Formal Approaches to Mission Planning using Temporal Logics

Sertac Karaman[1]

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

MACCCS Review, UMich, 2008

[1] joint work with Emilio Frazzoli, Ricardo Sanfelice, Amit Bhatia, Michelangelo Graziano, Roberto Naldi

Massachusetts Institute of Technology

Outline

Massachusetts Institute of Technology

Quick Introduction to Linear Temporal Logic

Linear Temporal Logic is an extension of classical propositional logic

Extends classical operators

NOT ($\neg P$), **AND** ($P \wedge Q$), **OR** ($P \vee Q$), **IMPLICATION** ($P \rightarrow Q$) with

- **EVENTUALLY** ($\Diamond P$) : Proposition P will eventually be true at some future time.
- **ALWAYS** ($\Box P$) : Proposition P will always be true throughout the future
- **UNTIL** ($P \mathcal{U} Q$) : P will hold to be until Q becomes true
- **UNLESS** ($P \mathcal{W} Q$) : P must be true unless

Quick Introduction to Linear Temporal Logic

Linear Temporal Logic is an extension of classical propositional logic
Extends classical operators
NOT ($\neg P$), **AND** ($P \wedge Q$), **OR** ($P \vee Q$), **IMPLICATION** ($P \rightarrow Q$) with

EVENTUALLY ($\Diamond P$) : Proposition P will eventually be true at some future time.

ALWAYS ($\Box P$) : Proposition P will always be true throughout the future

UNTIL ($P \mathcal{U} Q$) : P will hold to be until Q becomes true

UNLESS ($P \mathcal{W} Q$) : P must be true unless

Advantages of applications in mission planning

* LTL is remarkably close to natural language
* studied for several years by philosophers computer scientists
* fits quite well into a Vehicle Routing setting

Massachusetts Institute of Technology

Mission Planning Problems and LTL

Some examples of reasoning using Linear Temporal Logic

Safety

A SAM Site should always not be engaged

$\Box \neg \mathit{SAMSite}$

Reachability

A target should be monitored eventually

$\Diamond \mathit{Target}1$

Order

An event should happen unless an other one occurs

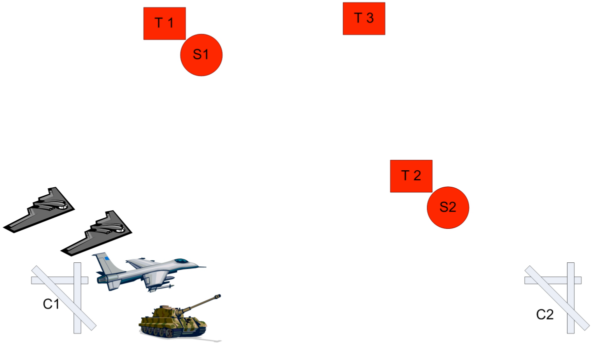
$(\neg \mathit{Target}1)\,\mathcal{W}\,\mathit{Target}2$

More complicated examples can be built using the operators of classical logic

From Mission Planning with LTL Specifications to MILP

- For any LTL formula we present Mixed-integer Linear Constraints that are satisfied if and only if the formula is satisfied
- These constraints can be merged with slightly modified versions of MILP based formulations of mission planning problems

Massachusetts Institute of Technology

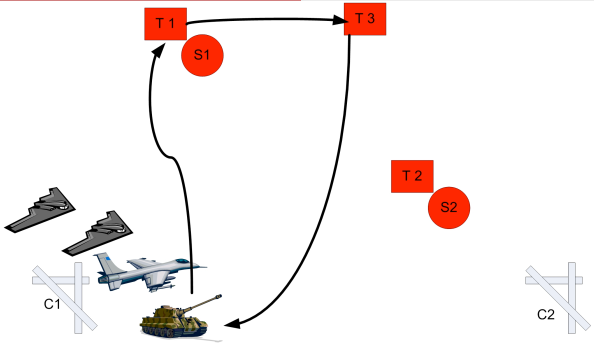


Mission Specifications

- Vehicles travel with $V1 : 25mph$, $V2 : 25mph$, $V3 : 40mph$, $V4 : 12mph$
- Mission is to either destroy T1 and T3 or T2 and T3. If T2 is destroyed then take V4 to C2.
- T1 and T2 are protected by SAMs S1 and S2. V1 is vulnerable to S1 where as V2 is vulnerable to both S1 and S2. V3 can not engage T1 and T2.

Several mission specifications can be represented using Linear Temporal Logics, e.g.,

$$\Diamond(T1 \vee T2) \wedge \Diamond T3 \qquad \neg(V1@T1 \vee V2@T1)\mathcal{W}SAM1 \qquad \Box\neg V2@T1$$



Mission Specifications

- Vehicles travel with $V1 : 25mph$, $V2 : 25mph$, $V3 : 40mph$, $V4 : 12mph$
- Mission is to either destroy T1 and T3 or T2 and T3. If T2 is destroyed then take V4 to C2.
- T1 and T2 are protected by SAMs S1 and S2. V1 is vulnerable to S1 where as V2 is vulnerable to both S1 and S2. V3 can not engage T1 and T2.

Solution of the mission: minimize the total time that UAVs were employed

The solution employs a single vehicle considering the risk factors.
Optimal solution changes in structure for slight changes in the mission.

A closer look at the cost function of the optimization problem

We have minimized the **total amount of time** that assets were employed, i.e.,

$f := \sum_{k=1}^{K} r_k t_k$ where t_k : time that asset k finishes the mission
r_k : relative risk coefficient of an asset

- Using an extra vehicle is of high risk.
- Optimal solution is generally to use small number of vehicles effectively

One can minimize the **mission time**, i.e.,

$f := t_{max}$ subject to $t_k \leq t_{max}$ for $k \in \{1, \ldots, K\}$ (t_{max} is the mission time)

- An extra vehicle can be employed as long as the mission time is not increased
- Optimal solution employs as many vehicles as possible to minimize the mission time

Massachusetts Institute of Technology

A closer look at the cost function of the optimization problem

We have minimized the **total amount of time** that assets were employed, i.e.,

$f := \sum_{k=1}^{K} r_k t_k$ where

t_k	: time that asset k finishes the mission
r_k	: relative risk coefficient of an asset

- Using an extra vehicle is of high risk.
- Optimal solution is generally to use small number of vehicles effectively

One can minimize the **mission time**, i.e.,

$f := t_{max}$ subject to $t_k \leq t_{max}$ for $k \in \{1, \ldots, K\}$ (t_{max} is the mission time)

- An extra vehicle can be employed as long as the mission time is not increased
- Optimal solution employs as many vehicles as possible to minimize the mission time

Massachusetts Institute of Technology

A closer look at the cost function of the optimization problem

We have minimized the **total amount of time** that assets were employed, i.e.,

$f := \sum_{k=1}^{K} r_k t_k$ where

t_k : time that asset k finishes the mission
r_k : relative risk coefficient of an asset

- Using an extra vehicle is of high risk.
- Optimal solution is generally to use small number of vehicles effectively

One can minimize the **mission time**, i.e.,

$f := t_{max}$ subject to $t_k \leq t_{max}$ for $k \in \{1, \ldots, K\}$ (t_{max} is the mission time)

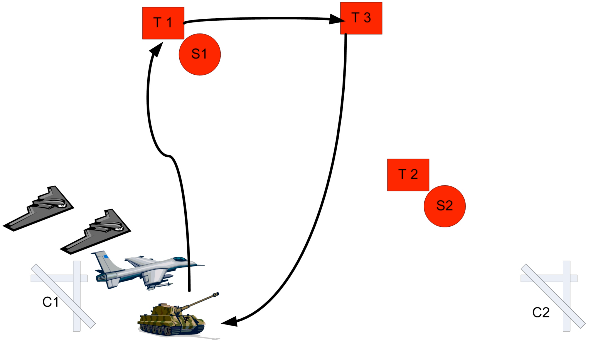
- An extra vehicle can be employed as long as the mission time is not increased
- Optimal solution employs as many vehicles as possible to minimize the mission time

Can we use a mixture of the two?

Let cost function be a convex combination of the two, i.e.,

$$f := \alpha(\textstyle\sum_{k=1}^{K} r_k t_k) + (1 - \alpha) t_{max} \quad \text{subject to} \quad t_k \leq t_{max} \text{ for } k \in \{1, \ldots, K\}$$

- α becomes a "knob" which can be tuned for desired performance (human supervision).
- $\alpha \to 1$: generate more **risk averse** solutions, employ few vehicles, do not worry about time
- $\alpha \to 0$: look for more **cooperative** solutions, get the whole mission done in minimum time



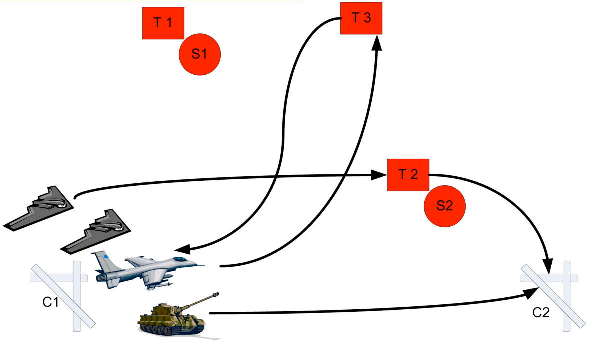
Mission Specifications

- Vehicles travel with $V1 : 25mph$, $V2 : 25mph$, $V3 : 40mph$, $V4 : 12mph$
- Mission is to either destroy T1 and T3 or T2 and T3. If T2 is destroyed then take V4 to C2.
- T1 and T2 are protected by SAMs S1 and S2. V1 is vulnerable to S1 where as V2 is vulnerable to both S1 and S2. V3 can not engage T1 and T2.

Minimum risk solution of the mission

The solution employs a single vehicle considering the **risk factors**. The mission time is not the best possible (using the slowest vehicle to do all the job by itself).

Total Time: 1.92h, **Mission Time:** 1.92h



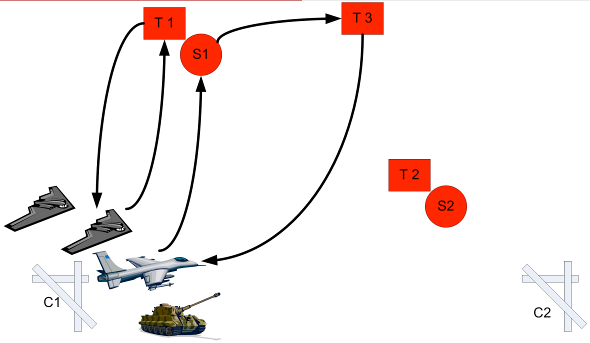
Mission Specifications

- Vehicles travel with $V1 : 25mph$, $V2 : 25mph$, $V3 : 40mph$, $V4 : 12mph$
- Mission is to either destroy T1 and T3 or T2 and T3. If T2 is destroyed then take V4 to C2.
- T1 and T2 are protected by SAMs S1 and S2. V1 is vulnerable to S1 where as V2 is vulnerable to both S1 and S2. V3 can not engage T1 and T2.

Minimum time solution of the mission

The optimal solution employs as many assets as needed to complete the mission in **minimum time**. **High risk** is taken since all the assets can be lost if mission fails.

Total Time: 2.62h, **Mission Time:** 0.94h



Mission Specifications

- Vehicles travel with $V1 : 25mph$, $V2 : 25mph$, $V3 : 40mph$, $V4 : 12mph$
- Mission is to either destroy T1 and T3 or T2 and T3. If T2 is destroyed then take V4 to C2.
- T1 and T2 are protected by SAMs S1 and S2. V1 is vulnerable to S1 where as V2 is vulnerable to both S1 and S2. V3 can not engage T1 and T2.

Solution for $\alpha = 0.5$

Only two vehicles are employed. Mission time is between the two extremes.

Total Time: 2.0h, **Mission Time:** 1.1h

Best Total Time: 1.92h (up to 2.62h), Best Mission Time: 0.94h (up to 1.92h)

Limitations of Linear Temporal Logic

LTL reasons about **qualitative** properties of time. **Quantitative** properties can not be expressed!

- **Safety** : After 30 minutes from the mission start always do not engage with Target
  Perhaps it will be too late - enemy has gathered more units near the target
- **Reachability**: Within the first 50 minutes eventually destroy the target
  The target must be destroyed and this must be done in first 50 minutes
- **Order**: Between 20 and 50 minutes do not engage target 1 unless SAM site is destroyed
  If target is to be destroyed in this interval then SAM site should have been destroyed first

Massachusetts Institute of Technology

Limitations of Linear Temporal Logic

LTL reasons about **qualitative** properties of time. **Quantitative** properties can not be expressed!

- **Safety** : After 30 minutes from the mission start always do not engage with Target
  Perhaps it will be too late - enemy has gathered more units near the target
- **Reachability**: Within the first 50 minutes eventually destroy the target
  The target must be destroyed and this must be done in first 50 minutes
- **Order**: Between 20 and 50 minutes do not engage target 1 unless SAM site is destroyed
  If target is to be destroyed in this interval then SAM site should have been destroyed first

Metric Temporal Logic (MTL)

- Metric Temporal Logic is one of the many real time logics
- MTL is based on bounded operators
- extends all the temporal operators with real intervals. Let $\mathcal{I}$ be any interval of $\mathbb{R}$
  - **Bounded Eventually** ($\Diamond_{\mathcal{I}} P$)
  - **Bounded Always** ($\Box_{\mathcal{I}} P$)
  - **Bounded Until** ($P \mathcal{U}_{\mathcal{I}} Q$)
  - **Bounded Unless** ($P \mathcal{W}_{\mathcal{I}} Q$)

Massachusetts Institute of Technology

Limitations of Linear Temporal Logic

LTL reasons about **qualitative** properties of time. **Quantitative** properties can not be expressed!

- **Safety** : After 30 minutes from the mission start always do not engage with Target
  Perhaps it will be too late - enemy has gathered more units near the target
- **Reachability**: Within the first 50 minutes eventually destroy the target
  The target must be destroyed and this must be done in first 50 minutes
- **Order**: Between 20 and 50 minutes do not engage target 1 unless SAM site is destroyed
  If target is to be destroyed in this interval then SAM site should have been destroyed first

Metric Temporal Logic (MTL)

- Metric Temporal Logic is one of the many real time logics
- MTL is based on bounded operators
- extends all the temporal operators with real intervals. Let $\mathcal{I}$ be any interval of $\mathbb{R}$
  - **Bounded Eventually** ($\Diamond_{\mathcal{I}} P$)
  - **Bounded Always** ($\Box_{\mathcal{I}} P$)
  - **Bounded Until** ($P \mathcal{U}_{\mathcal{I}} Q$)
  - **Bounded Unless** ($P \mathcal{W}_{\mathcal{I}} Q$)

Examples of reasoning using MTL

- **Bounded Safety** : $\Box_{[30,\infty]} \neg Target$
- **Bounded Reachability**: $\Diamond_{[0,50]} Target$
- **Bounded Order**: $(\neg Target) \mathcal{W}_{[20,50]} SAMSite$

Massachusetts Institute of Technology

Mission Planning Problems with MTL Specifications

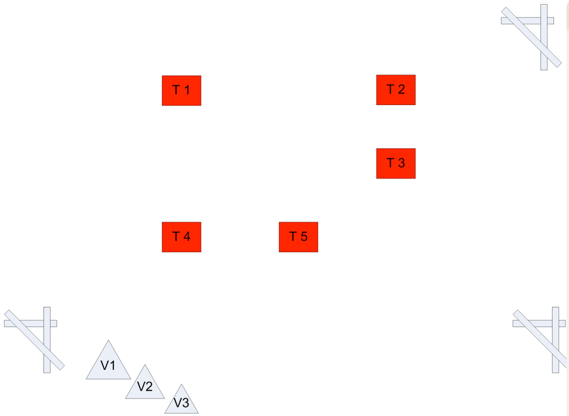
Advantages of Metric Temporal Logics in a Vehicle Routing setting

- quantitative properties are crucial in mission planning
- MTL is quite close to natural language (unlike most other real-time logics)
- MTL is as expressive as other real-time propositional temporal logics

Mission Planning with MTL Specifications to MILP

- If the temporal operators are only applied to atomic propositions or their negations then we provide a set of mixed-integer linear constraints that are satisfied if and only if the formula is satisfied
- These constraints can be merged with the existing MILP-based formulations of mission planning

Massachusetts Institute of Technology

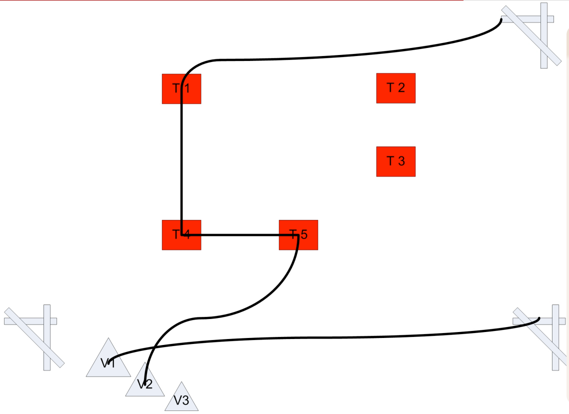


Mission objectives

- Vehicles *V*1, *V*2, and *V*3 have speeds 15, 18, and 20 mph
- service T2, T3 in 0.7 hours or service T1, T4, T5 in 1.5 hours
- if former option then eventually V1@C2 and no landing on C2 before N4 is destroyed
- T2 must be serviced by V2 and do not service T4 in the first 0.4 hours
- If latter option then do not service T2 in first 0.6 hours
- Do not service T3 unless T2 is serviced

Metric Temporal Logics

$$(\Diamond_{[0,1.5]} T_1 \wedge \Diamond_{[0,1.5]} V_2@T_4 \wedge \Diamond_{[0,1.5]} T_5 \wedge \Diamond_{[0,1.5]} V_1@C_2 \wedge \Box_{[0,0.4]}\neg V_2@T_4 \wedge (\neg V_1@C_2)\mathcal{U} V_2@T_4) \vee (\Diamond_{[0,0.7]} V_2@T_2 \wedge (\neg T_2)\mathcal{U}_{[0,0.7]} V_3@T_3 \wedge \Box_{[0,0.6]}\neg T_2)$$



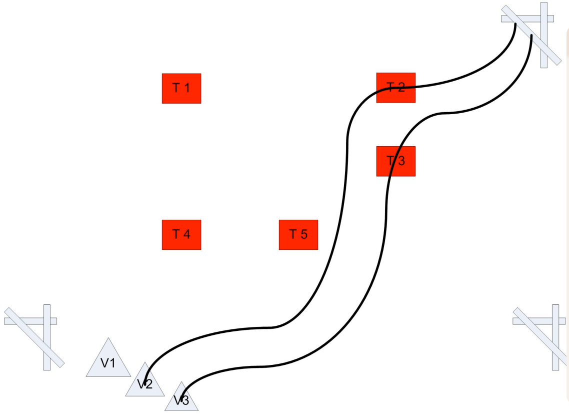
Mission objectives

- service T2, T3 in 0.7 hours or service T1, T4, T5 in 1.5 hours
- if former option then eventually V1@C2 and no landing on C2 before N4 is destroyed
- T2 must be serviced by V2 and do not service T4 in the first 0.4 hours
- If latter option then do not service T2 in first 0.6 hours
- Do not service T3 unless T2 is serviced

Solution: minimizing the risk

- V1 launches after 0.05 hours. $t_{T5} = 0.32$, $t_{T4} = 0.43$, and $t_{T1} = 0.59$

Massachusetts Institute of Technology



Mission objectives

- service T2, T3 in 0.7 hours or service T1, T4, T5 in 1.5 hours
- if former option then eventually V1@C2 and no landing on C2 before N4 is destroyed
- T2 must be serviced by V2 and do not service T4 in the first 0.4 hours
- If latter option then do not service T2 in first 0.6 hours
- Do not service T3 unless T2 is serviced

Old solution

- V1 launches after 0.05 hours. $t_{T5} = 0.32$, $t_{T4} = 0.43$, and $t_{T1} = 0.59$

New solution : slightly modified scenario

- T4 can not be serviced within first 0.5 hours
- V2 starts after 0.11 hours. $t_{T2} = 0.6$ and $t_{T3} = 0.4$

How general is our MILP formulation?

- For many years, LTL has been used as a specification language in AI planning
- Recently, LTL has also been employed in control theory to synthesize controllers
- Most of these control techniques employ *model checking* based techniques
- We have also been doing planning: among all the possible plans we pick the **optimal**
- Is it possible to reverse this process allowing us to propose new model checking algorithms?

Massachusetts Institute of Technology

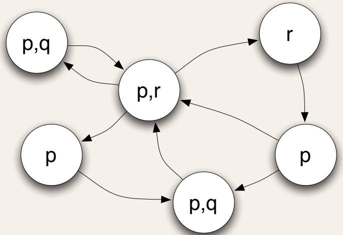
How general is our MILP formulation?

- For many years, LTL has been used as a specification language in AI planning
- Recently, LTL has also been employed in control theory to synthesize controllers
- Most of these control techniques employ *model checking* based techniques
- We have also been doing planning: among all the possible plans we pick the **optimal**
- Is it possible to reverse this process allowing us to propose new model checking algorithms?

Quick Introduction to Model Checking

Problem Definition : (*Clarke et.al, Model Checking '99*)

Given a Transition System (states, transitions, and labeling) and a temporal logic formula ϕ expressing some desired specification, find the set of all states that satisfy ϕ



Model Checking

- Many applications in debugging concurrent algorithms
- Several model checker software available, e.g., SPIN and NuSMV
  - Given several programs executing concurrently and a formula, model checker gives the set of all initial conditions for the programs to satisfy the properties
    - formula can be a **safety** property (always avoid infinite loops or deadlocks),
    - a **reachability** condition (all programs eventually terminate),
    - an **ordering** property (program 1 completes before program 2 prints its message)
- **Global Model Checking**: Find all the states that satisfy ϕ
- **Local Model Checking**: Find out whether a given specific state satisfies ϕ

Massachusetts Institute of Technology

Model Checking

- Many applications in debugging concurrent algorithms
- Several model checker software available, e.g., SPIN and NuSMV
  - Given several programs executing concurrently and a formula, model checker gives the set of all initial conditions for the programs to satisfy the properties
    - formula can be a **safety** property (always avoid infinite loops or deadlocks),
    - a **reachability** condition (all programs eventually terminate),
    - an **ordering** property (program 1 completes before program 2 prints its message)
- **Global Model Checking**: Find all the states that satisfy ϕ
- **Local Model Checking**: Find out whether a given specific state satisfies ϕ

Model Checking Dynamical Systems

Problem Definition

Given a dynamical system and a temporal logic formula ϕ find all the initial states of the dynamical system that satisfy the formula ϕ

(atomic propositions are polytopes in state space)

- model checking controllable linear systems is decidable! (Pappas'03)
- such results led to control design techniques (Pappas TAC'06, Belta TAC'07)

Massachusetts Institute of Technology

Finite Time Horizon (FTH) Model Checking of Linear Systems (Kwan and Agha HSCC'08)

- model checking linear systems is decidable under finite time horizon
- stress that their results can be used in LTL model predictive control
- we also employ the FTH in mission planning examples

Massachusetts Institute of Technology

Finite Time Horizon (FTH) Model Checking of Linear Systems (Kwan and Agha HSCC'08)

- model checking linear systems is decidable under finite time horizon
- stress that their results can be used in LTL model predictive control
- we also employ the FTH in mission planning examples

FTH model checking of hybrid affine systems is decidable (Karaman et.al. CDC'08)

- model checking hybrid affine systems is decidable under finite time horizon assumption
- we show that our results are directly applicable to optimal control
- similar MILP formulation is employed which was used for mission planning
- work in progress for model predictive control

Massachusetts Institute of Technology

Finite Time Horizon (FTH) Model Checking of Linear Systems (Kwan and Agha HSCC'08)

- model checking linear systems is decidable under finite time horizon
- stress that their results can be used in LTL model predictive control
- we also employ the FTH in mission planning examples

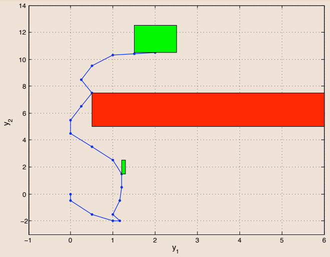
FTH model checking of hybrid affine systems is decidable (Karaman et.al. CDC'08)

- model checking hybrid affine systems is decidable under finite time horizon assumption
- we show that our results are directly applicable to optimal control
- similar MILP formulation is employed which was used for mission planning
- work in progress for model predictive control

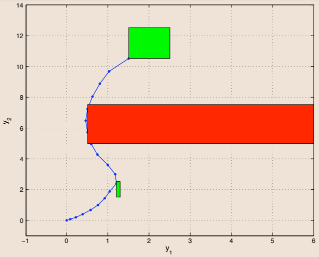
Small illustrative example: second order hybrid linear system in $x - y$ plane

Eventually visit region 1 and region 2 and always avoid region 3, i.e., $\Diamond(R_1 \wedge R_2) \wedge \Box\neg R_3$

Model Checking



Optimal Control



Relaxing the finite time horizon assumption

Recent Results

- Very recently we have extended our algorithm to relax FTH assumption!
- We provide an algorithm which terminates in finite time if the formula is satisfied and runs for ever otherwise
- Work in progress to find a way to detect termination conditions for the algorithm
- Our results still extend to the optimal control case
- We propose the first Integer Programming based LTL Satisfiability Solver
    - In practice, SAT Solvers are used as a subroutine for model checking of concurrent programs (There are commercial model checkers using SAT solvers, e.g. NuSMV)
    - There are Integer programming based SAT solvers for classical propositional logic
    - We have not experimented the performance of our SAT solver in a comparison study yet

Massachusetts Institute of Technology

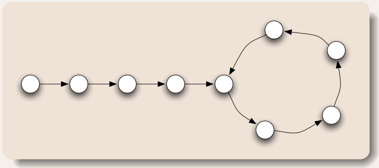
Relaxing the finite time horizon assumption

Recent Results

- Very recently we have extended our algorithm to relax FTH assumption!
- We provide an algorithm which terminates in finite time if the formula is satisfied and runs for ever otherwise
- Work in progress to find a way to detect termination conditions for the algorithm
- Our results still extend to the optimal control case
- We propose the first Integer Programming based LTL Satisfiability Solver
    - In practice, SAT Solvers are used as a subroutine for model checking of concurrent programs (There are commercial model checkers using SAT solvers, e.g. NuSMV)
    - There are Integer programming based SAT solvers for classical propositional logic
    - We have not experimented the performance of our SAT solver in a comparison study yet

Two time horizons and extending the formulation

Consider a **prefix** and a **suffix** time horizon which will model a loop and a path to the loop



Theorem: If there exists an infinite execution that satisfies an LTL formula then there must be an infinite execution with a periodic loop that satisfies the formula

An illustrative example

Double integrator in $x-y$ plane

The model

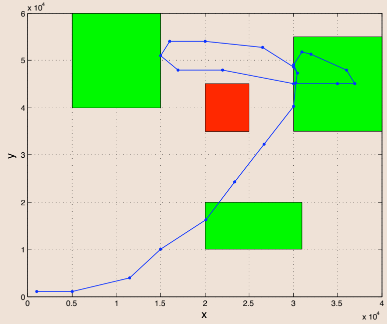
$$
\begin{aligned}
x_1(t+1) &= x_1(t) + \Delta t x_2(t)\\
x_2(t+1) &= x_2(t) + \Delta t u_x(t)\\
y_1(t+1) &= y_1(t) + \Delta t y_2(t)\\
y_2(t+1) &= y_2(t) + \Delta t u_y(t)
\end{aligned}
$$

LTL Specification

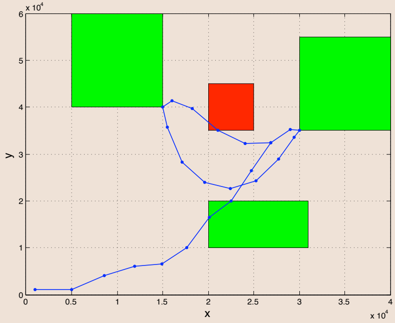
Eventually go to region 1 and always eventually visit region 2 and always eventually visit region 3 always avoid region 4.

$$\Diamond R_1 \wedge \Box\Diamond R_2 \wedge \Box\Diamond R_3 \wedge \Box\neg R_4$$

Model Checking



Optimal Control



Modeling Persistency: Routing, Surveillance, and Reconnaissance Missions

Does the VRP make sense without the Finite Time Horizon Assumption?

- In mission planning we have assumed finite time horizon, i.e., UAVs are launched and after servicing a number of targets they all land and the mission is completed
- Let us consider a mission that is persistent in the sense that
  - Mission plan for a vehicle can be service some of the targets then go to a base to get fuel and then service a possible different set of targets and go back to the same base to get fuel again. Repeat the final loop forever.
  - A feasible mission plan satisfies the given temporal formula and finds a *loop* and a *path* to that loop for each vehicle that is launched
  - Then one can optimize to obtain the best solution among the feasible

Massachusetts Institute of Technology

Modeling Persistency: Routing, Surveillance, and Reconnaissance Missions

Does the VRP make sense without the Finite Time Horizon Assumption?

- In mission planning we have assumed finite time horizon, i.e., UAVs are launched and after servicing a number of targets they all land and the mission is completed
- Let us consider a mission that is persistent in the sense that
  - Mission plan for a vehicle can be service some of the targets then go to a base to get fuel and then service a possible different set of targets and go back to the same base to get fuel again. Repeat the final loop forever.
  - A feasible mission plan satisfies the given temporal formula and finds a *loop* and a *path* to that loop for each vehicle that is launched
  - Then one can optimize to obtain the best solution among the feasible

LTL with persistent missions

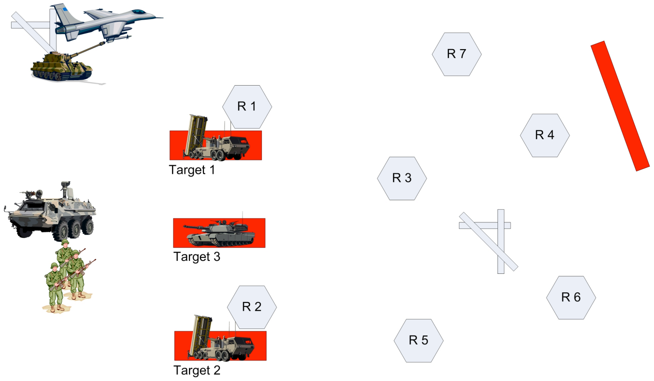
- Always eventually visit a region : $\Box\Diamond P$
- It is always the case that an event requires occurrence of another eventually : $\Box(P \rightarrow \Diamond Q)$

Quantitative properties

Metric Temporal Logics also has this infinite time understanding. Similar arguments also hold for the MTL case. Even more interesting conditions can be modeled.

$$\Box\Diamond_{[0,5]} P \wedge \Box\left(P \rightarrow \left(\Diamond_{[0,2]} Q \wedge \Box_{(0,2]} \neg P\right)\right)$$

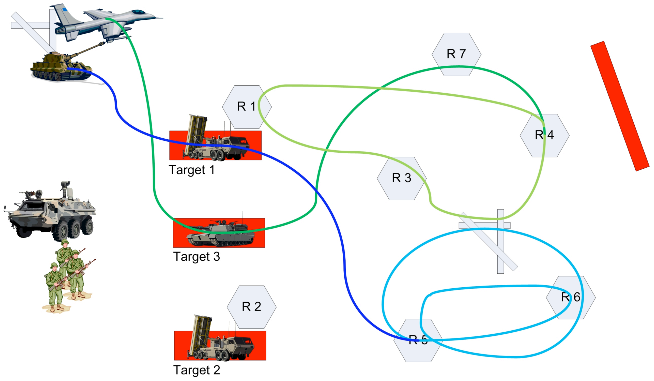
Modeling Persistency: Routing, Surveillance, and Reconnaissance Missions



Mission Specification

- Eventually destroy Target 3 and either Target 1 or 2 : $(\Diamond T1 \vee \Diamond T2) \wedge \Diamond T3$
- Always patrol R3, R4, R5, and R6 : $\Box\Diamond R3 \wedge \Box\Diamond R4 \wedge \Box\Diamond R5 \wedge \Box\Diamond R6$
- If destroyed Target 1 then always patrol R1 : $(\Diamond T1) \rightarrow (\Box\Diamond R1)$
- If destroyed Target 2 then always patrol R2 : $(\Diamond T2) \rightarrow (\Box\Diamond R2)$
- If destroyed T1 then eventually search R7 : $(\Diamond T1) \rightarrow (\Diamond R7)$

Modeling Persistency: Routing, Surveillance, and Reconnaissance Missions



Mission Specification

- Eventually destroy Target 3 and either Target 1 or 2 : $(\Diamond T1 \vee \Diamond T2) \wedge \Diamond T3$
- Always patrol R3, R4, R5, and R6 : $\Box\Diamond R3 \wedge \Box\Diamond R4 \wedge \Box\Diamond R5 \wedge \Box\Diamond R6$
- If destroyed Target 1 then always patrol R1 : $(\Diamond T1) \rightarrow (\Box\Diamond R1)$
- If destroyed Target 2 then always patrol R2 : $(\Diamond T2) \rightarrow (\Box\Diamond R2)$
- If destroyed T1 then eventually search R7 : $(\Diamond T1) \rightarrow (\Diamond R7)$

Human supervised strategic mission planning with optimal tactics

Humans vs. Computers

Humans

- Humans are good at seeing the big picture
- They can provide high level insights
- They have much better reasoning abilities
- They have a better sense of **strategy**

Computers

- Computers are good at running messy computations quickly given algorithms
- They can determine the **tactics** that involve several complex and conflicting details

Massachusetts Institute of Technology

Human supervised strategic mission planning with optimal tactics

Humans vs. Computers

Humans

- Humans are good at seeing the big picture
- They can provide high level insights
- They have much better reasoning abilities
- They have a better sense of **strategy**

Computers

- Computers are good at running messy computations quickly given algorithms
- They can determine the **tactics** that involve several complex and conflicting details

Humans provide reasonable strategy

Humans provide several different decoupled missions using LTL. A human operator associates vehicles with missions providing **strategy**

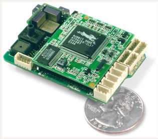
Computers provide optimal tactics

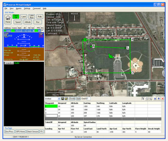
Computer first checks feasibility of mission and vehicle assignments. If so computes the optimal **tactics**

Massachusetts Institute of Technology

off-the-shelf autopilots / off-the-shelf algorithms

Procerus UAV, Kestrel Autopilot and Virtual Cockpit (Not funded through MACCCS)





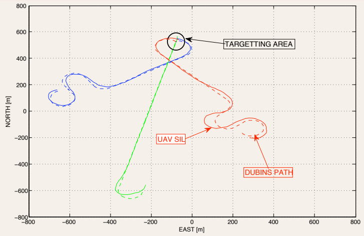
- Virtual Cockpit provides a TCP based communication interface
- Access to data and uploading commands to the UAV in real-time

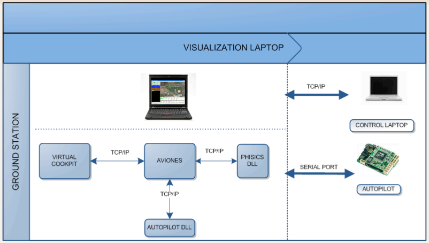
Tracking Dubins Paths in Hardware-in-the-loop simulation

- Several planning algorithms assume bounded curvature model (Dubins model) of the UAVs and generate motion plans accordingly
- Can we control Procerus UAVs to effectively follow a given Dubins path?

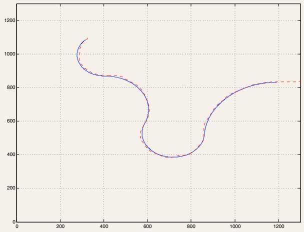
A Dubins Path Tracking Implementation in HiL Environment

Implementation of a rendezvous algorithm using waypoint commands



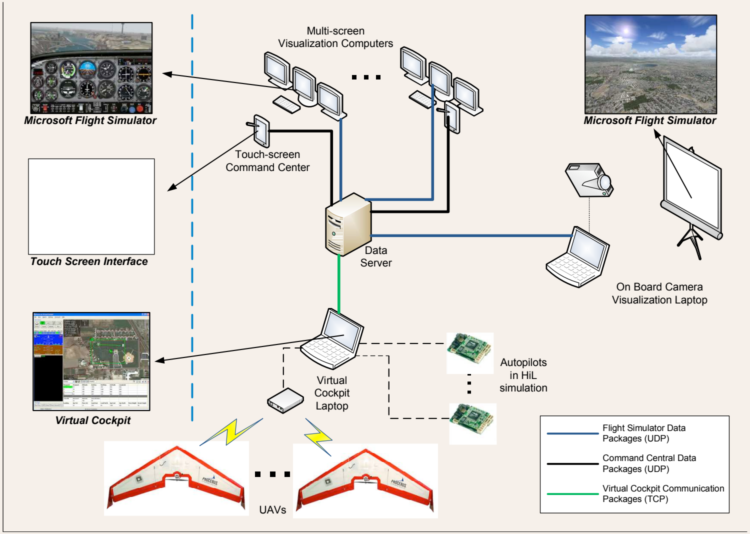


Implementation using roll and speed commands



- Better tracking performance by directly using the roll and speed commands
- Work in progress to implement several different planning algorithms that rely on Dubins models

Hardware Diagram


Microsoft Flight Simulator
Touch Screen Interface
Virtual Cockpit
Multi-screen Visualization Computers
Touch-screen Command Center
Data Server
Microsoft Flight Simulator
On Board Camera Visualization Laptop
Virtual Cockpit Laptop
Autopilots in HiL simulation
UAVs
Flight Simulator Data Packages (UDP)
Command Central Data Packages (UDP)
Virtual Cockpit Communication Packages (TCP)

Massachusetts Institute of Technology

Conclusions

- Linear Temporal Logic is remarkably close to natural language and turns out to model several interesting cases in mission planning problems.
- Metric Temporal Logics are even more expressive than LTL. MTL seems to model several crucial quantitative properties naturally.
- We have shown application of our methods to model checking of dynamical systems and controls.
- Infinite time horizon in Temporal Logics allows modeling several interesting scenarios. We are employing theoretical tools from automata theory and infinite games to solve these problems.
- We are putting together an experimental setup that will allow us to do many human-in-the-loop experiments by realistic simulation.

Massachusetts Institute of Technology

Thanks

Formal Approaches to Mission Planning

Questions?

Massachusetts Institute of Technology