Temporal Logic Motion Planning for Mobile Robots

Georgios E. Fainekos, Hadas Kress-Gazit and George J. Pappas

Penn University of Pennsylvania

GRASP

Problem Formulation

Model: We consider a fully actuated, planar model of robot motion operating in a polygonal environment P. The motion of the robot is expressed as:

$$\dot{x}(t) = u(t) \quad x(t) \in P \subseteq R^2 \quad u(t) \in U \subseteq R^2$$

Specification: A linear temporal logic (LTL) formula φ that captures the robots' desired behavior.

Problem: Given robot model, environment P, initial condition x(0), and an LTL_{-X} temporal logic formula φ , find control input u(t) such that x(t) satisfies φ.

LTL_{-X} specifications

The formulas are built from a finite number of atomic propositions Π which label areas of interest in the environment such as *rooms* or *obstacles*. Proposition π∈Π represents an area of interest in the environment which can be characterized by a convex set of the form:

$$P_i = \{x \in \Re^2 \mid \bigwedge a_k^T x + b_k \leq 0,\ a_k \in R^2, b_k \in R\}$$

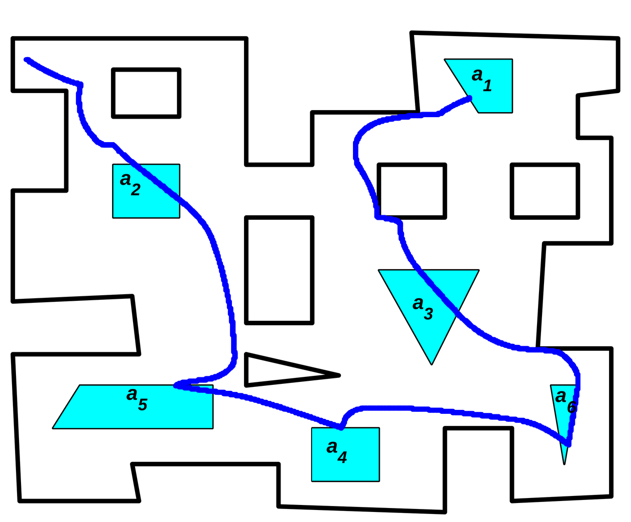
The propositional formulas are formed using the traditional operators of *conjunction* (∧), *disjunction* (∨), *negation* (¬), *implication* (⇒), and *equivalence* (⇔). LTL_{-X} formulas are obtained from the standard propositional logic by adding temporal operators such as *eventually* (◊), *always* (□), and *until* (U).

Some LTL examples that express interesting properties in the context of mobile robot motion planning include:

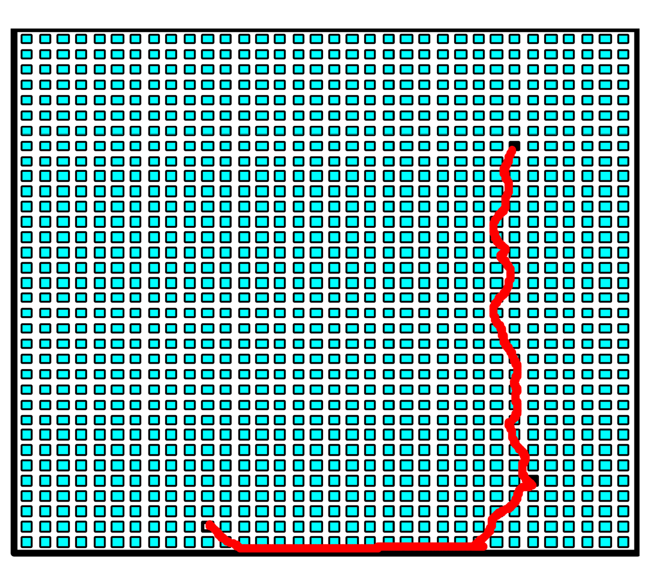
- Reachability: The formula "◊π" means that the trajectory of the robot should eventually reach a state where we observe π.
- Reachability while avoiding obstacles: The formula "¬ ($o_1 \vee o_2 \vee ... \vee o_n$) U π" expresses the property that eventually π will be true, and until π is reached, we must avoid all obstacles labeled as o_i, i=1,…,n.
- Sequencing: The requirement that we must visit π_1, π_2 and π_3 in that order is naturally captured by the formula "◊ (π_1 ∧ ◊ (π_2 ∧ ◊ π_3))".
- Coverage: Formula "◊ π_1 ∧ ◊ π_2 ∧ … ∧ ◊ π_m" reads as the robot will eventually reach π_1 and eventually π_2 and … and eventually π_m, requiring the robot to eventually visit all regions of interest.
- Recurrence: The formula "□(◊ π_1 ∧ ◊ π_2 ∧ … ∧ ◊ π_m)" requires that the robot trajectory does whatever the coverage does and, in addition, will force the robot to repeat the desired objective infinitely.

Examples

Spec: Go to areas 1, 2, 3, 4, 5, 6 in no particular order

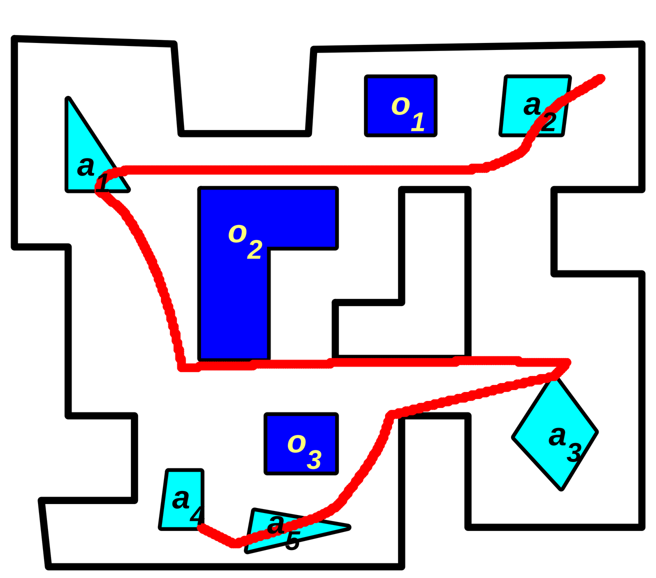


Spec: Go to the two black rooms

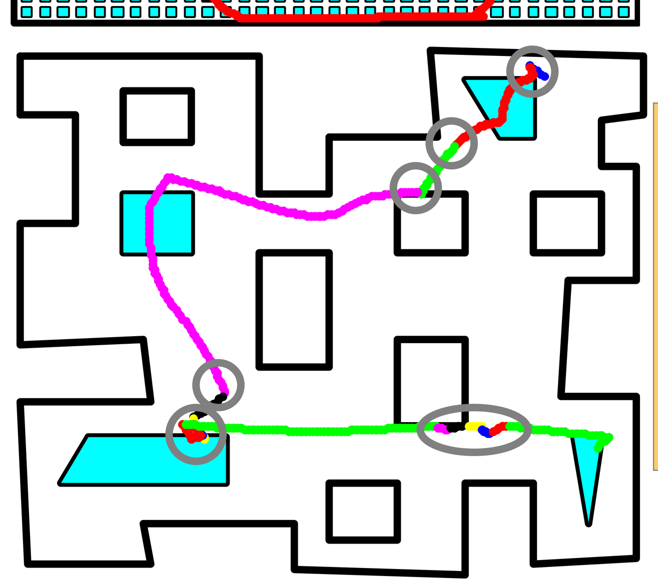


Problem Size
1156 observables
9250 triangles
Solution path:
145 triangles
145 controllers

Computation time
Triangulation:
A few seconds
NuSMV:
55 seconds
Matlab:
90 seconds

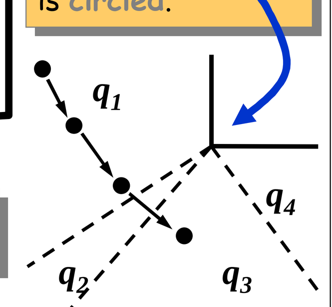


Spec: Go to area 2, then to area 1 and then cover areas 3, 4, 5 – all this, while avoiding obstacles o_1, o_2, o_3

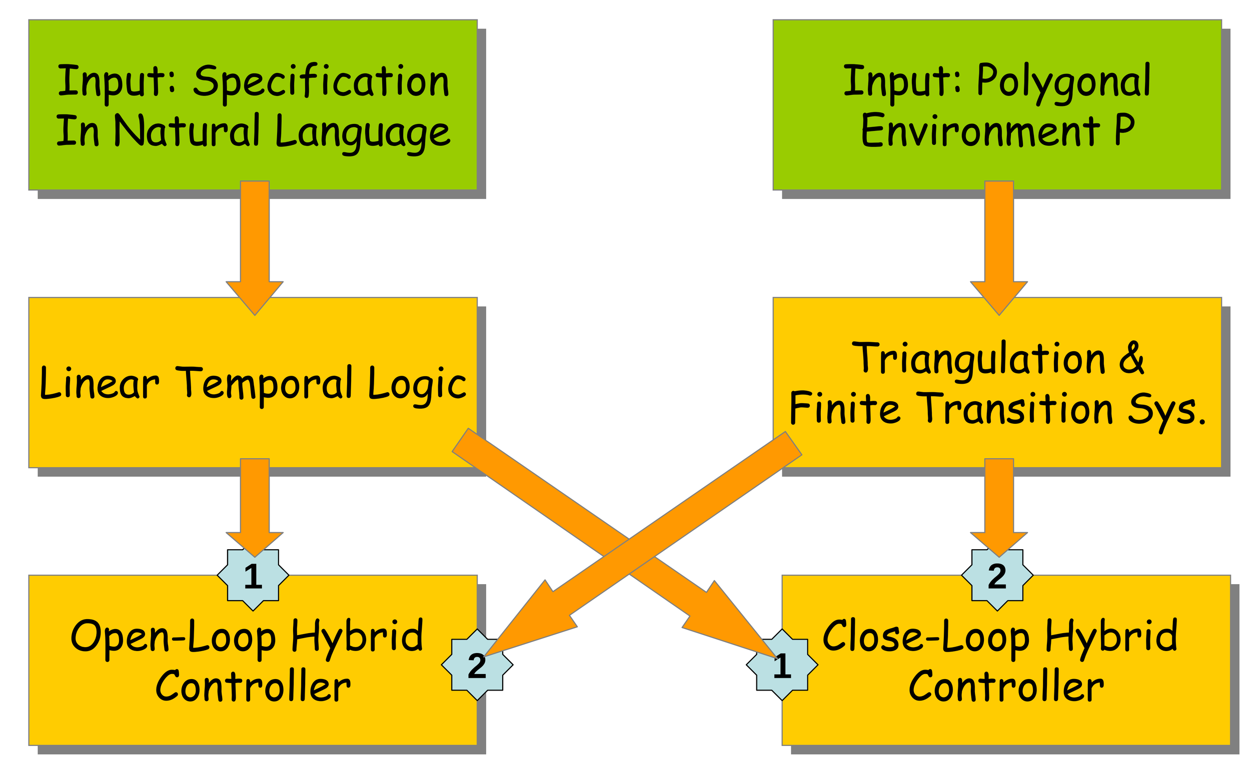


Spec: Visit all highlighted areas

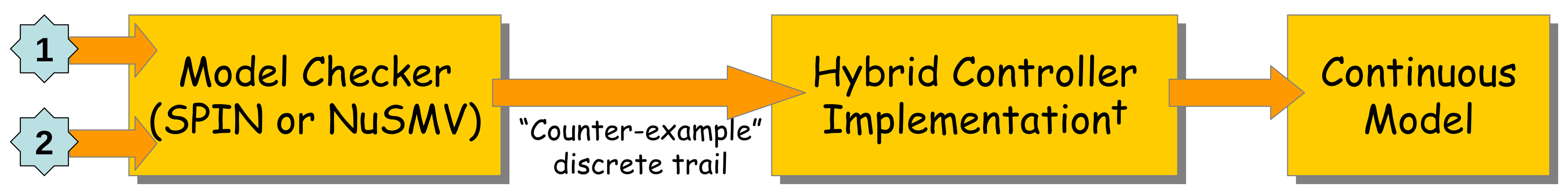
Trajectory created by the **close-loop hybrid** controller. Each divergence from the planned discrete path due to sampling errors is circled.



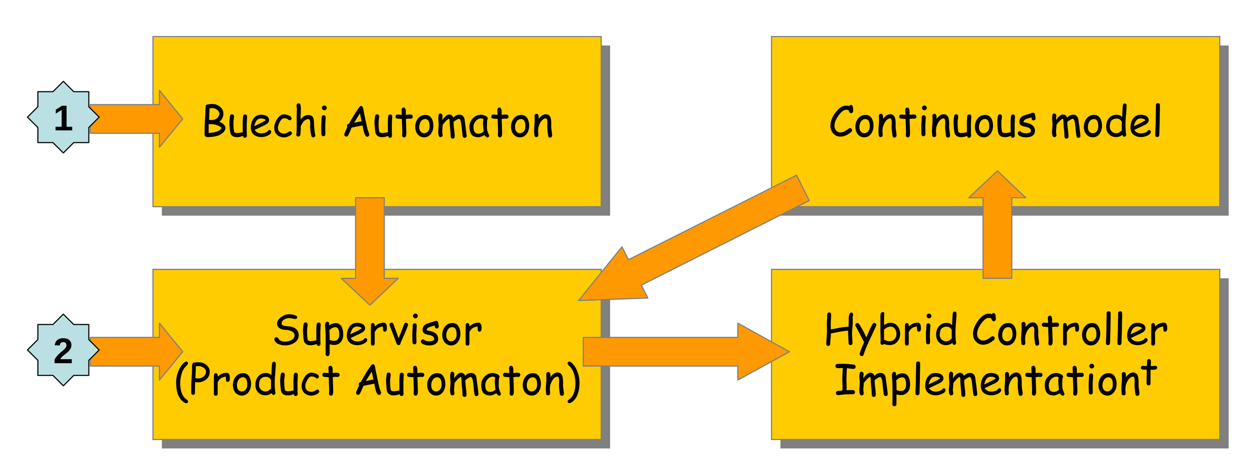
Overview of Temporal Logic Motion Planning



Open-Loop Hybrid Controller♣

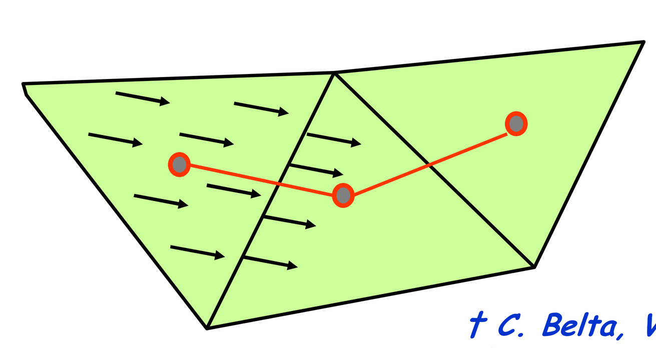


Close-Loop Hybrid Controller♥



Hybrid Controller Implementation†

A triangulation is a **bi-simulation** if the system can move between any two adjacent triangles regardless of the initial state. For each triangle, we design three controllers ensuring that system exits the triangle from the desired facet to the adjacent triangle.



Thm: There exist (many) affine vector fields

$$\dot{x}_P = u_P \qquad u_P = Ax + b \in U_P$$

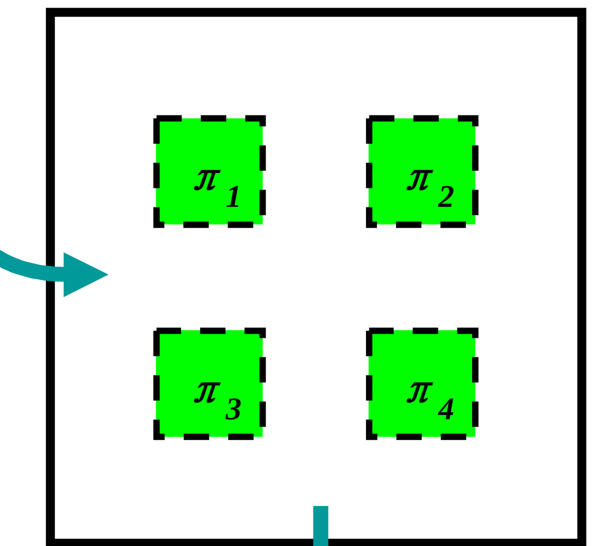
on any triangle, satisfying the bisimulation property.

Affine functions on simplexes are uniquely defined on vertices. The set of all controllers can be parameterized by the values on the vertices.

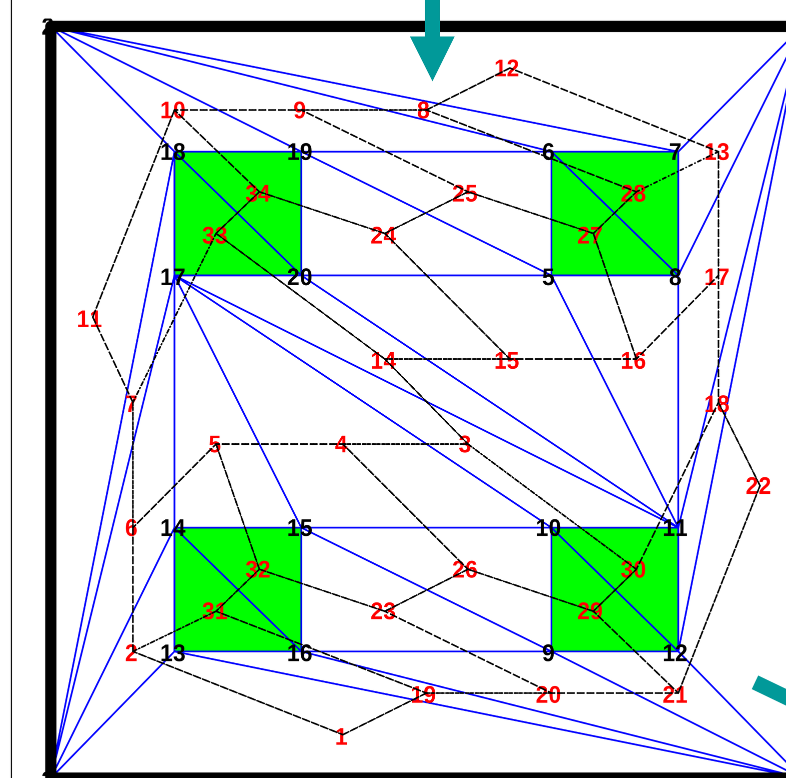
† C. Belta, V. Isler, and G. J. Pappas, "Discrete abstractions for robot motion planning and control," IEEE Transactions on Robotics, Accepted for publication.

A Simple Example

Consider a robot that is moving in a square environment with four areas of interest denoted by π_1, π_2, π_3 and π_4. Initially, the robot is placed somewhere in the region labeled by π_1. The desired specification for the robot given in natural language is: "Visit area π_2 then area π_3 then area π_4 and, finally, return to region π_1 while avoiding areas π_2 and π_3".



$$\varphi = \Diamond (\pi_2 \wedge \Diamond (\pi_3 \wedge \Diamond (\pi_4 \wedge (\neg\pi_2 \wedge \neg\pi_3) U \pi_1)))$$



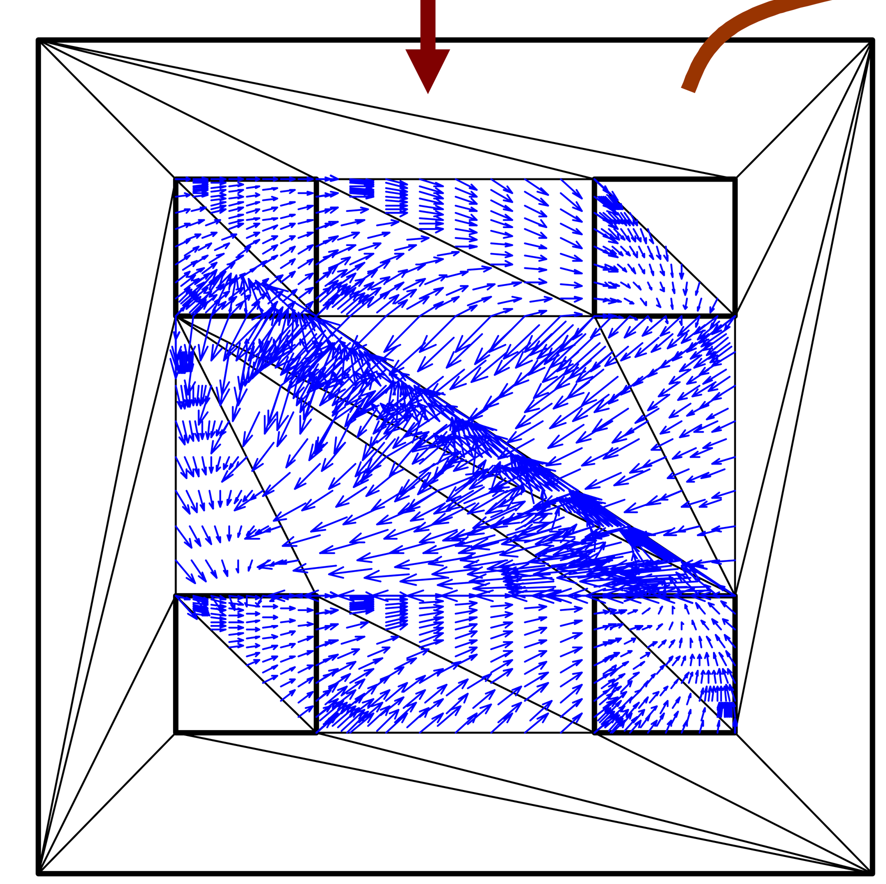
Triangulation and Dual Graph

NuSMV Code

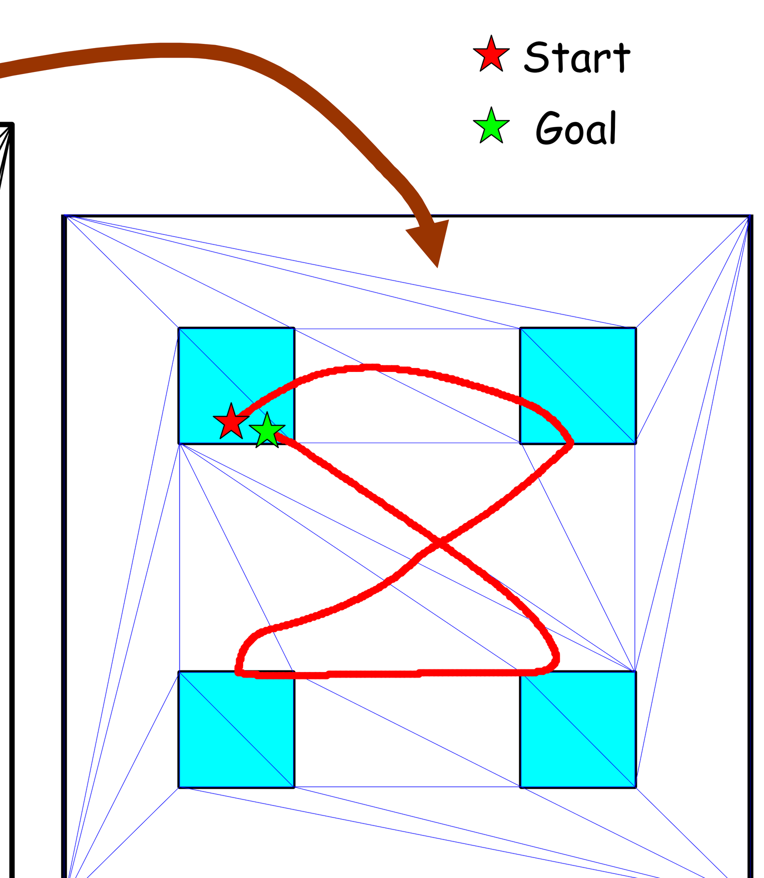
```
MODULE main
    VAR
        robot_1 : process robot(33);
    LTLSPEC ( ! F ( (pi2) & F ( (pi3) &
F ( (pi4) & ( !(pi2) & ! (pi3) U (pi1)))))

MODULE robot(init_cond)
    VAR
        node : 0 .. 34;
    ASSIGN
    init(node) := init_cond;
    next(node) := case
        node=1 :  {2,19};
        node=2 : {1,6,31};
        node=3 : {4,14,30};
        node=4 : {3,5,26};
        node=5 : {4,6,32};
        node=6 : {2,5,7};
        node=7 : {6,11,33};
        …
        node=32 : {5,23,31};
        node=33 : {7,14,34};
        node=34 : {10,24,33};
    esac;
```

The trajectory generated by NuSMV, satisfying this formula is:
33, 34, 24, 25, 27, 16, 15, 14, 3, 4, 5, 32, 23, 26, 29, 30, 3, 14, 33



Open-Loop Hybrid Controller



Resulting continuous mobile robot trajectory

♣ G. E. Fainekos, H. Kress-Gazit and G. J. Pappas, Temporal Logic Motion Planning for Mobile Robots, to appear in the Proceedings of the International Conference on Robotics and Automation, Barcelona, April 2005

♥ G. E. Fainekos, H. Kress-Gazit and G. J. Pappas, Hybrid Controllers for Path Planning: A Temporal Logic Approach, Submitted to the 44th IEEE Conference on Decision and Control, Seville, Spain, 2005