson, Edward Schmerling, Mac Schwager, Bartolomeo Stellato, and Marco Pavone. "Coco: Online mixed-integer control via supervised learning." *IEEE Robotics and Automation Letters* 7, no. 2 (2021

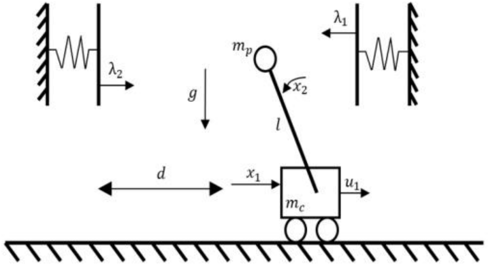
[2] Hogan, Francois R., and Alberto Rodriguez. "Reactive planar non-prehensile manipulation with hybrid model predictive control." The International Journal of Robotics Research

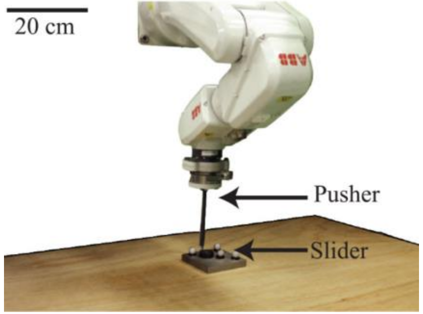
[3] Zhu, Jia-Jie, and Georg Martius. "Fast non-parametric learning to accelerate mixed-integer programming for hybrid model predictive control." IFAC-PapersOnLine

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

However, learning policies requires a lot of data

In this research, we investigate how we can make data-collection and training significantly faster





Problem	Cart-pole with soft walls	Planar pushing
Problem size	N=10, 40 binaries	N=35, 105 binaries
Training data	90000 (Cauligi, Abhishek, et al)	100000 (Hogan, F. R., & Rodriguez, A.)

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

For MIQPs:

- If mode sequence **δ** is fixed, the problem is QP(**δ**)

- Parametric QP has been studied by Explicit LQR paper [1], its feasible set is convex and cost function is convex and piecewise quadratic

- Difficult to write them down analytically as there are too many active sets

- We use duality theory to build boundaries of feasible set and lower bounds of cost function

[1] Bemporad, Alberto, Manfred Morari, Vivek Dua, and Efstratios N. Pistikopoulos. "The explicit linear quadratic regulator for constrained systems." *Automatica* 38, no. 1 (2002): 3-20.

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Duality theory can give lower bounds for the cost through cutting planes

We construct feasible set $V(\boldsymbol{\theta})$ for $\boldsymbol{\delta}$, and cost map $\mathrm{v}(\boldsymbol{\theta}, \boldsymbol{\delta})$

- parameter $\boldsymbol{\theta}$ includes initial conditions

Solving MIQP takes 2 steps:

1. Build feasible set $V(\boldsymbol{\theta})$ and cost map $\mathrm{v}(\boldsymbol{\theta}, \boldsymbol{\delta})$
2. Traverse the cost map

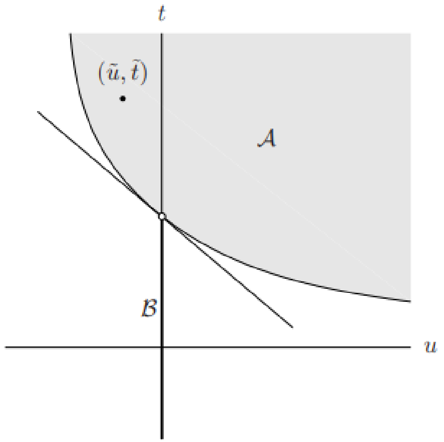


Figure 5.6, Chapter 5 Duality, Boyd, Stephen P., and Lieven Vandenberghe. *Convex optimization*

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

However, parametrically solve the problem still requires too many cutting planes

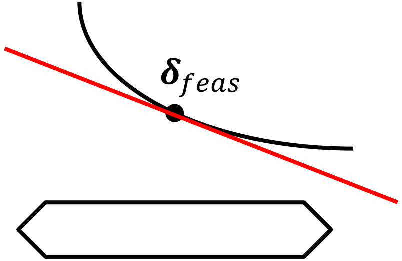
- Active sets =* extreme points and rays in dual cone ~ cutting planes

Instead, we start from an empty set of cuts, then pick up the necessary cuts “on the fly”

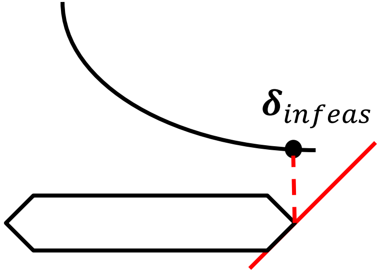
This technique is called Benders decomposition

- The part that provides binary sequences using $V(\boldsymbol{\theta})$, $\mathrm{v}(\boldsymbol{\theta}, \boldsymbol{\delta})$ is called Benders master problem

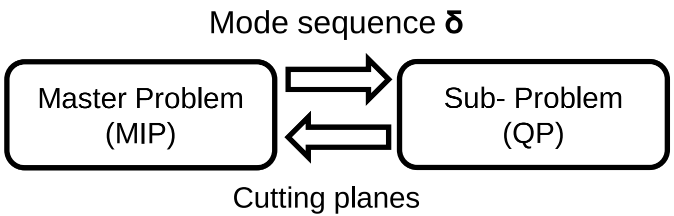
- The QP with fixed mode sequence **δ** is called Benders subproblem



QP feasible: Optimality cuts



QP Infeasible: Feasibility cuts



*Ignore degeneracy

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Benders decomposition is an old algorithm [1,2]

I propose a few new techniques to improve it:

- Continual learning and warm-start
- Fast cold-start using dynamic model
- Customized algorithm for master problem leveraging on sparsity of feasibility cuts

[1] J. Benders, “Partitioning procedures for solving mixed-variables programming problems” Numerische mathematik, vol. 4, no. 1, pp. 238–252, 1962.
[2] A. M. Geoffrion, “Generalized benders decomposition,” Journal of optimization theory and applications, vol. 10, pp. 237–260, 1972.

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Continual learning and warm-start

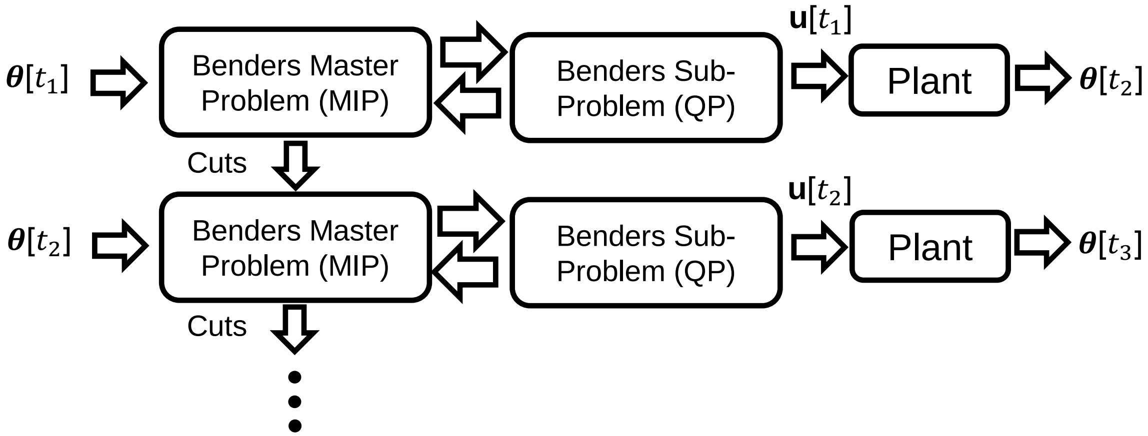
Use previous cuts for a new problem

$$\boldsymbol{\lambda}\boldsymbol{\theta}[t_1]+\boldsymbol{\upsilon}\boldsymbol{\delta} \geq 0$$

$$\downarrow$$

$$\boldsymbol{\lambda}\boldsymbol{\theta}[t_2]+\boldsymbol{\upsilon}\boldsymbol{\delta} \geq 0$$

Relies on the fast that dual cone (hence $\boldsymbol{\lambda}$,$\boldsymbol{\upsilon}$) is independent of $\boldsymbol{\theta}$



Observation: A small amount of cuts can already do decent warm-starts

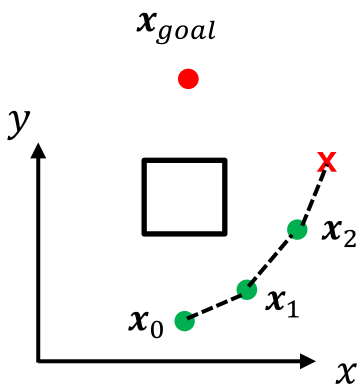
Great if this procedure can happen fast (before the system goes unstable)

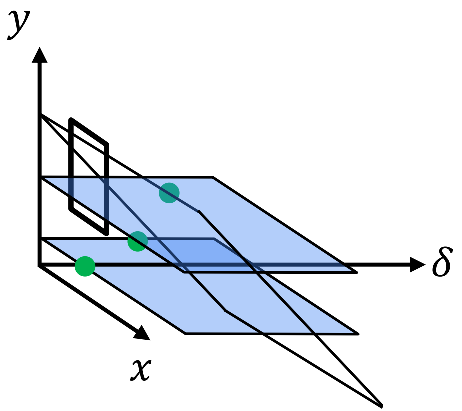
Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Fast cold-start using dynamic model

- Intuitively, the failure modes in the future time can also be informative in the current time

- This is done by shifting feasibility cuts backward in time





Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Customized Benders master problem solver utilizing sparsity of feasibility cuts

- A lot of master problems can be resolved in the presolve stage if Gurobi is used

- Greedy approach to solve binary variables step-by-step

$$\min \boxed{\boldsymbol{C}_1\boldsymbol{\delta}[t_1]} + \boldsymbol{C}_2\boldsymbol{\delta}[t_2] + \boldsymbol{C}_3\boldsymbol{\delta}[t_3]$$

$$\begin{aligned} \text{s.t.} \quad & \boxed{\boldsymbol{S}_{11}\boldsymbol{\delta}[t_1] \geq \boldsymbol{S}_{10}} \\ & \boldsymbol{S}_{21}\boldsymbol{\delta}[t_1] + \boldsymbol{S}_{22}\boldsymbol{\delta}[t_2] \geq \boldsymbol{S}_{20} \\ & \boldsymbol{S}_{31}\boldsymbol{\delta}[t_1] + \boldsymbol{S}_{32}\boldsymbol{\delta}[t_2] + \boldsymbol{S}_{33}\boldsymbol{\delta}[t_3] \geq \boldsymbol{S}_{30} \end{aligned}$$

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Customized Benders master problem solver utilizing sparsity of feasibility cuts

- A lot of master problems can be resolved in the presolve stage if Gurobi is used

- Greedy approach to solve binary variables step-by-step

$$\min \boxed{\boldsymbol{C}_2\boldsymbol{\delta}[t_2]} + \boldsymbol{C}_3\boldsymbol{\delta}[t_3]$$

$$\text{s.t.}$$

$$\boxed{\boldsymbol{S}_{22}\ \boldsymbol{\delta}[t_2] \geq \boldsymbol{S}'_{20}}$$

$$\boldsymbol{S}_{32}\ \boldsymbol{\delta}[t_2] + \boldsymbol{S}_{33}\ \boldsymbol{\delta}[t_3] \geq \boldsymbol{S}'_{30}$$

If $\boldsymbol{\delta}[t_2]$ becomes infeasible, go back to $\boldsymbol{\delta}[t_1]$ and find another solution

At the scale (N<20) we tested, it is effective

- Won’t be effective for long horizon problems, or controls that require taking “detour”

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Experiments

Computer: Intel Core i7-12800H × 20 laptop with 16GB memory

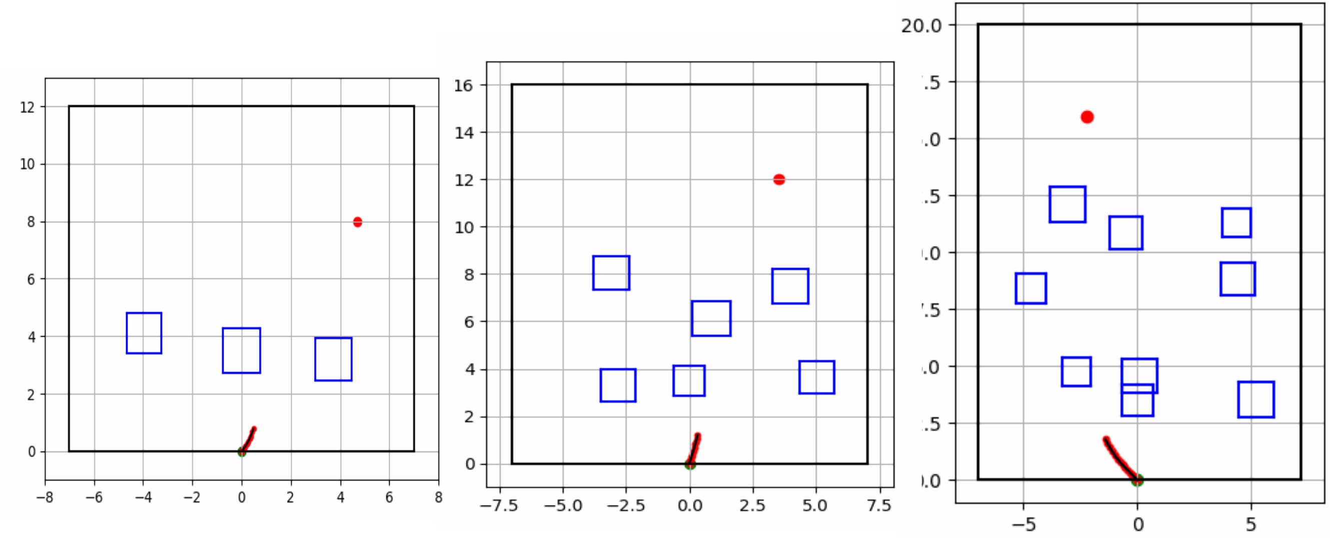
Proposed Benders MIQP solver: coded in Python, subproblem QPs solved by Gurobi

Benchmark: Use Gurobi 10.03 to directly solve MIQP

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Obstacle avoidance

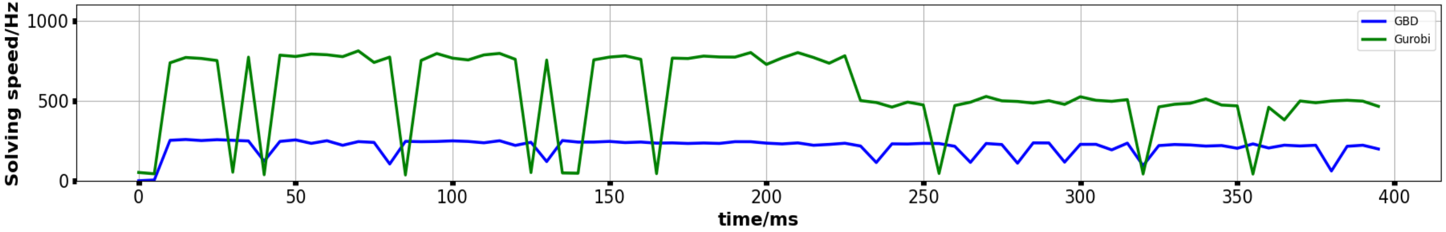
Mass $m = 1$
Random force disturbance $f = N(0,5)$



# of obstacles	3	6	9	12
Time horizon (N)	9	12	15	20
# of binaries	54	144	270	480

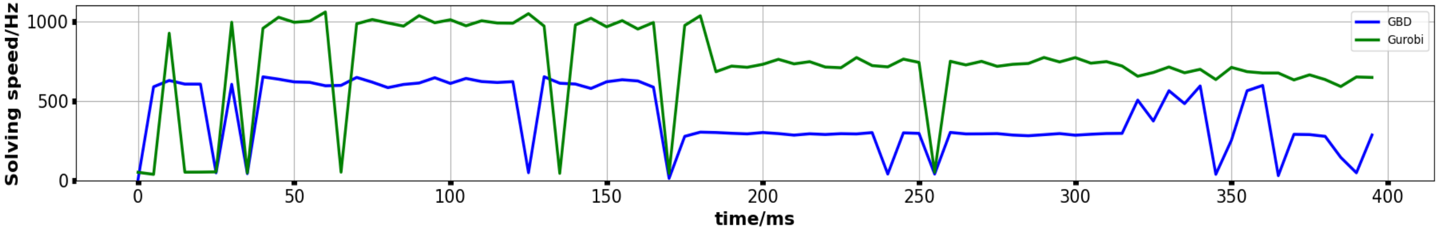
Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

3 obstacles



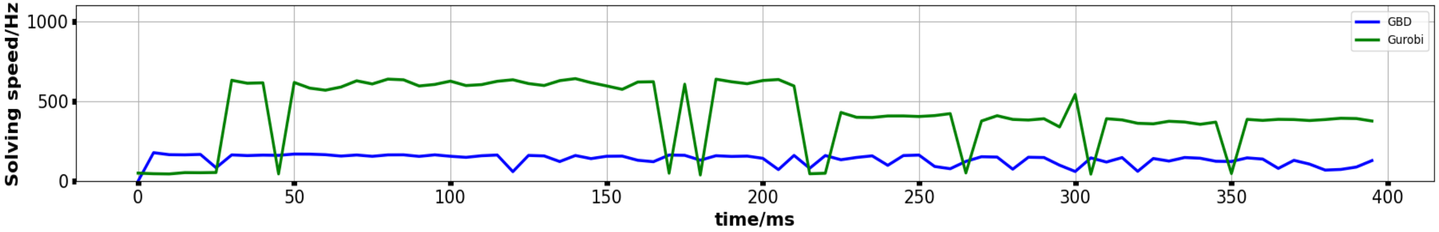
Stored feas. cuts: ~50
Stored opt. cuts: ~10

6 obstacles



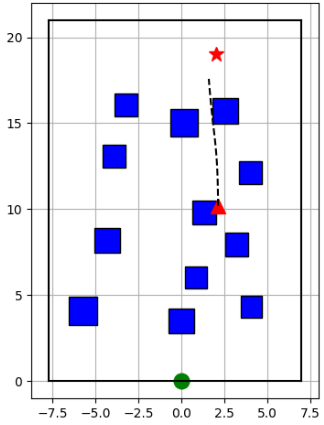
Stored feas. cuts: ~120
Stored opt. cuts: ~15

9 obstacles

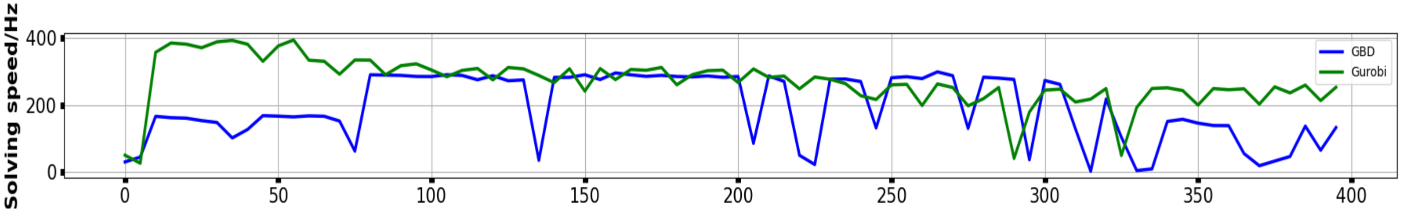


Stored feas. cuts: ~200
Stored opt. cuts: ~20

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition



12 obstacles



- GBD solver picks most of the feasibility cuts in the first few problems
- Better data efficiency than CoCo (Cauligi, Abhishek, et al)
- Speed competitive to Gurobi

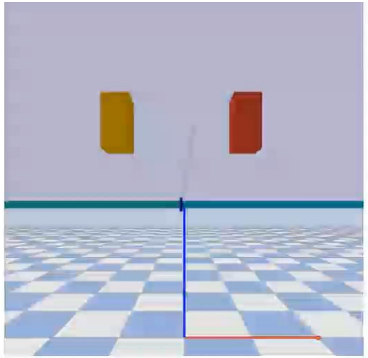
Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Cart-pole with moving soft walls

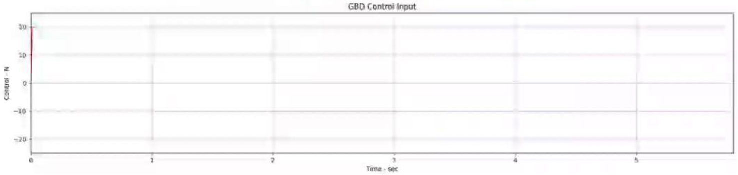
Mass $\text{cart} = 1$, $\text{cart} = 0.4$, $\text{pole length} = 0.6$
Random torque disturbance $\tau = N(0{,}8)$

Time horizon (N)	10	15
# of binaries	20	30

PyBullet Simulation Environment



Control input (N)



GBD Solving Speed (Hz)

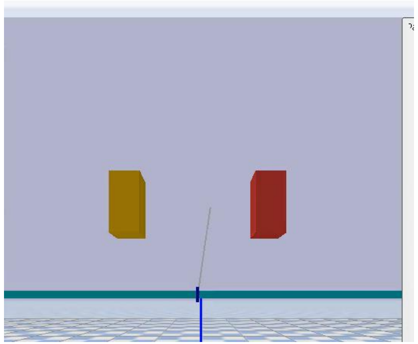
Cold-start

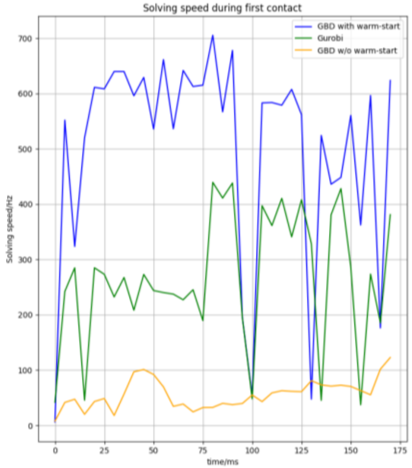
Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

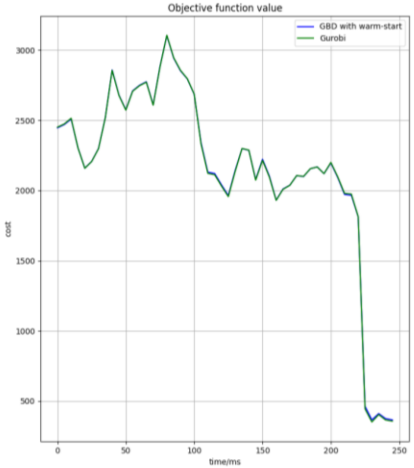
Solving speed during the first contact (N=10)

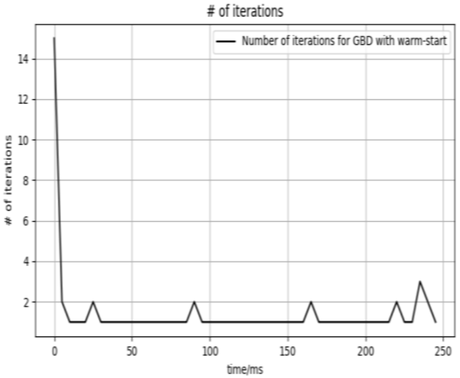
- GBD solver picks most of the feasibility cuts in the first problem

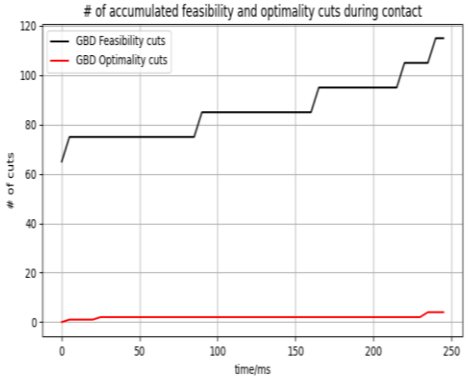
- Without warm-start, much slower









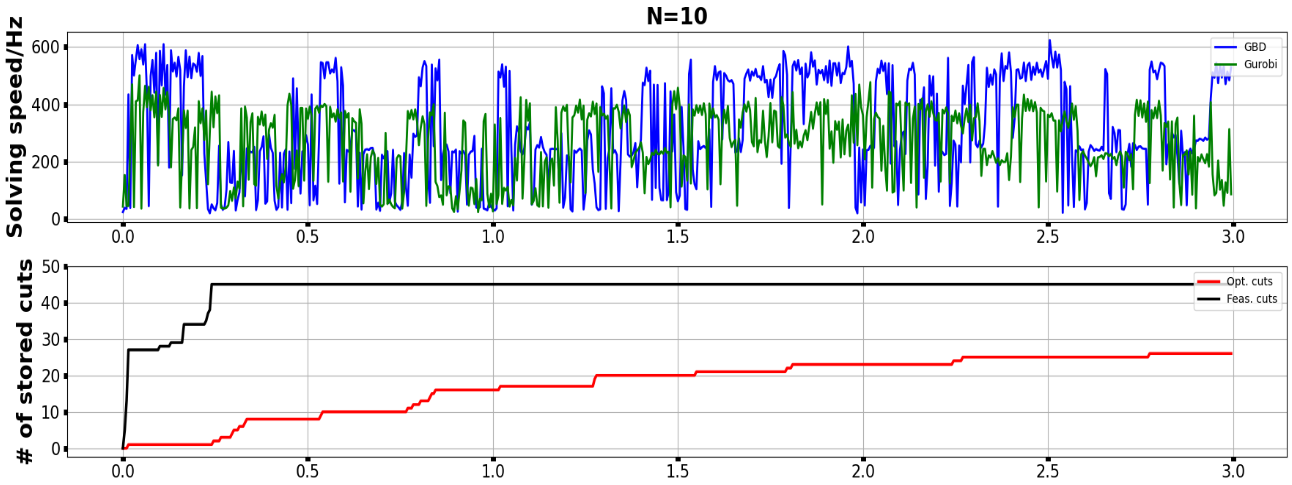


Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Solving speed during long time (N=10, 20 binaries)

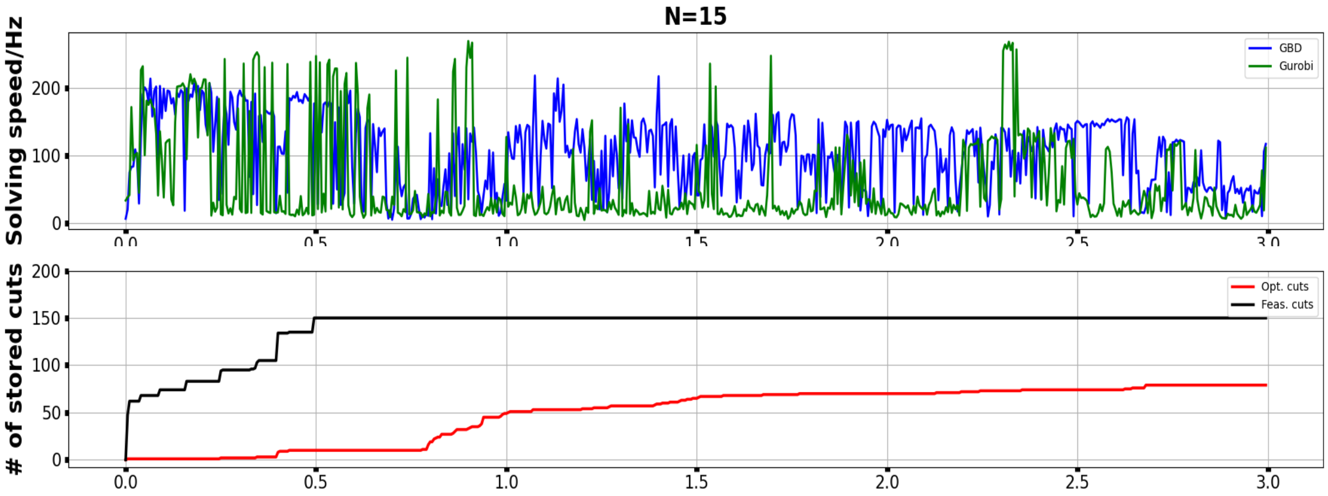
- Overall speed 2-3 times faster than Gurobi

- Better data efficiency than CoCo (Cauligi, Abhishek, et al)



Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Solving speed during long time (N=15, 30 binaries)



Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Real interesting things happen when this technique is combined with branching heuristics designed offline

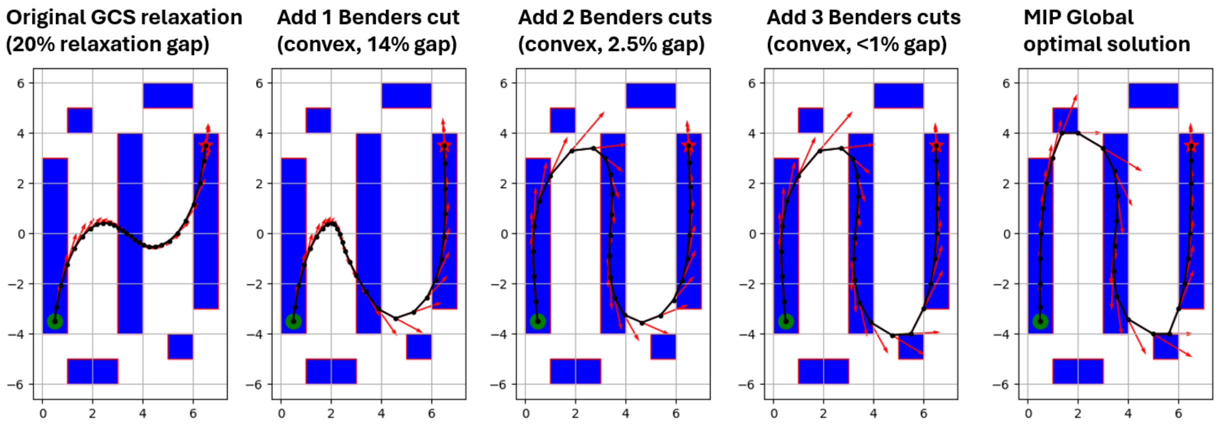
1. Branching heuristics designed offline helps to cold-start
2. This technique can be used with the relaxation-based methods (e.g. GCS)
3. Fast learning helps generalization to out-of-distribution cases

Fast and Continual Learning for Hybrid Control Policies using Generalized Benders Decomposition

Using Benders cuts with graph of convex sets

- Example taken from section 9.3 in your GCS paper [1])

- GCS formulation scales up well, Benders cuts quickly improve its optimality



[1] Marcucci, Tobia, et al. "Shortest paths in graphs of convex sets." *arXiv preprint arXiv:2101.11565* (2021).

END