Optimized Integrated Task and Motion Planning

Grigoriy Dubrovskiy

University of Notre Dame

May 2, 2017

Outline for Section 1

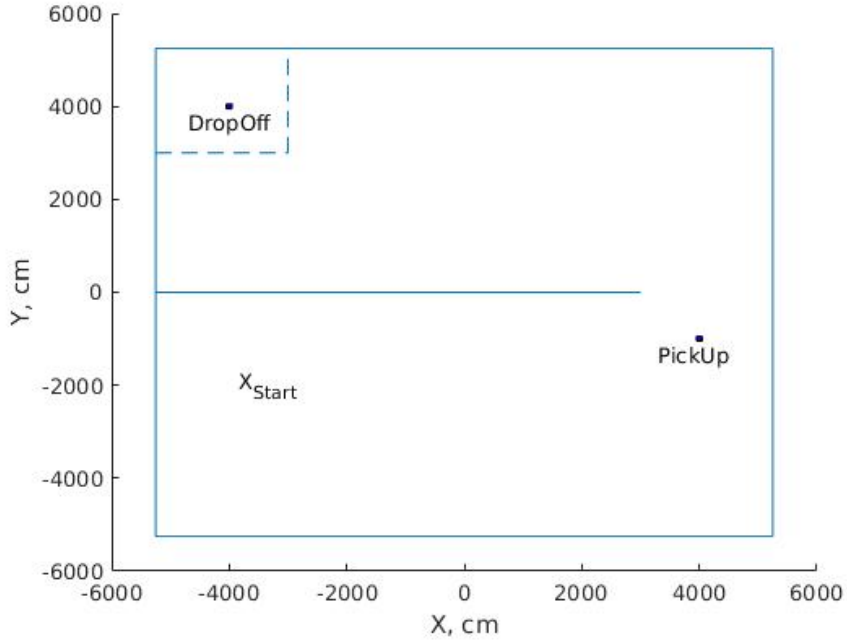
1 Optimized ITMP
- Problem at glance
- CLTLB(D) as MILP
- Results
- Possible extensions for Optimized ITMP

2 Related papers to Safe Autonomous Flight

3 Future Work

Problem at glance
Optimized ITMP

- **Integrated Task and Motion Planning**
- **Minimization of Manhattan norm**
- **Minimization of waypoints' number**
- Automatic generation of local-level plan (as waypoints) from a specification, which lead to accomplishment of a Global Mission



Specification for a robot *i*, formulated using Counter Linear Temporal Logic Over Constraint System CLTLB(D):

$$\phi_i \equiv PickUp_1 \ \mathcal{U} \ DropOff_1(-4000, 4000) \ \mathcal{U} \ GoHome$$

Optimized Motion Primitives

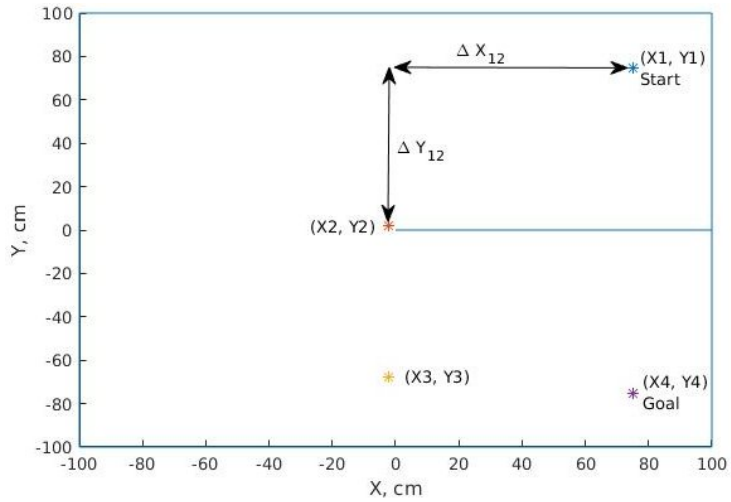
Primitive GoTo

$$\phi^{\mathcal{O}}_{GoTo} \equiv \forall j \in \mathcal{N}_{\mathcal{O}} : \Box\Big[(\pi = GoTo) \longrightarrow r^{j}_{left,o} \vee r^{j}_{right,o} \vee r^{j}_{below,o} \vee r^{j}_{above,o}\Big],$$

Whole specification for the primitive GoTo can be formulated as follows:

$$\phi_{GoTo} \equiv \Box\Big[\pi = GoTo \longrightarrow rin_x \wedge rin_y \wedge \phi^{\mathcal{O}}_{GoTo}\Big],$$

where π is a variable with a symbol of currently applied motion primitive; rin_x and rin_y are conditions to be inside the workspace.



Optimized Motion Primitives

Primitives' specifications

Specification for the primitive PickUp will look as follows:

$$\phi_{PickUp} \equiv \forall j \in \mathcal{N}_{\mathcal{B}} : \square\Big[\pi = PickUp_j \longrightarrow \bigwedge_{\forall l \in N_{\mathcal{B}}} \left(\neg q_b^l.p \wedge \bigcirc p_{carry}^{j,l}\right) \wedge r_{static} \wedge r_{object}^j\Big],$$

where $\neg q_b^l.p$ states that currently robot does not carrying any object from the set of objects $N_{\mathcal{B}}$; $\bigcirc p_{carry}^{j,l}$ means that robot will carry the object j; r_{static} - states of the robot will not change; r_{object}^j robot is posing in front of object j.

Using the same approach, we can formulate specifications for a primitive DropOff.

Optimized Motion Primitives

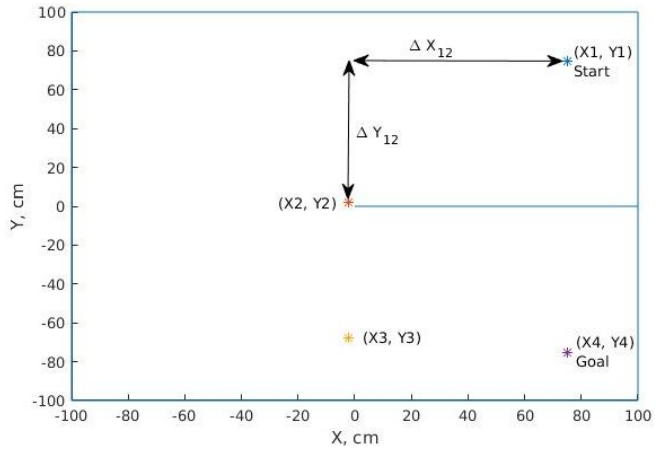
Primitive GoTo

Constraints:

$$r^j_{left,o} \equiv (\max(\bigcirc q_r.x, q_r.x) \leq \min(o_j.x_i, o_j.x_f) - \frac{a.l}{2}),$$
$$r^j_{right,o} \equiv (\max(\bigcirc q_r.x, q_r.x) \geq \max(o_j.y_i, o_j.y_f) + \frac{a.l}{2}),$$
$$r^j_{below,o} \equiv (\max(\bigcirc q_r.y, q_r.y) \leq \min(o_j.y_i, o_j.y_f) - \frac{a.l}{2}),$$
$$r^j_{above,o} \equiv (\max(\bigcirc q_r.y, q_r.y) \geq \max(o_j.y_i, o_j.y_f) + \frac{a.l}{2}),$$

$$\phi^{\mathcal{O}}_{GoTo} \equiv \forall j \in \mathcal{N}_{\mathcal{O}} : \Box\Big[(\pi = GoTo) \longrightarrow r^j_{left,o} \vee r^j_{right,o} \vee r^j_{below,o} \vee r^j_{above,o}\Big],$$

$$\textbf{Cost} = \text{Min} \sum_{k=1}^{n} \|\Delta d_{i,i+1}\|$$



Minimization of Manhattan norm for GoTo (for each step):

$$\text{Min} \|\Delta d_{i,i+1}\| = \text{Min} \sum_{m \in \{X,Y\}} |\Delta m_{i,i+1}| = \text{Min} \left(|\Delta X|_{i,i+1} + |\Delta Y|_{i,i+1} \right)$$

CLTLB(D) = Counter Linear Temporal Logic over Constraint System

Each state variable defined in the CLTLB(D) specifications are encoded as a vector of variables in the CPLEX solver.

For example, a robot's state is encoded as 3 vectors, corresponding to coordinates X, Y and to Angle theta (i.e. q_r.x, q_r.y, q_r.th for all time instances before K, which is a planning horizon).

Objects states' are encoded as a two dimensional array such that each element q_b[j].x is q_b[j].x[k], where j is an index of an object and k is a time instance.

Each motion primitives $\pi(k) \in Q_\pi$ is represented as an array such that each element is $\pi[k]$, because the values of which primitive to use should be determined during solving Optimization problem.

Specifications as MILP

Encoding specification of primitives as linear constraints.

Time operator $\bigcirc$ is encoded by adding or subtracting an array index, e.g. $\bigcirc q_r.x[k] \equiv q_r.x[k+1]$

Logical AND (y1): can be encoded using linear constraints*:

- y1 ≥ x1+x2-1
- y1≤x1
- y1≤x2
- 0≤y1≤1

where y1 is constrained to be an integer.

Logical NOT (y3): Use y3=1-x1, where y3 is constrained to be a binary integer.

Logical OR (y2): Use the linear constraints y2≤x1+x2, y2≥x1, y2≥x2, 0≤y2≤1, where y2 is constrained to be an integer.

Logical implication (y4): To express y4=(x1 $\Longrightarrow$ x2) (i.e., y4=¬x1∨ x2), we can adapt the construction for logical OR. In particular, use the linear constraints y4≤1-x1+x2, y4≥1-x1, y4≥x2, 0≤y4≤1, where y4 is constrained to be an integer.

* **https://cs.stackexchange.com/questions/12102/express-boolean-logic-operations-in-zero-one-integer-linear-progra**

Encoding specification of primitives as linear constraints

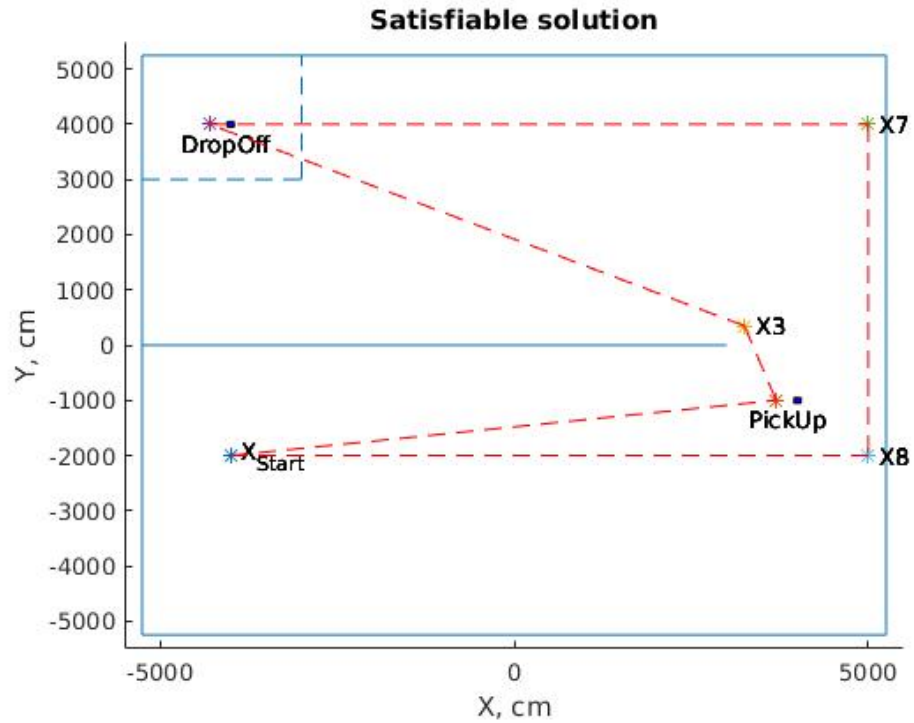
Encoding temporal logic quantifiers to first order logic requires quantifiers $\forall$ and $\exists$ in relation to the time instants.
The quantifier $\forall k \in N_p : \Psi[k]$ can be implemented using **for** loop.
The quantifier $\exists k \in N_p : \Psi[k]$ can be encoded by using an auxiliary variable j such as $\forall k \in N_p : (k = j) \implies \Psi[k] \wedge j \in N_p$ and, then, also implemented using a **for** loop.
Therefore, we can encode CLTLB(D) quantifiers as linear constraints to CPLEX:

- $\bigcirc^j \psi \Leftrightarrow j \in N_p \wedge \Psi[j]$
- $\psi_1 \mathcal{U} \psi_2 \Leftrightarrow \wedge_{\forall k \in N} \Big[(k < j \rightarrow \Psi_1[k]) \wedge (k = j \rightarrow \Psi_2[k]) \Big] \wedge j \in N_p$
- $\Box \psi \Leftrightarrow \wedge_{\forall k \in N} \Psi[k]$
- $\Diamond \psi \Leftrightarrow \wedge_{\forall k \in N} \Big[k = j \rightarrow \Psi[k] \Big] \wedge j \in N_p$
- $Last[\psi] \Leftrightarrow \Psi[K]$
- $\psi_1 \mathcal{U} (\psi_2 ... \mathcal{U} \psi_N) \Leftrightarrow \begin{cases} \wedge_{\forall k \in N_p} \Big[(k < j_1 \rightarrow \Psi[k]) \wedge \\ (j_1 \leq k < j_2 \rightarrow \Psi_2[k]) \wedge \\ ... \wedge (k = j_N \rightarrow \Psi_N[k]) \Big] \wedge \\ j_1, ..., j_N \in N_p \wedge j_1 < j_2 < ... < j_N \end{cases}$

Satisfiable solution from CPLEX

Objective = 0

```
Waypoints:
0.Initial      (-4000,-2000,0)
1.GoTo         (3700,-1000,0)
2.PickUp [1]   (3700,-1000,0)
3.GoTo         (3250,350,0)
4.GoTo         (-4300,4000,0)
5.DropOff [1]  (-4300,4000,0)
6.GoTo         (5000,4000,-0)
7.GoTo         (5000,-2000,0)
8.GoTo         (-4000,-2000,0)
9.GoTo         (-2464,-2000,0)
------------------------------
Other data:
0.Objects(0,0,0)
1.Objects(0,0,0)
2.Objects(1,0,0)
3.Objects(1,0,0)
4.Objects(1,0,0)
5.Objects(0,0,0)
6.Objects(0,0,0)
7.Objects(0,0,0)
8.Objects(0,0,0)
9.Objects(0,0,0)
```

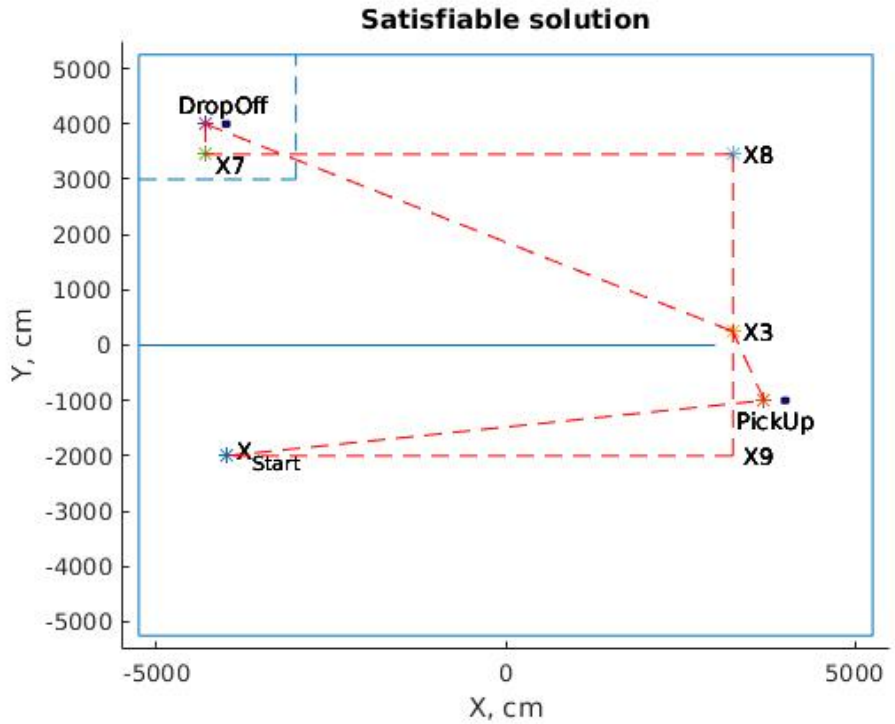


T = 4377.42ms

Satisfiable solution from Z3

Objective = 0

```
GoTo(3700,-1000,0)
PickUp(3700,-1000,0)
GoTo(3250,250,0)
GoTo(-4300,4000,0)
DropOff(-4300,4000,0)
GoTo(-4301,3450,0)
GoTo(3250,3450,0)
GoTo(3251,-1999,0)
GoTo(-4000,-2000,0)
```

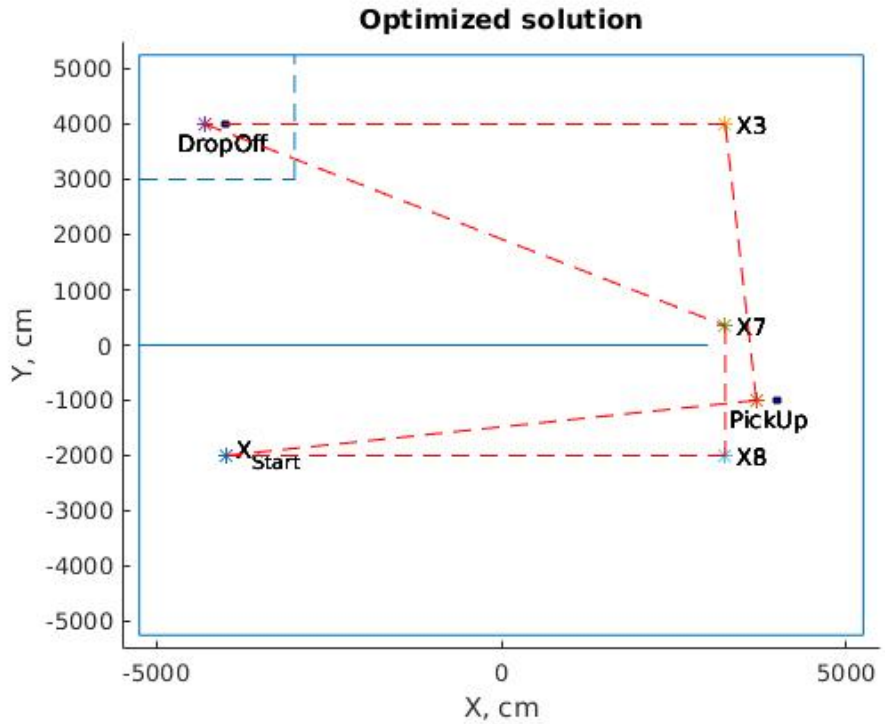


T = 699.384ms

Optimized solution wtih CPLEX

Objective = Min $\|\Delta d_{12}\|$

```
Waypoints:
0.Initial      (-4000,-2000,0)
1.GoTo         (3700,-1000,0)
2.PickUp [1]   (3700,-1000,0)
3.GoTo         (3250,4000,0)
4.GoTo         (-4300,4000,0)
5.DropOff [1]  (-4300,4000,0)
6.GoTo         (3250,350,0)
7.GoTo         (3250,-2000,0)
8.GoTo         (-4000,-2000,0)
9.GoTo         (-4000,-2000,0)
------------------------------
Other data:
0.Objects(0,0,0)
1.Objects(0,0,0)
2.Objects(1,0,0)
3.Objects(1,0,0)
4.Objects(1,0,0)
5.Objects(0,0,0)
6.Objects(0,0,0)
7.Objects(0,0,0)
8.Objects(0,0,0)
9.Objects(0,0,0)
```



T = 5246.05ms

Optimized solution with Z3?

Objective = Min $\|\Delta d_{12}\|$

T = ?? ms

CLTLB(D) = Counter Linear Temporal Logic over Constraint System

Definition (Semantics). The semantics of a CLTLB($\mathcal{D}$) formula ϕ at an instant $k \in \mathcal{N}_\rho$ is as follows:

- $\rho(k) \vDash p \iff p \vdash \rho(k)$.
- $\rho(k) \vDash R(\varphi_1, ..., \varphi_n) \iff R(\varphi_1, ..., \varphi_n) \vdash \rho(k)$.
- $\rho(k) \vDash \neg\phi \iff \rho(k) \nvDash \phi$.
- $\rho(k) \vDash \phi_1 \wedge \phi_2 \iff \rho(k) \vDash \phi_1 \wedge \rho(k) \vDash \phi_2$.
- $\rho(k) \vDash \bigcirc\phi \iff \rho(k+1) \vDash \phi$.
- $\rho(k) \vDash \bigcirc^{-1}\phi \iff \rho(k-1) \vDash \phi$.
- $\rho(k) \vDash \phi_1 \mathbf{U} \phi_2 \iff \begin{cases} \exists i \in [k, K] : \rho(i) \vDash \phi_2 \wedge \\ \forall j \in [k, i-1] : \rho(j) \vDash \phi_1 \end{cases}$.
- $\rho(k) \vDash \phi_1 \mathbf{S} \phi_2 \iff \begin{cases} \exists i \in [0, k] : \rho(i) \vDash \phi_2 \wedge \\ \forall j \in [i+1, k] : \rho(j) \vDash \phi_1 \end{cases}$.

$$\phi_i \equiv \mathit{PickUp}\big[1\big]\mathbf{U}\ \mathit{DropOff}\big[1, (-4000, 4000)\big]\mathbf{U}\ \mathit{GoHome}$$

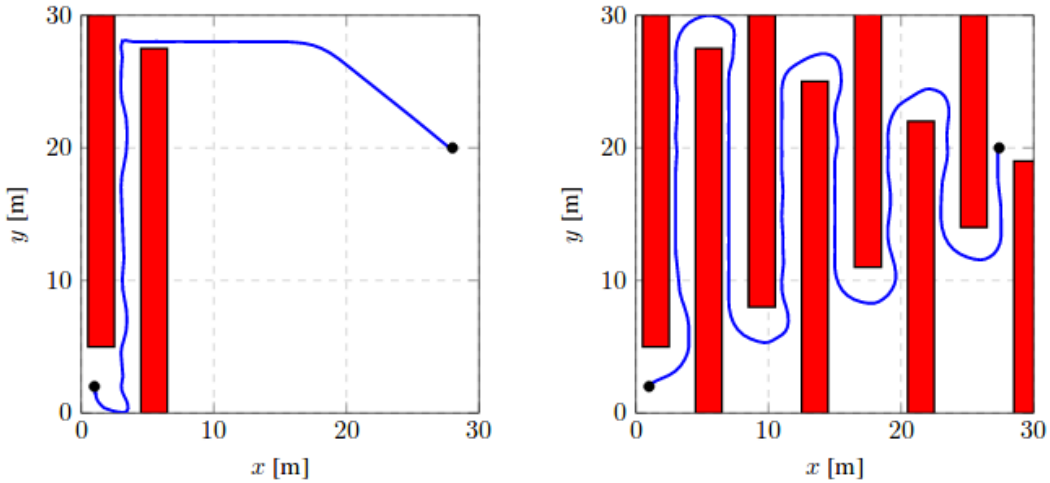
Here variables *i* were equal to 2, 5 and 9 (the robot PickUp at k=2, DropOff at k=5 and Home at k=9).

Possible extensions for Optimized ITMP

Possible future work:

- Compare with results from [”Scalable lazy SMT-based motion planning.” CDC 2016.];
- Compare with performance of SMT in optimization;
- Optimize Euclidean distance;
- Working on implementing more general specifications.

Compare with results from†



Number of passages	SMT-Based,Motion,Planner, [s]			RRT [s]	LTL OPT [s]
	Discrete abstraction	Dis-Plan	Con-Plan		
1	1.9975	0.1360	0.2542	10.4381	>7200
2	7.1461	1.1290	0.9294	122.3017	time out
3	19.3267	3.6495	1.0053	423.6957	time out
4	43.0985	4.0913	1.9204	1002.4193	time out

†**Shoukry, Yasser, et al. "Scalable lazy SMT-based motion planning." Decision and Control (CDC), 2016 IEEE 55th Conference on. IEEE, 2016..**

Compare performance of CPLEX with SMT optimization - joint work with Rafael

- SMT is using SAT to solve problems;
- CPLEX is using convex programming and solve via interior point methods.

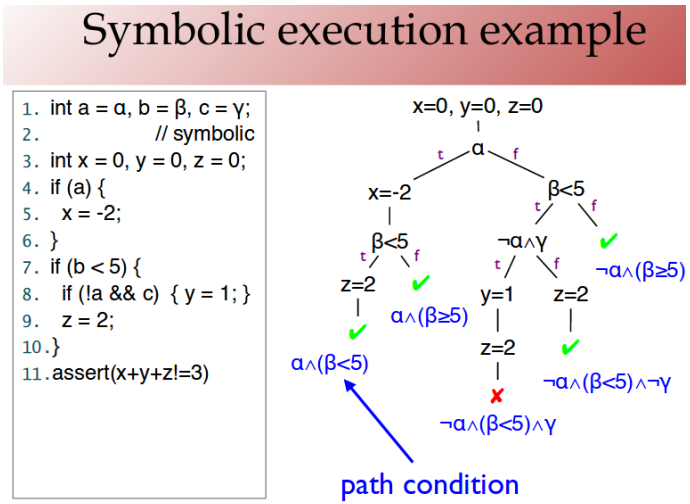


Figure : Symbolic Execution in SMT solvers

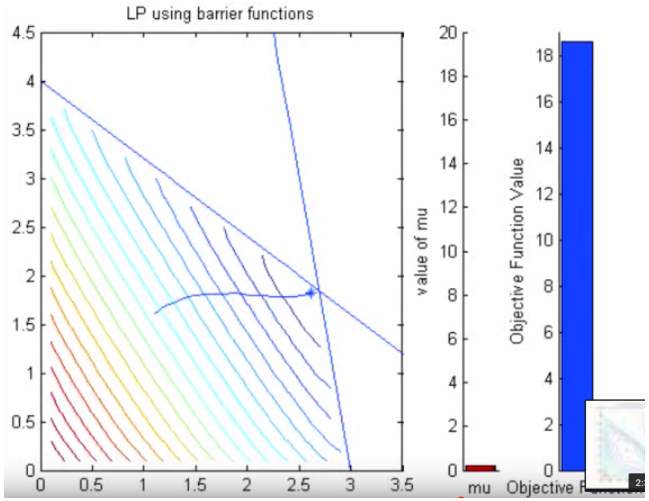


Figure : Linear Programming using barrier functions

Picture of SMT is from[‡]
Picture of Interior point method demonstration is from[§]

[‡] **http://www.seas.harvard.edu/courses/cs252/2011sp/slides/Lec13-SymExec.pdf**.
[§] **https://www.youtube.com/watch?v=MsgpSl5JRbI**.

Compare performance of CPLEX with SMT optimization

In MILP the ”bottleneck” is the logical constraints (e.g. AND, OR, Implication).

Logical AND (y1): Use the linear constraints[¶]:

- $y1 \geq x1+x2-1$
- $y1 \leq x1$
- $y1 \leq x2$
- $0 \leq y1 \leq 1$

where y1 is constrained to be an integer. This enforces the desired relationship.

Logical OR (y2): Use the linear constraints $y2 \leq x1+x2$, $y2 \geq x1$, $y2 \geq x2$, $0 \leq y2 \leq 1$, where y2 is constrained to be an integer.

Logical NOT (y3): Use $y3=1-x1$.

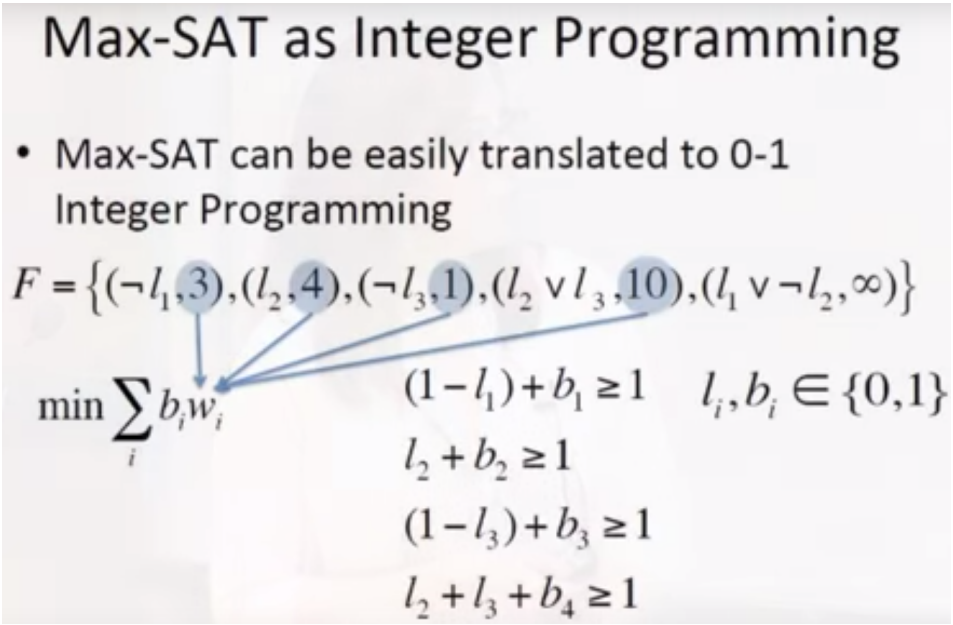
Logical implication (y4): To express $y4=(x1 \implies x2)$ (i.e., $y4=\neg x1 \vee x2$), we can adapt the construction for logical OR. In particular, use the linear constraints $y4 \leq 1-x1+x2$, $y4 \geq 1-x1$, $y4 \geq x2$, $0 \leq y4 \leq 1$, where y4 is constrained to be an integer.

[¶] **https://cs.stackexchange.com/questions/12102/express-boolean-logic-operations-in-zero-one-integer-linear-progra**

Compare performance of CPLEX with SMT optimization

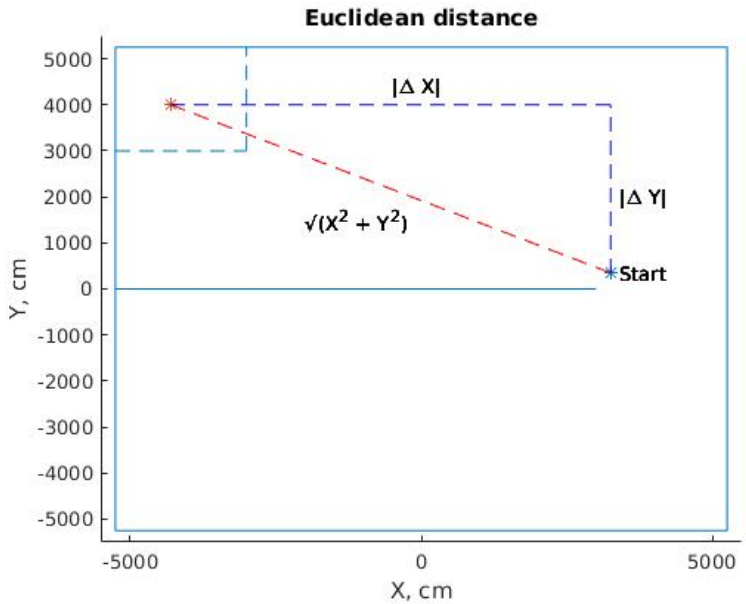
In SMT optimization the ”bottleneck” should be optimization of non-binary integer variables, because it doesn’t solve relaxation of the MILP, but search a satisfiable solution (and then try to improve the cost function by search?).

Are there guarantees on optimization solution? Can it return a local optimum for a centralized problem?



Optimize Euclidean distance

- Formulate objective as a convex quadratic function?
- Solve iteratively?



Working on implementing more general specifications

Implement the following operators (via using already implemented operators $\bigcirc, \mathcal{U}, \neg$):

$\Diamond$	Eventually	, which stands for $\neg \Box \neg \phi$
$\Box$	Always	, which stands for $true \mathcal{U} \phi$
$Last[\phi]$	Last	, which stands for $\Diamond(\neg \bigcirc true) \wedge \phi$, where $\neg true$ on finite trace is only true at last instant

Probabilistic ITMP

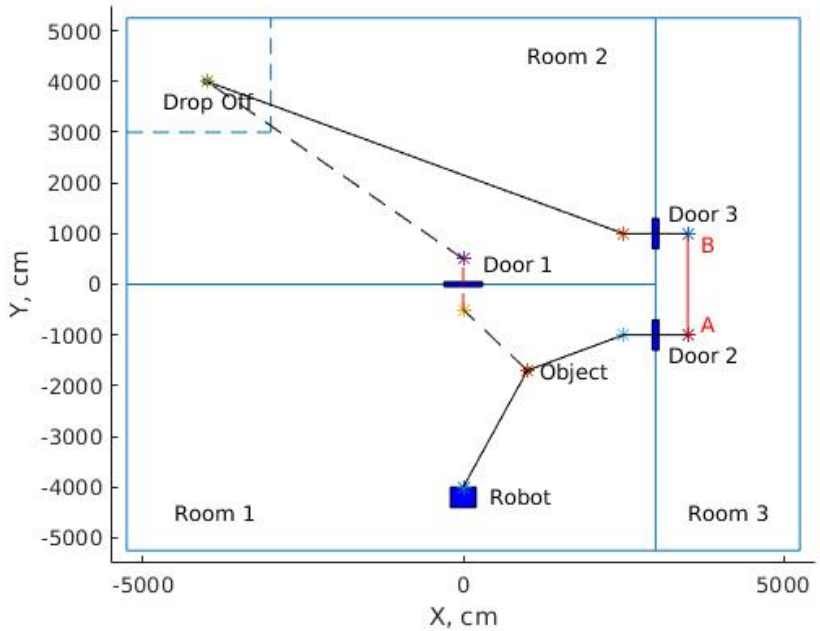
Maximize (*Prob of satisfying constraints*)

s.t.

$$\phi_{\mathcal{G}}^{Clean} \equiv PickUp_1 \ \mathcal{U} \ DropOff_1(-4000, 4000) \ \mathcal{U} \ GoHome$$

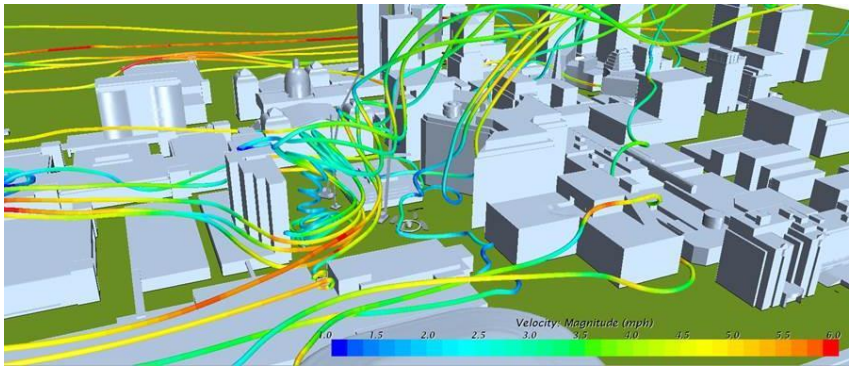
(Prob of satisfying constraints) $\geq 90\%$

$$P(Door1_{open} == true) = 20\%$$
$$P(Door2_{open} == true) = 80\%$$
$$P(Door3_{open} == true) = 80\%$$



Probabilistic ITMP

- **Maximize (probability) p of fulfilling all tasks before a deadline (e.g. P>90%)**
- Wind is a big deal: the speed of the drone is probabilistic (the real speed might be 2-7m/s, instead of nominal)



Picture is from[‖]

[‖] **Krishnakumar, Kalmanje S., et al. "Safe Autonomous Flight Environment (SAFE50) for the Notional Last "50 ft" of Operation of "55 lb" Class of UA**

Outline for Section 2

1 Optimized ITMP
- Problem at glance
- CLTLB(D) as MILP
- Results
- Possible extensions for Optimized ITMP

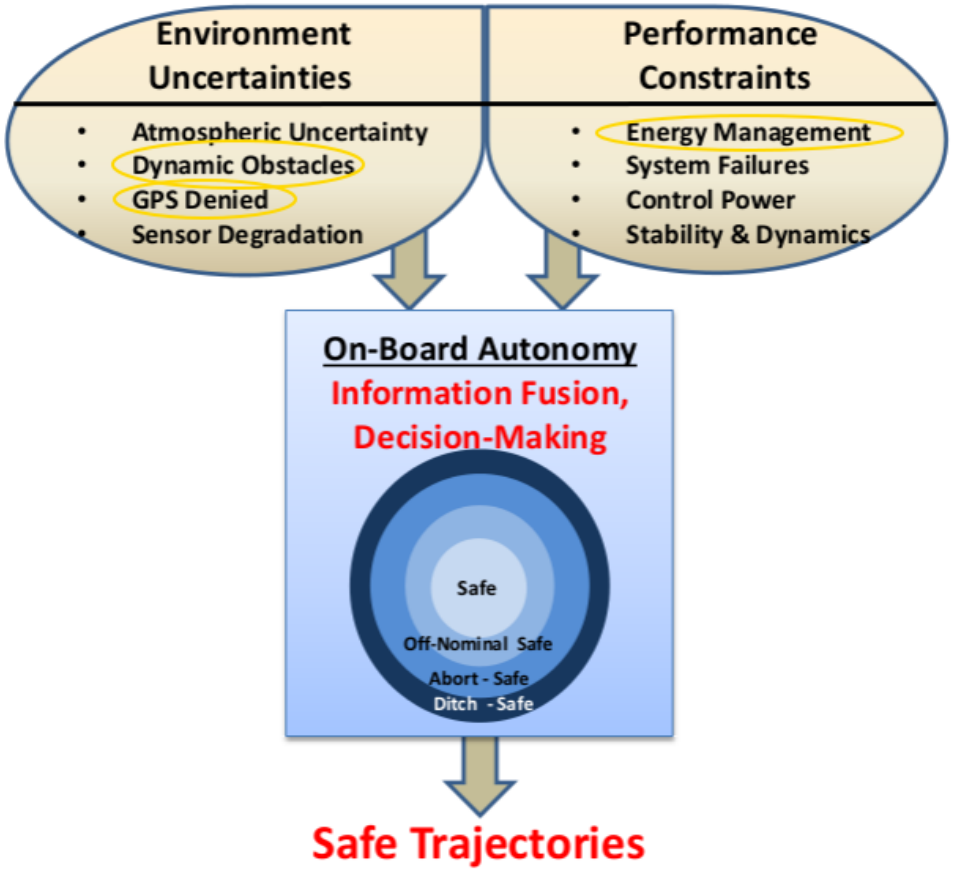
2 Related papers to Safe Autonomous Flight

3 Future Work

Unmanned Aerial Systems Applications

[Safe and Autonomous Drones for Urban Flight, presentation from NASA, Oct 2016]

Safe & Autonomous for Urban Flight



[Safe and Autonomous Drones for Urban Flight, presentation from NASA, Oct 2016]

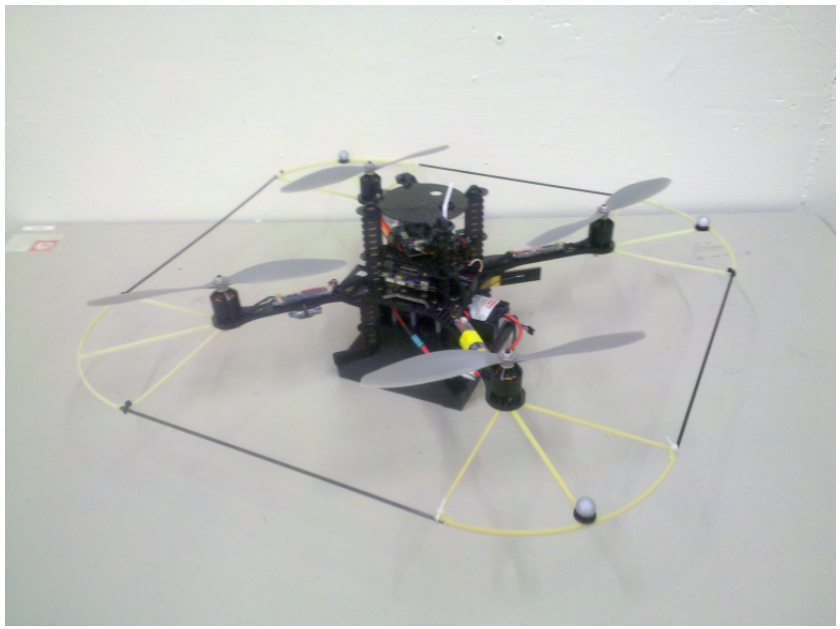
We could improve safety in GPS denied environments and against Dynamic Obstacles. And work optimizing Energy Management through thoughtful planning (in ITMP).

”Reinforcement Learning in Robotics: A Survey”, (2013, J.Kober (Bielefeld Univ.), J. Andrew Bagnell (CMU), Jan Peters(Max Planck Institute for Intelligent Systems))

From a survey paper in Reinforcement Learning:
”**... To apply reinforcement learning in robotics, safe exploration becomes a key issue of the learning process ..., a problem often neglected in the general reinforcement learning community. ...**”

Even if we can describe safety or performance in a Temporal Logic formulas, we need guarantees that such a ”safe” control signal might always be generated.

"Reducing Conservativeness in Safety Guarantees by Learning Disturbances Online: Iterated Guaranteed Safe Online Learning", (2014, J.H.Gillula and C.J. Tomlin)



They combined Reinforcement Learning with a Hamilton-Jacobi-Isaacs (HJI) reachability analysis to perform altitude tracking, while guaranteeing safety.

"Reducing Conservativeness in Safety Guarantees by Learning Disturbances Online: Iterated Guaranteed Safe Online Learning", (2014, J.H.Gillula and C.J. Tomlin)

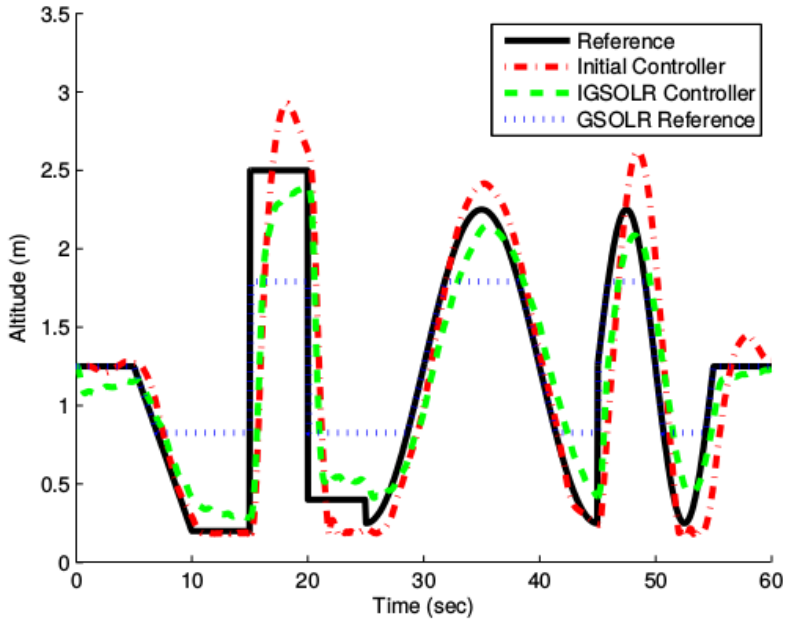


Figure : Sample trajectory of the quadrotor's altitude using the hand-tuned controller versus the IGSOLR controller.

"Reducing Conservativeness in Safety Guarantees by Learning Disturbances Online: Iterated Guaranteed Safe Online Learning", (2014, J.H.Gillula and C.J. Tomlin)

"One approach to overcoming this limitation ** **is to wrap the controller being learned inside another controller that seeks to guarantee safety;** this is the approach taken by Gillula and Tomlin [††], in which we proposed a framework known as Guaranteed Safe Online Learning via Reachability (GSOLR). **This framework combines machine learning techniques with Hamilton-Jacobi-Isaacs (HJI) reachability analysis in a way that prevents the system being controlled from taking an unsafe action, while avoiding any changes to the machine learning algorithm being used** [a paper on GSOLR].

In essence, GSOLR is a least-restrictive safe controller: it allows the machine learning algorithm to control the system (e.g. via RL) except when the state is detected as being near a safety threshold, at which time a safety-guaranteeing control (dictated by HJI reachability analysis) is used instead."

** **safety during learning**.

†† **Gillulay, Jeremy H.,
and Claire J. Tomlin. "Guaranteed safe online learning of a bounded system." Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE, 2011.**.

”Preliminary Results of Powerline Reconstruction From Airborne LiDAR for Safe Autonomous Low-Altitude Urban Operations of Small UAS ”, (2014, C.Ippolito, K. Krishnakumar (NASA Ames Research Center), S.Hening(University of Santa Cruz))

NASA researchers use LiDAR for detecting Powerlines.

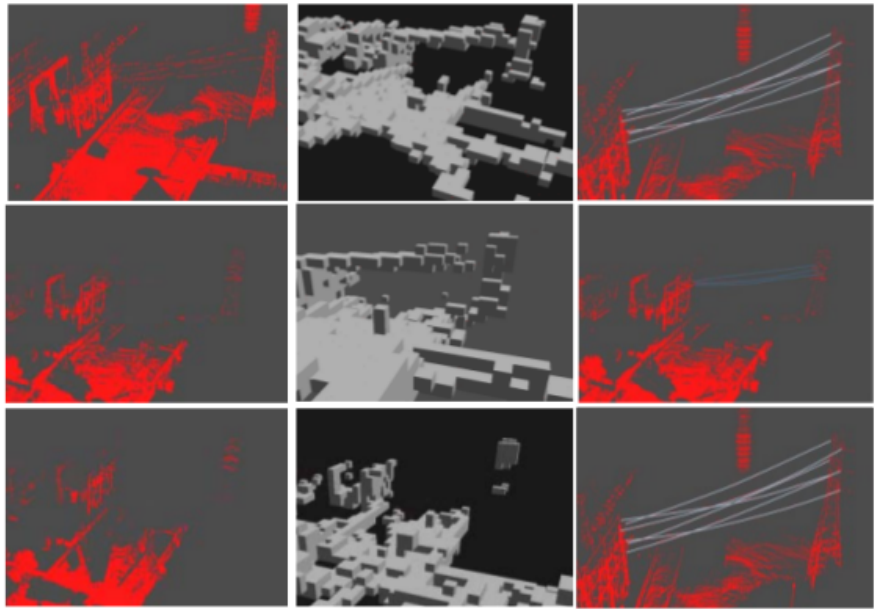


Figure : Powerline reconstruction of aircraft LiDAR data. Raw LiDAR point clouds (left), voxel processing (middle), reconstructed powerlines (right), at 20m (top), 40m (middle) and 60m (bottom).

Outline for Section 3

1 Optimized ITMP
- Problem at glance
- CLTLB(D) as MILP
- Results
- Possible extensions for Optimized ITMP

2 Related papers to Safe Autonomous Flight

3 Future Work

Future Work

- Explore other options for guaranteeing Safe Learning;
- Explore possibility of using Reachability analysis (and other model-based approaches) with LiDAR data for guaranteeing safety during a flight;
- Learn how to use a model-free Reinforcement Learning algorithm: ”Policy Gradient with the Signed Derivative”;
- Work on extensions for Optimized ITMP?