MARYLAND ROBOTICS CENTER
THE INSTITUTE FOR SYSTEMS RESEARCH

Automated Behavior Generation for Unmanned Ground Vehicles Using Virtual Environments

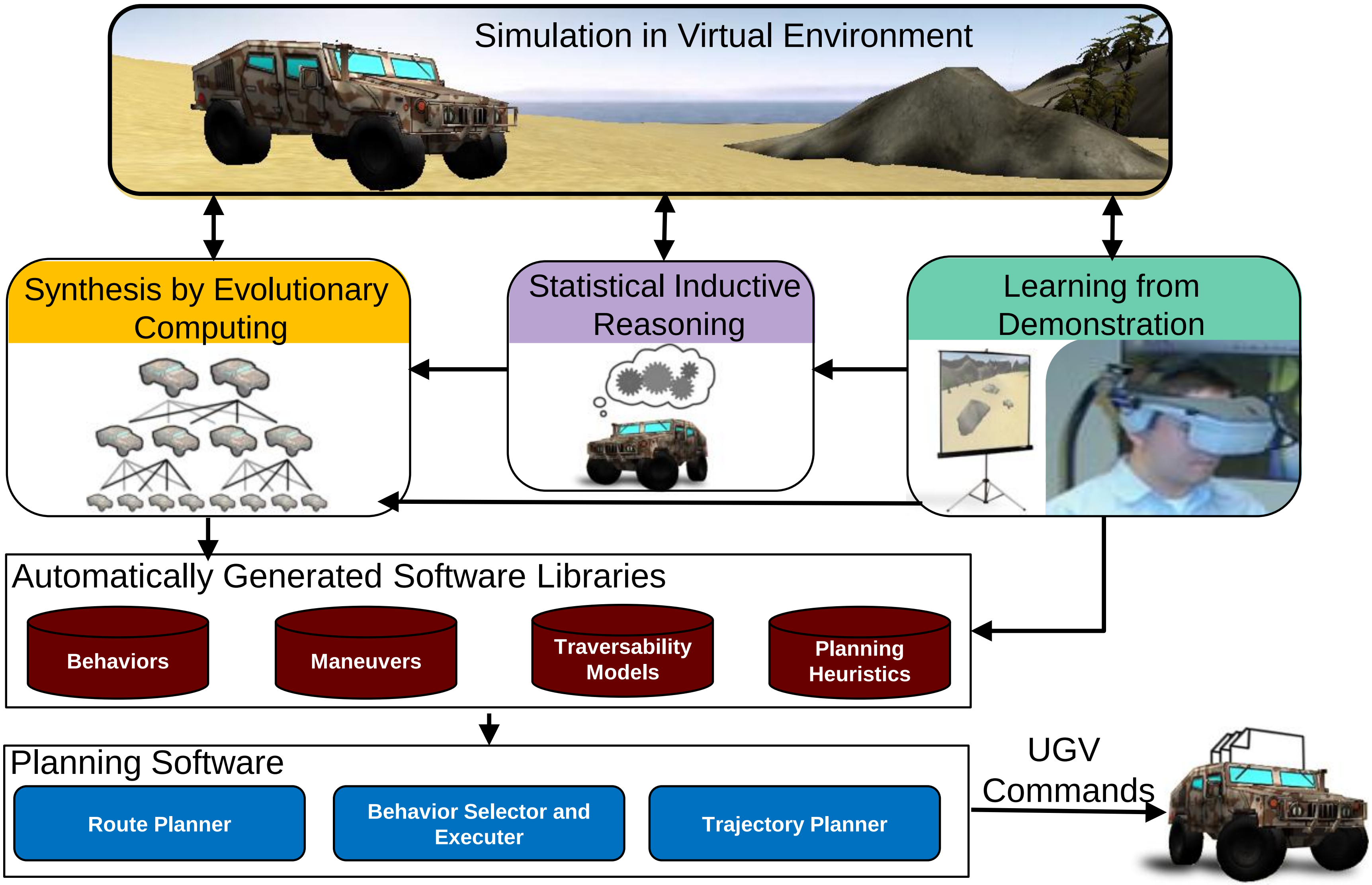
Satyandra K. Gupta, Bob Kavetsky, Steve Lubard,
Madan Dabbeeru, Patrick Fry, Josh Langsfeld, Atul Thakur, and Petr Svec

Sponsor: Office of Naval Research

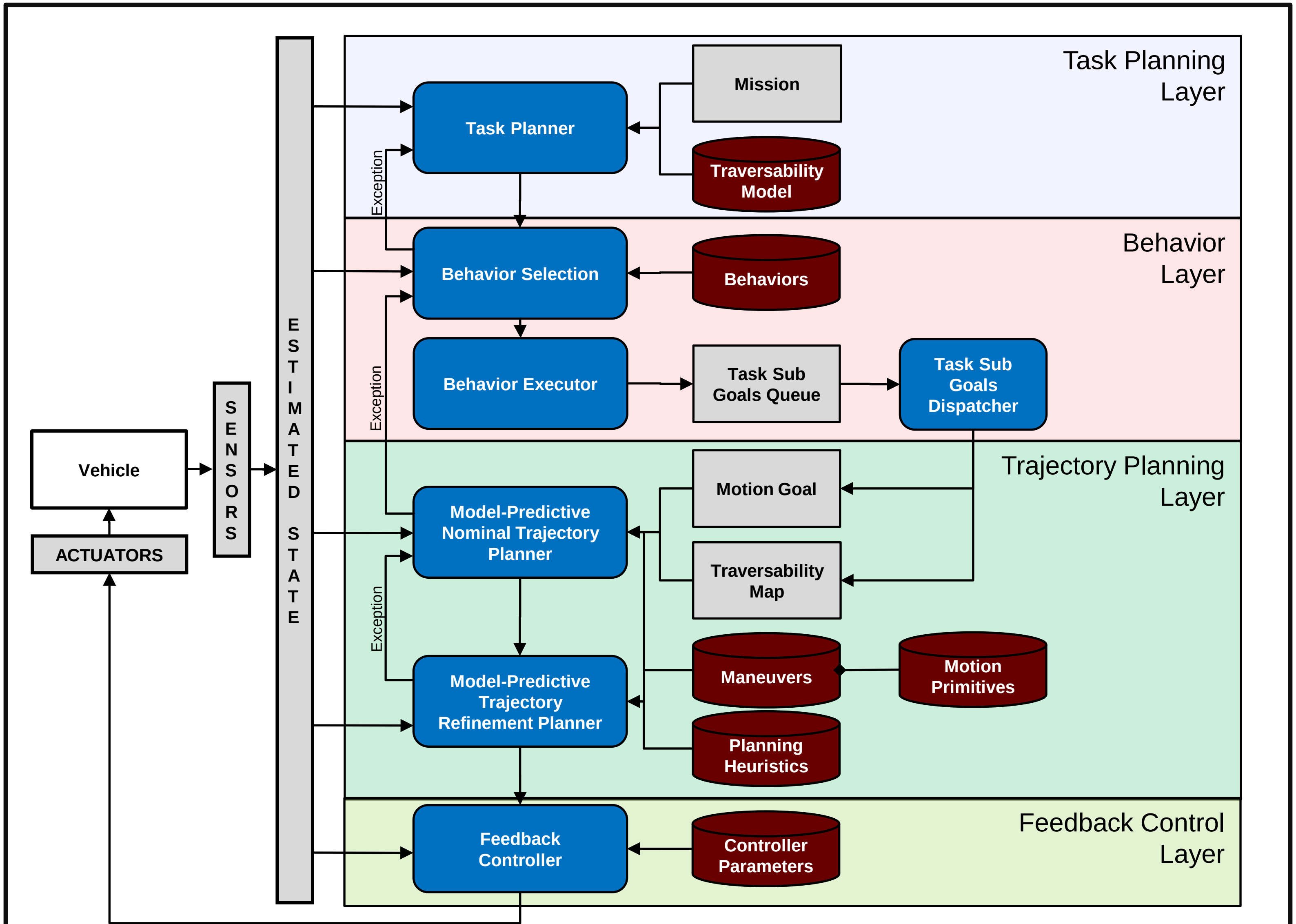
Objectives

- Develop a computational framework for automated generation of behaviors and maneuvers using simulations for autonomous UGVs
  - Behaviors are automatically generated and verified through simulations
  - Symbolic representation facilitates human verification

Overview of Approach



UGV System Architecture



Anticipated Benefits of Our Approach

- Able to generate complex reactive behaviors
  - Handles fractured state space
  - Handles time-delayed action consequences
  - Offers flexibility in terms of function choices
- Works in conjunction with state of the art trajectory planners
  - Can cope up with dynamically moving objects in environment
  - Incorporates vehicle's dynamics constraints using simulations
- Cope with human-made rules
  - Introduces context dependent traversability constraints

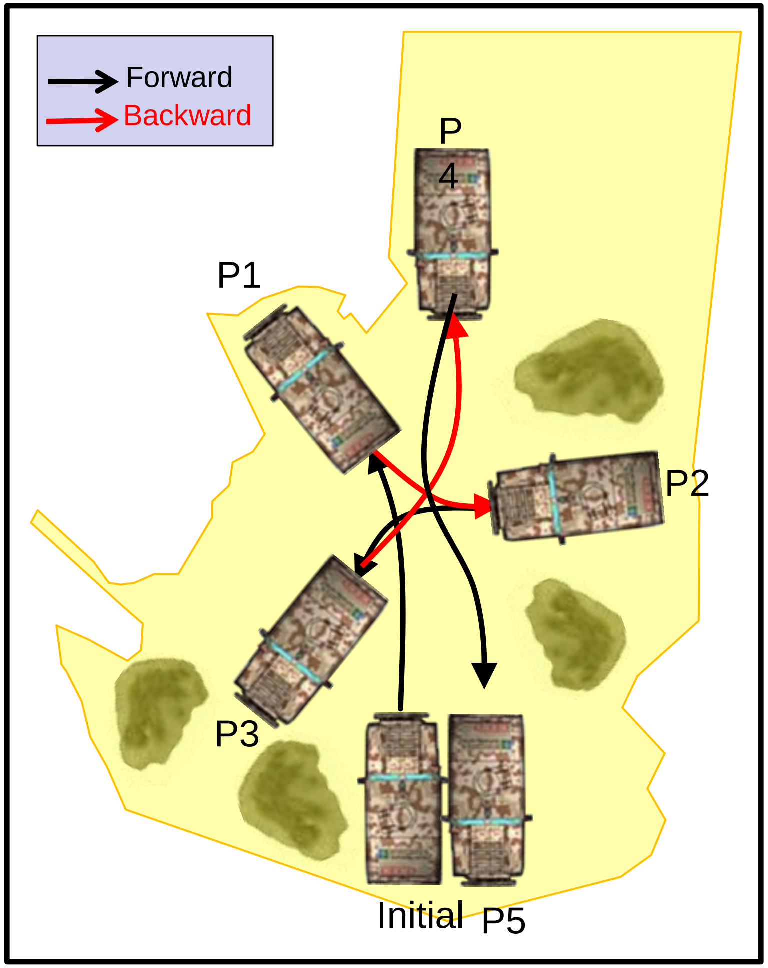
UGV Simulator

- Modular architecture allows physics engine, vehicle controller, and machine learning algorithms to be kept separate
  - Both Unreal engine and high-fidelity Vortex vehicle dynamics engine implemented
- Terrain modeled manually or imported from Digital Elevation Model files representing real-world locations
- Vehicles
  - Are modeled in CAD/3D programs and imported into simulator
  - Have dynamics models that can be tuned for realistic behavior
  - Can be controlled by a human, scripts, or machine-learning algorithms
  - Can have their movement recorded and played back
- Multiple types of sensors can be emulated
  - Vision, velocity, acceleration, wheel spin, rangefinders, etc. are all modeled
- Additional static and dynamic objects, each with their own physical properties, can be added to the environment and controlled for machine learning experiments

N-Point Turn in Constrained Space Behavior

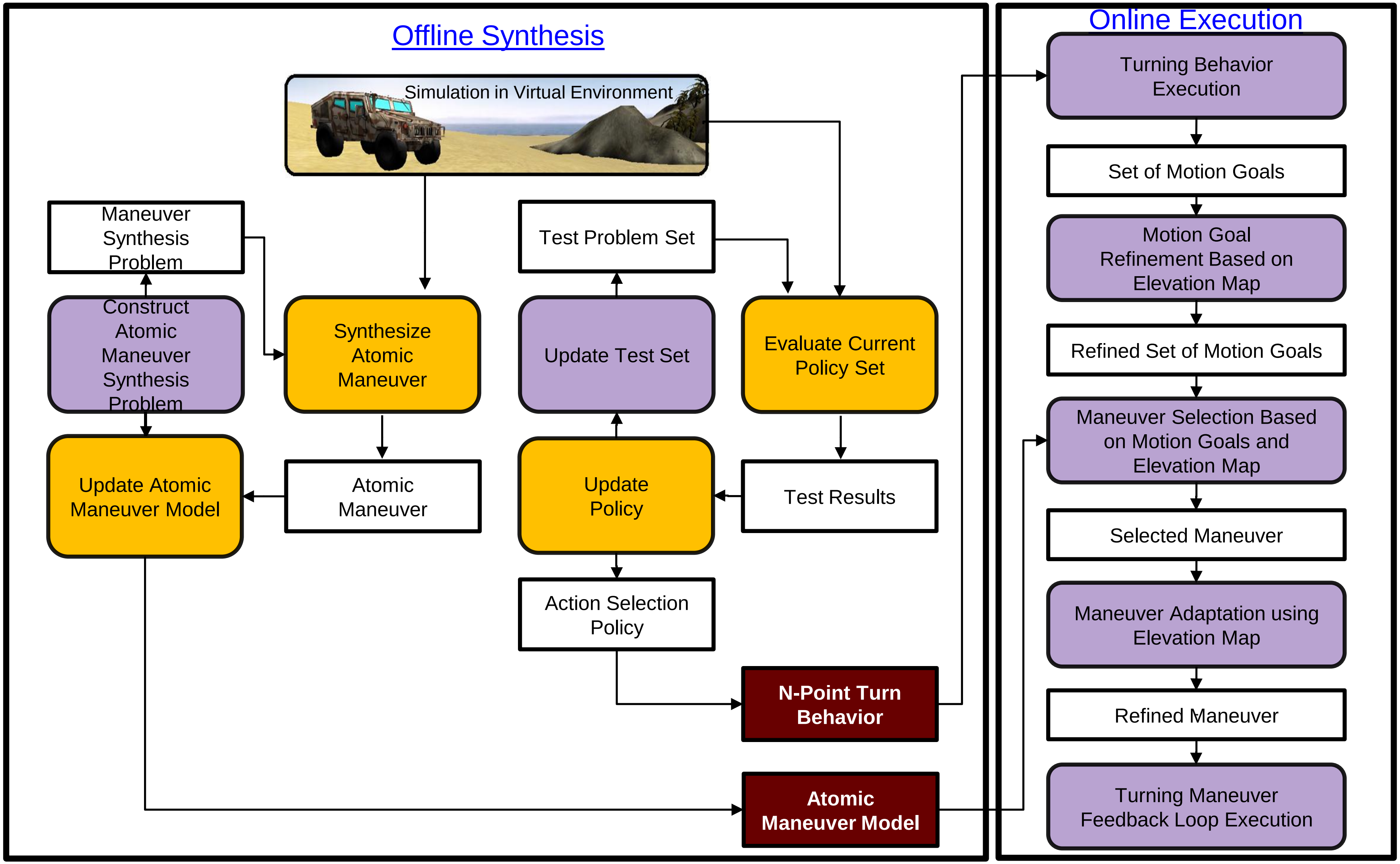
- Generate N-point turn maneuver between two arbitrary states $x_I = [x\ y\ \theta\ v]^T$ and $x_G = [x\ y\ \theta\ v]^T$, that satisfies the dynamics of the vehicle $f_M(x, u, t)$ and minimizes the execution time

- Generated maneuver should reliably be executed on a rugged, highly constrained terrain



5-point turn in a cluttered environment

Approach for Generating N-point Turn Behavior



Results

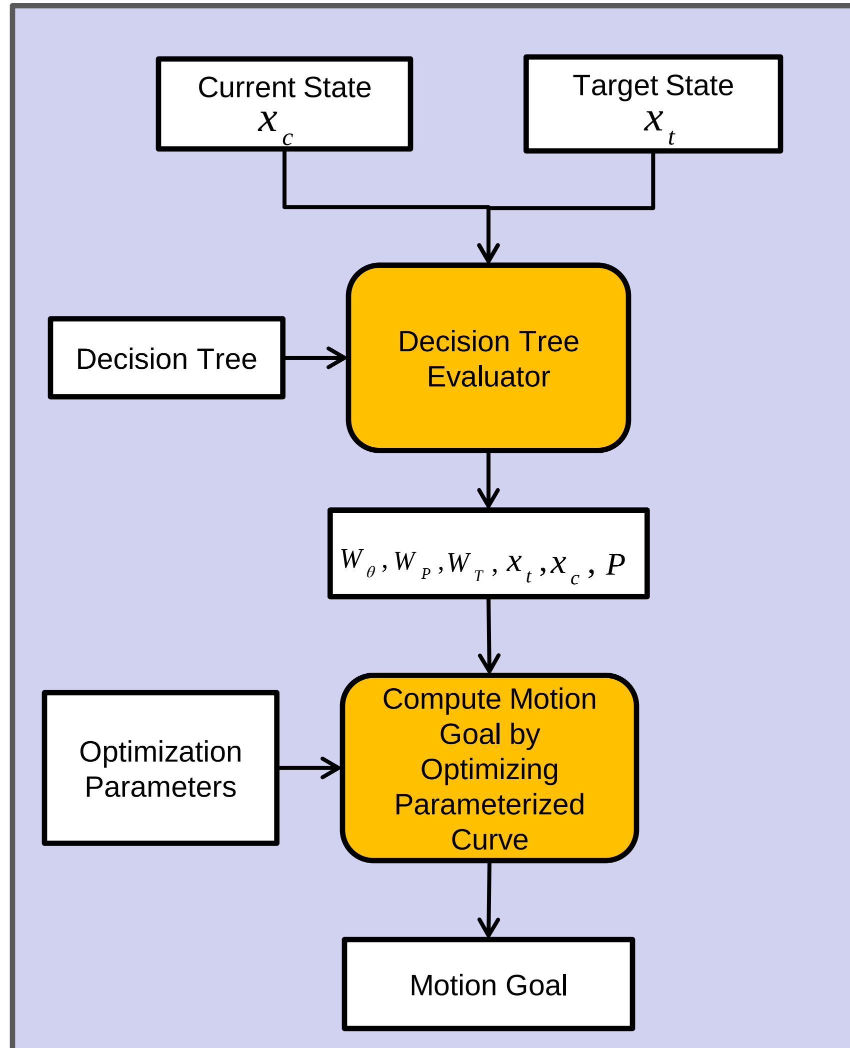
- Policy is represented as a decision tree
- Inputs:
  - Distance to target pose
  - Geometric configuration of obstacles
  - Maximum allowable clearances around the current and target poses
- Outputs:
  - Position error weight W_P
  - Orientation error weight W_θ
  - Terrain error weight W_T
  - Parameterized curve P
- It takes less that 16 seconds to compute motion goals for 13 point turns

$$C = W_\theta C_\theta + W_P C_p + W_T C_T$$

$$C_\theta = \frac{-dot(\vec{v}.\vec{v}')}{2} + \frac{1}{2} \longrightarrow \textit{Orientation}$$

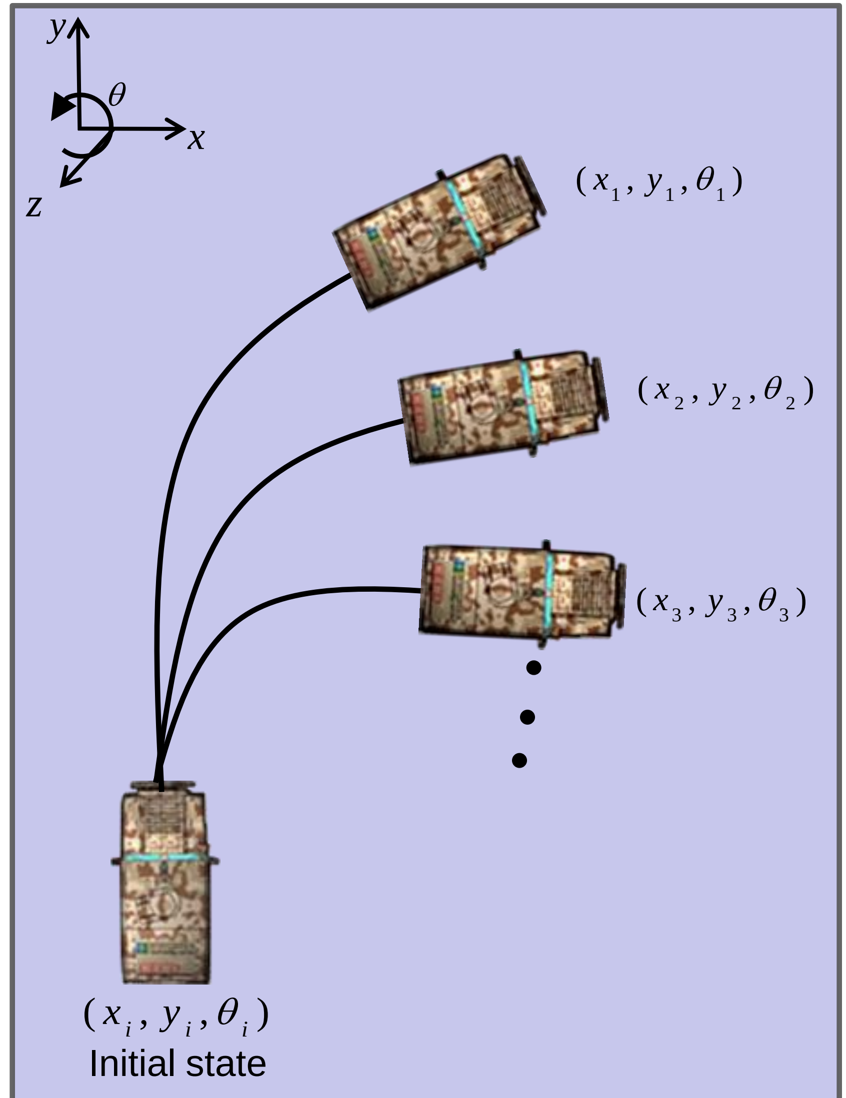
$$C_P = \|x_t - x_c\| \longrightarrow \textit{Position}$$

$$C_T = \begin{cases} 1 & \textit{if collision} \\ 0 & \textit{otherwise} \end{cases} \longrightarrow \textit{Collision}$$



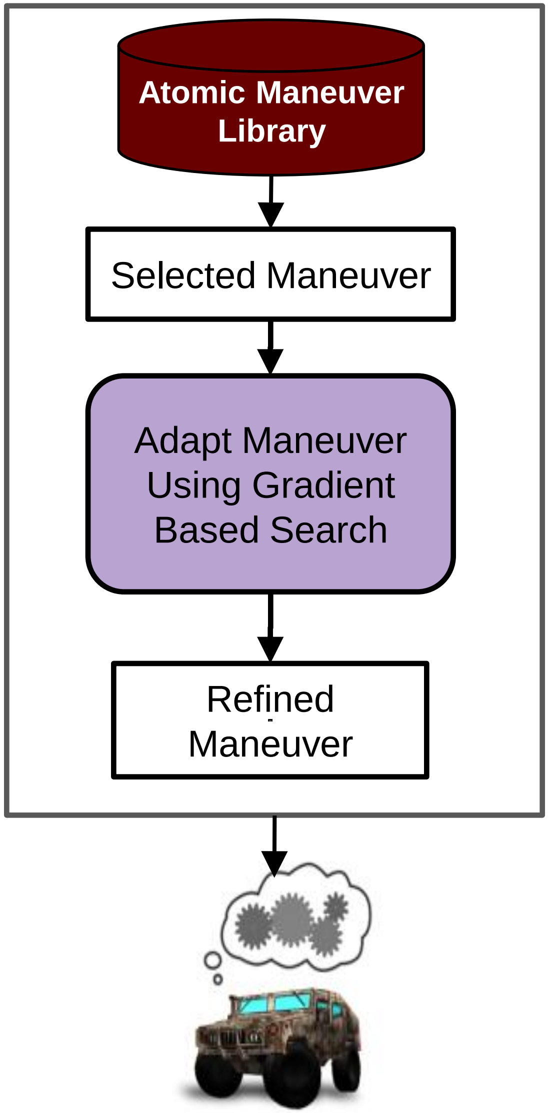
Results (Cont.)

- Automatically synthesized atomic maneuvers using a combination of evolutionary and local search and adaption of nearest neighbor
  - Represented using two 5-point B-splines
    - Force profile (throttle and brake)
    - Steering angle profile
  - Computed over 5 dimensional space
    - Δx Δy and $\Delta \theta$ are the position and orientation differences of the vehicle in the initial and final states
    - α and β are slopes of the terrain
    - Initial and final velocities are considered zero for each atomic maneuver
- Maneuvers are robust with respect to small perturbations in the terrains
  - Tested by introducing waviness in terrain

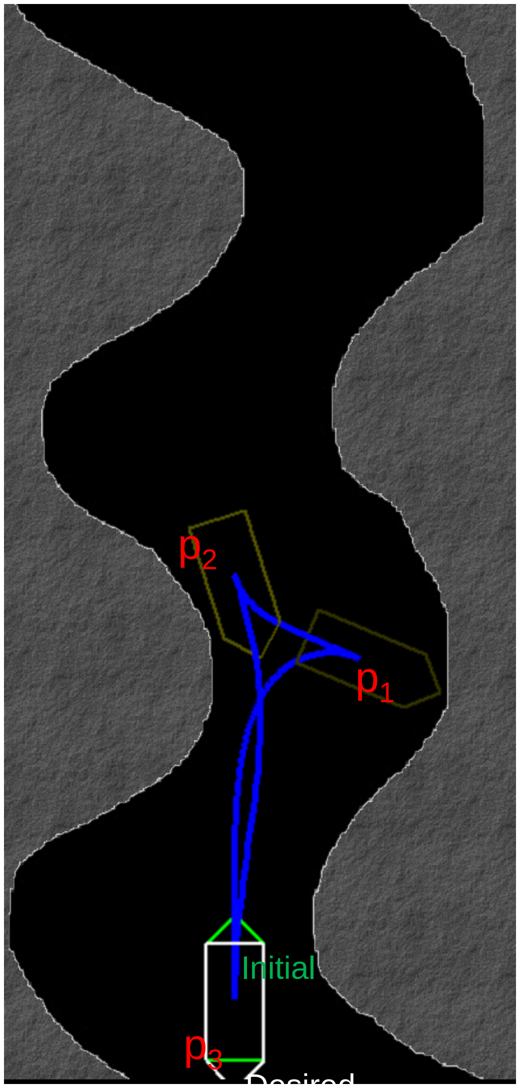


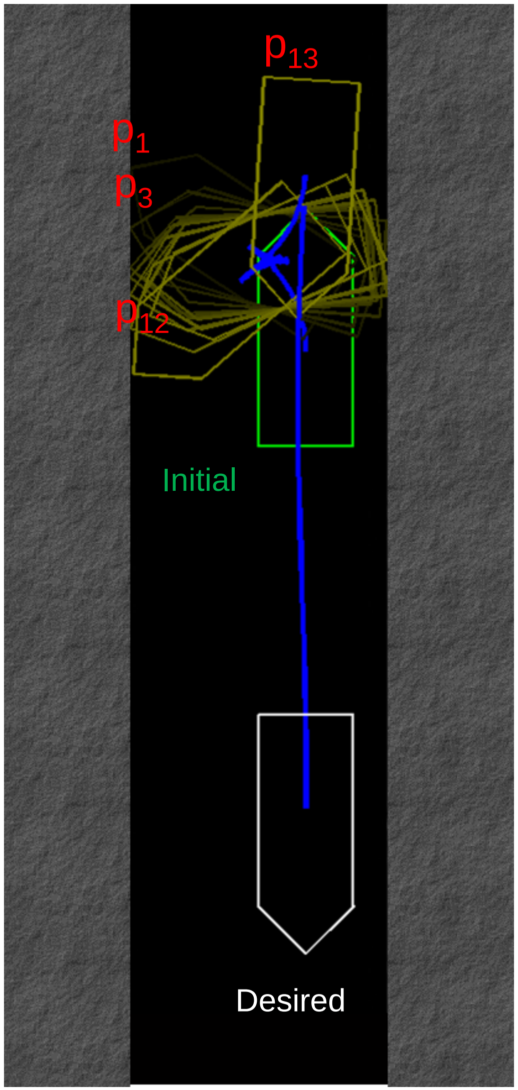
Results (Cont.)

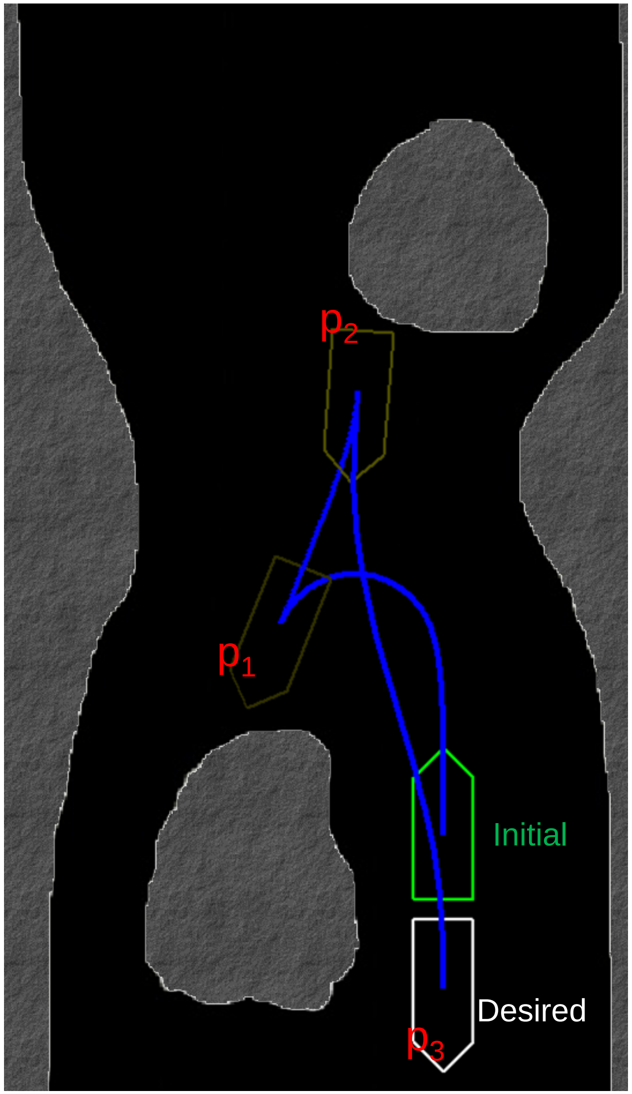
- Identified rules for selecting a reference maneuver from the atomic maneuver library
  - Evaluated several different interpolation strategies
  - Nearest neighbor produced the best results
  - Decided on appropriate grid resolution to allow fast adaptation
- Developed adaptation algorithm for adapting the reference maneuver
  - Reference maneuver adapted using gradient based search methods
  - To improve the faster convergence we used adaptive step-size searching techniques where the step-size is decided based on the information acquired from previous iterations
  - Average time for adaptation: 2 sec



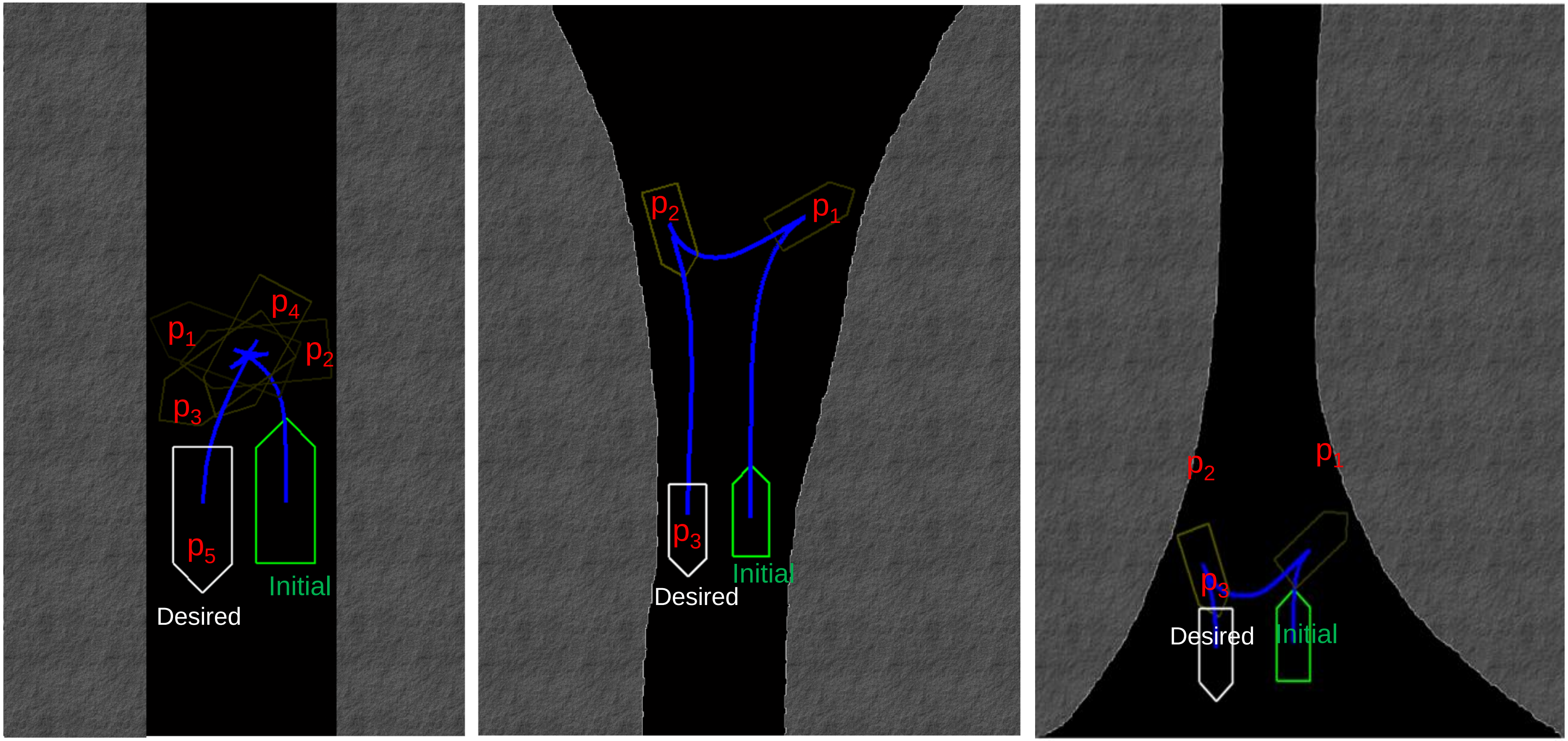
Examples


p_2
p_1
Initial
p_3


p_{13}
p_1
p_3
p_{12}
Initial
Desired

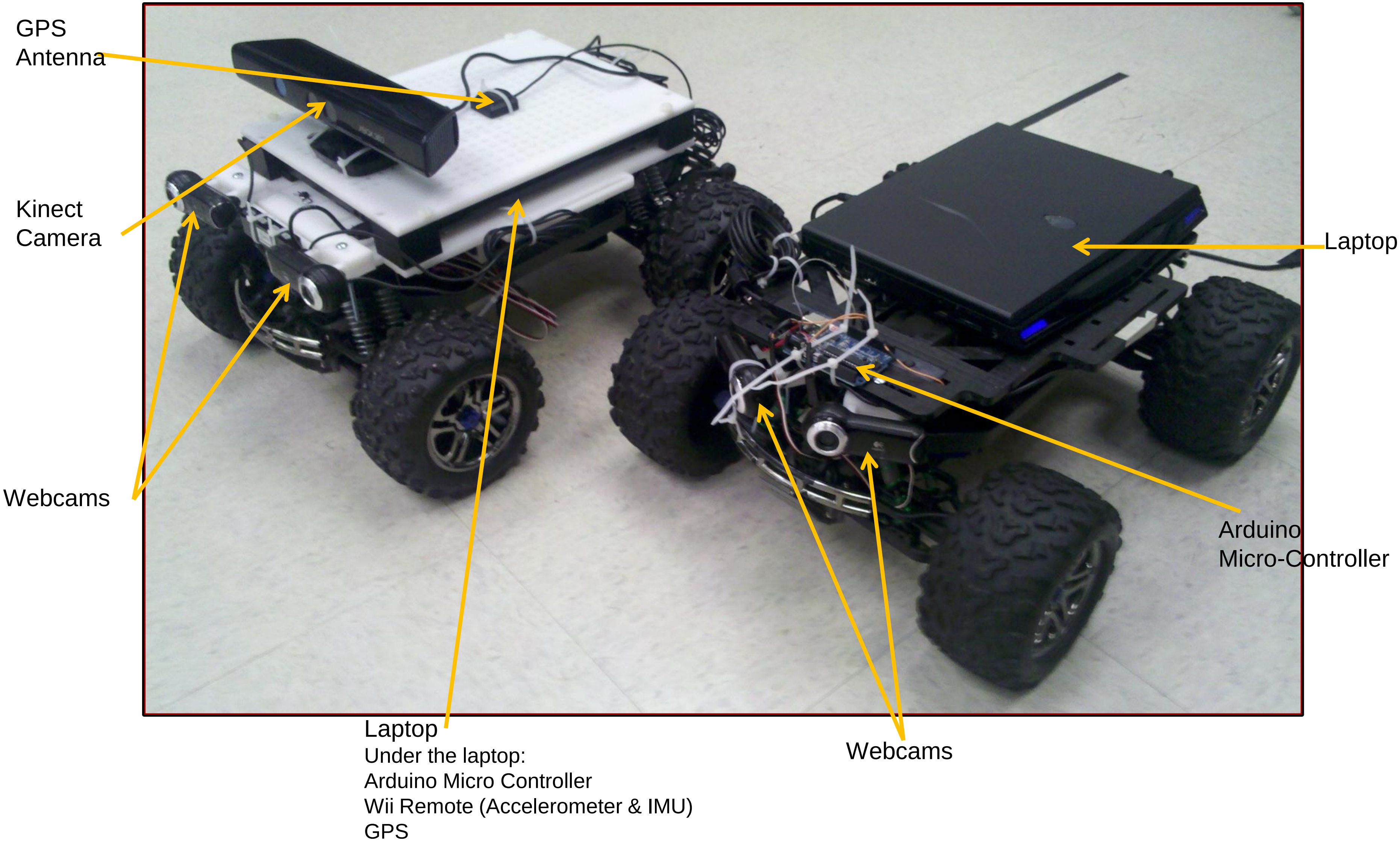

p_2
p_1
Initial
Desired
p_3

Examples (Cont.)

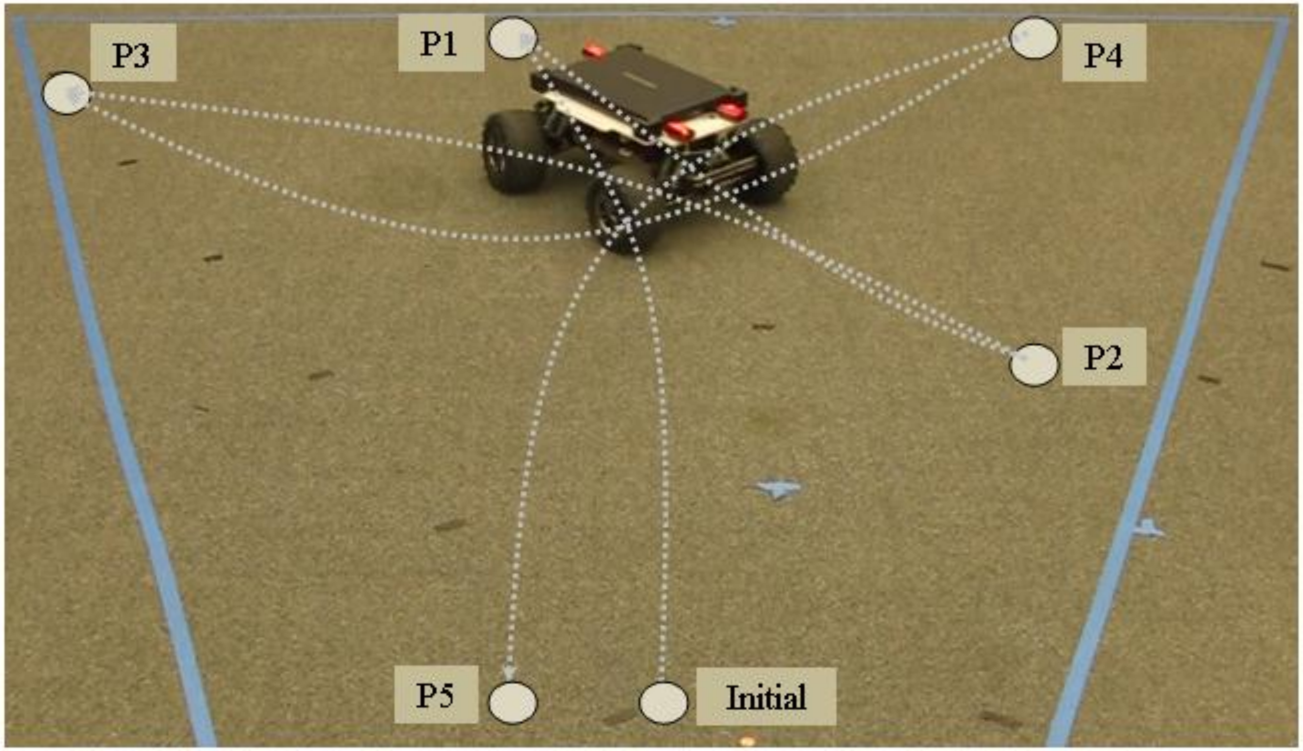


MARYLAND ROBOTICS CENTER
THE INSTITUTE FOR SYSTEMS RESEARCH

Physical Test Platform



Executing 5-point Turn Behavior at Maryland Robotics Realization Laboratory

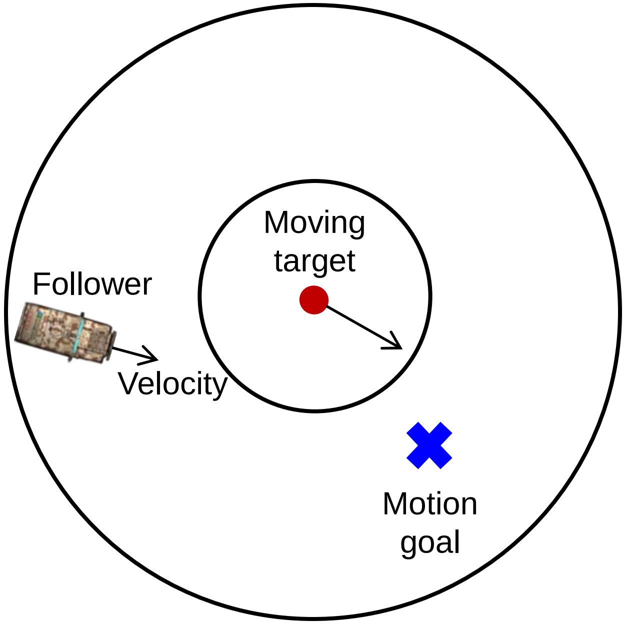


Executing 5-point turn behavior without crossing the blue border lines

MARYLAND ROBOTICS CENTER
THE INSTITUTE FOR SYSTEMS RESEARCH

Follow Target Behavior

- Generate a behavior for robust and efficient following of a moving target on a rugged terrain
  - Follower should stay within given minimum a maximum distance from the target
  - Follower should anticipate the next most probable movement of the target given its behavioral model
  - Target may have fundamentally different motion characteristics compared to the vehicle
  - Target exhibits a behavior similar to a human scouting a complex terrain

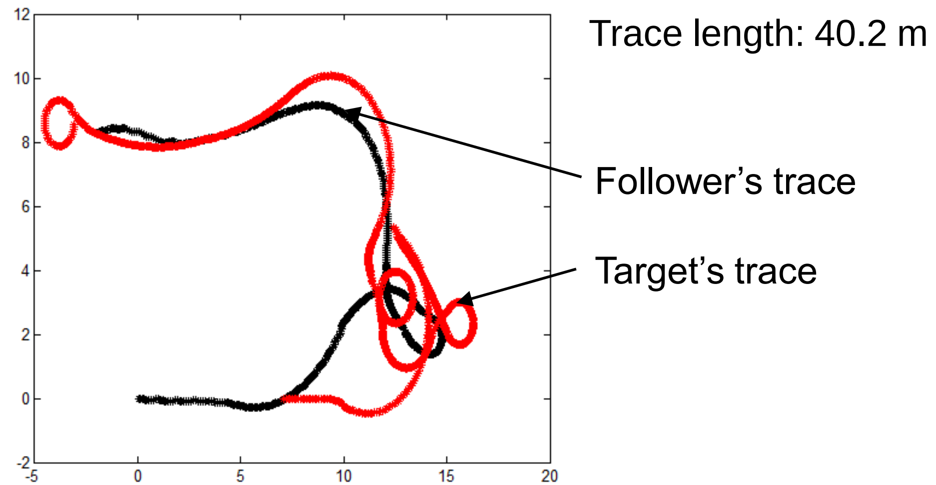


MARYLAND ROBOTICS CENTER
THE INSTITUTE FOR SYSTEMS RESEARCH

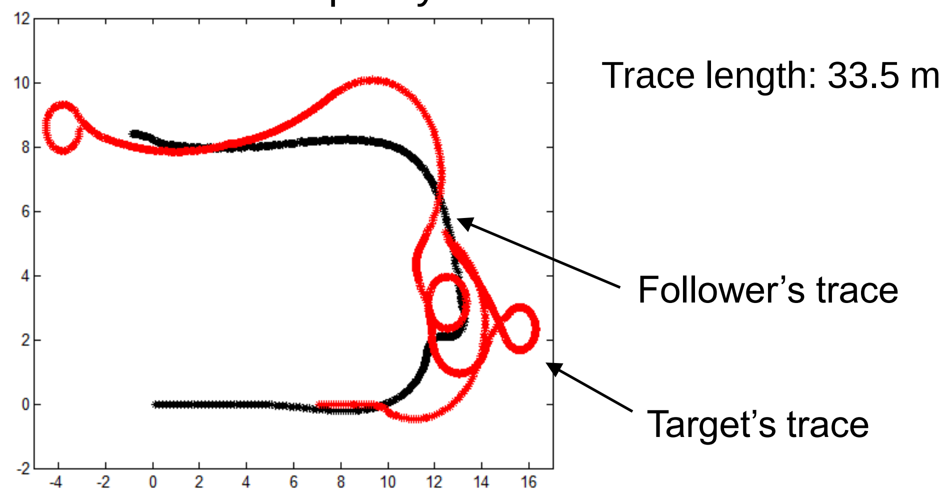
Initial Results

- Evolved a policy for placing motion goal and controlling velocity with respect to the moving target
- Performance of evolved policy exceeds the performance of manually coded rules for following a moving target
  - Target exhibits a scouting behavior

Simple hand-coded policy



Evolved policy



Summary

- We are developing a new approach for automated behavior synthesis for UGVs
- We have shown that useful behaviors can be synthesized using simulations
  - N-Point Turn behavior
  - Follow Target behavior
- The next steps would be
  - Demonstrate that the behavior generation system can be easily reconfigured to generate new behaviors
  - Demonstrate that the synthesized behaviors can be executed on a physical platform without deterioration in performance
  - Demonstrate that the useful complex behaviors can be synthesized automatically

For more information please contact:

Satyandra K. Gupta (skgupta@umd.edu)