Reactive Trajectory Planning and Tracking for Pedestrian Aware Autonomous Driving in Urban Environments

Robert G. Cofield and Rakesh Gupta
Honda Research Institute USA, Inc.
375 Ravendale Drive, Suite 100
Mountain View, CA 94043

Agenda

- Problem
- Input Subsystems
- System Architecture
- Trajectory Planner
- State Transition Graph
- High Level Planner
- Implementation
- Summary

Autonomous Urban Driving

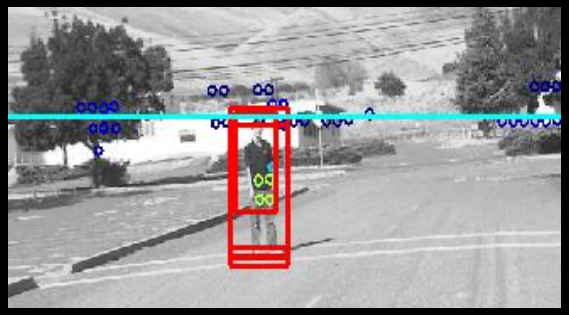
- Travel autonomously from start to destination while honoring traffic laws and avoiding pedestrians in the roadway
- Assume no other traffic on roadway

Desired Behavior:

- Optimize time to travel
- Stop and wait appropriate amount of time at Stop signs
- Stop, slow down, ignore pedestrians as appropriate
  - Crosswalk / jaywalking
  - On-road / off-road
  - Moving / static

Input subsystems

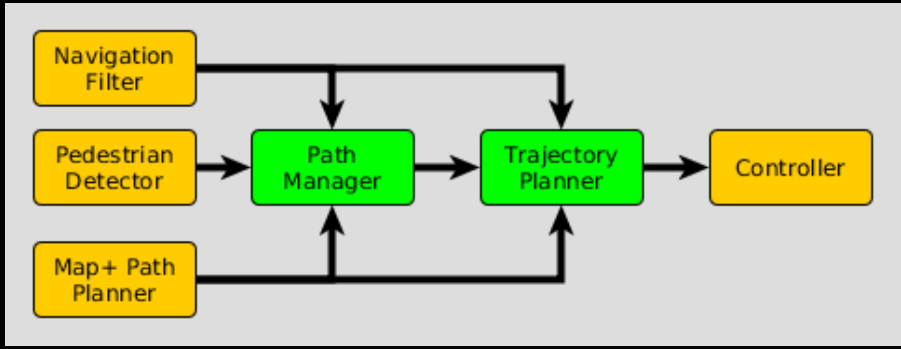
- Path Planner
  - Gives lane level paths
- Pedestrian Detection
  - Using deformable part models with camera, lidar and GPS data
  - Gives location of pedestrian



- Navigation Filter
  - ADMA-G Commercial automotive GPS/INS
  - Receives RTK corrections via cell modem

- High Resolution Map
  - Road extents, lanes, stop signs

System Architecture



Path Manager

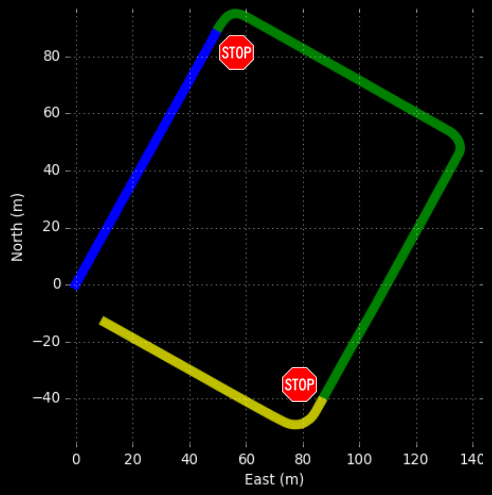
- Implements State Transition Maps
- Stops and restarts vehicle at stop signs
- Governs operation of Trajectory Planner

Trajectory Planner

- Compute medium range trajectories
- Revise trajectories to react to pedestrians
- Input next 2 seconds to controller @ 10 Hz

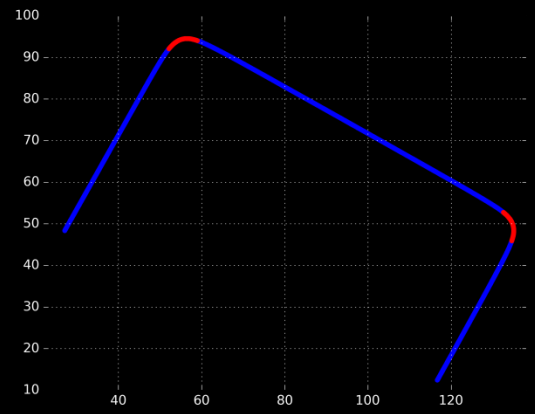
Software Controller

- Governs engine brake, speed and steering

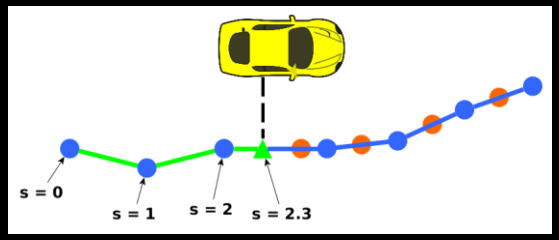


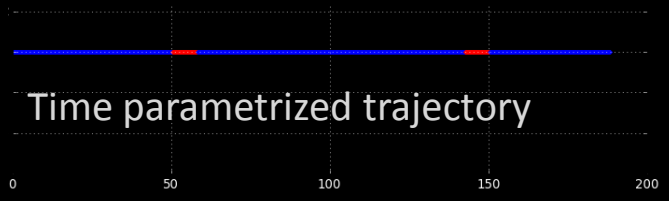
Trajectory Planner

- Solution by parts
  - Break path into segments by speed limit
    - Acceleration and curvature limit speed
    - Legal speed limit
  - Plan trajectory for each segment and unify
- Time parametrize the trajectory
  - Planned time matched to real time by distance travelled so far on the path



Segments in 2D path



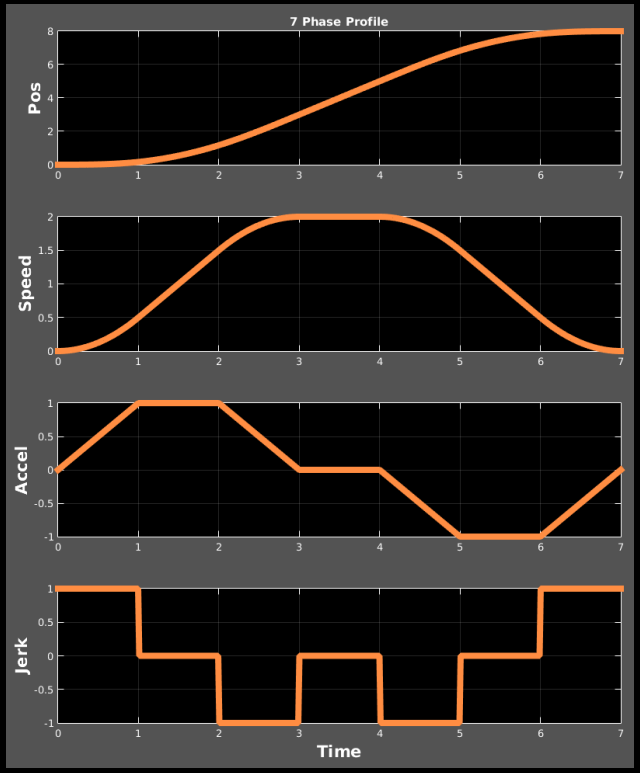


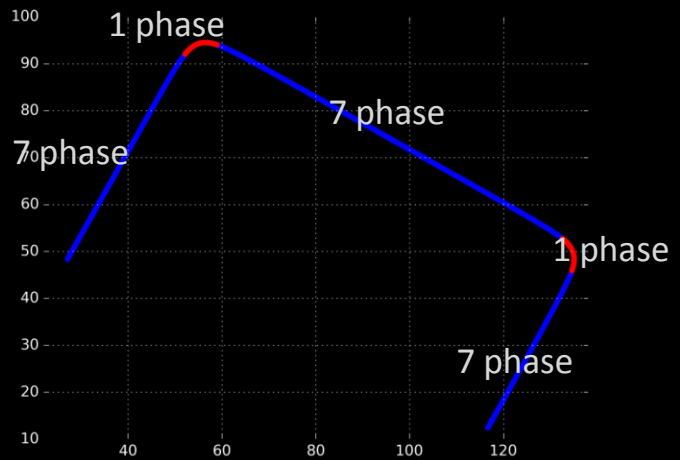
Trajectory Segment Solution

In each segment
- Limit acceleration, jerk, and speed
- Design constant jerk intervals with closed-form solution

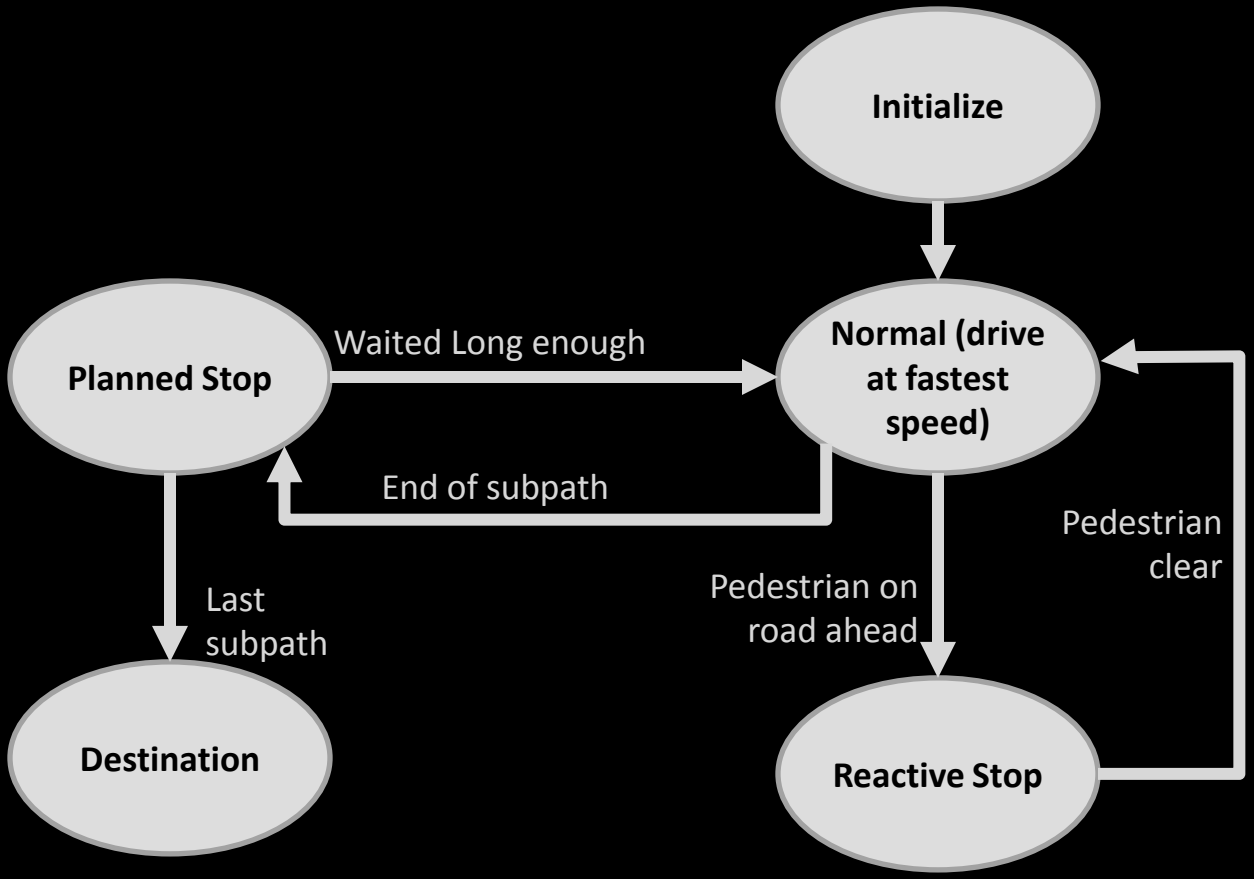
Attempt most time-optimal solution first
- 7 → 6 → 4 → 4R → 3
- No speed change: 1 phase

General Solution
- Integrate j(t)
- Know s(t), v(t), a(t)
- Solve for time intervals

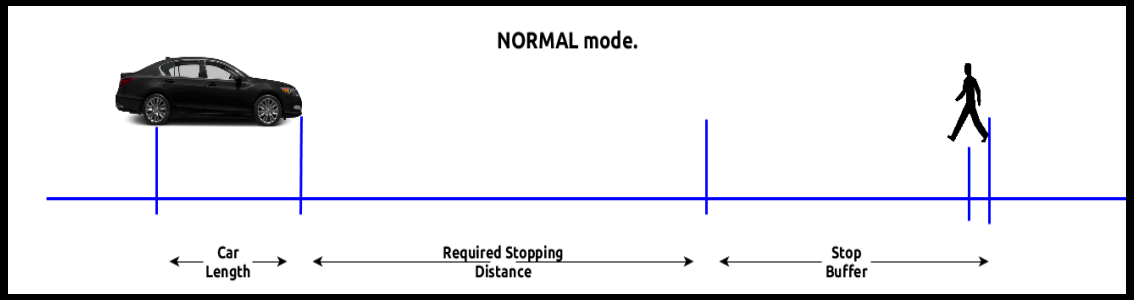




State Transition Graph

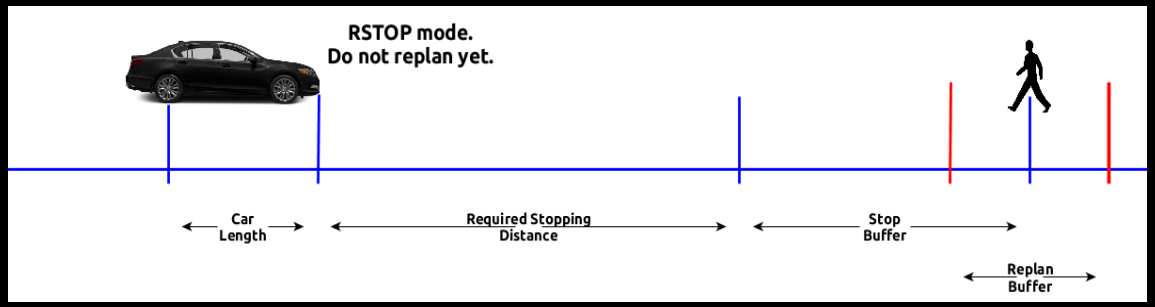


Pedestrian Ahead



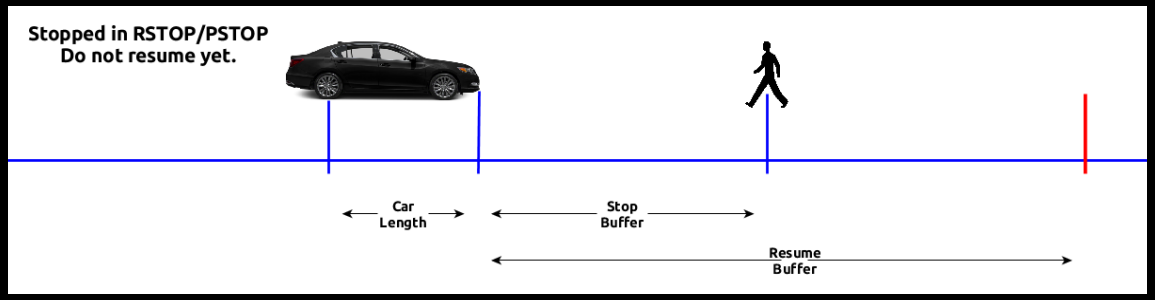
- Project pedestrian position to time parametrized path
- Consider closest pedestrian only
- Execute a 3 phase stop profile when available distance ≤ required distance

Revision of Trajectory



- Revise trajectory if pedestrian moves outside replan buffer

Back to normal driving



- Fully stopped? Transition to normal state when:
  - Pedestrian leaves roadway
  - Pedestrian moves farther past the replan buffer
  - Have stopped for 2 seconds at stop sign
- Slow down without stopping where appropriate

Implementation

- System tested 100+ hours
- Planner online operation @ 10 Hz
- Our planner handled corner cases gracefully
- Demo worked 35 times in closed course with no disengagements

Extensions

- Support self-intersecting, overlapping paths
- Perceive and stop for red-lights
- Handle other roadway traffic, change lanes

Summary

- We built a system for trajectory planning
  - That is reactive to pedestrians
  - Limits jerk, velocity and acceleration

- System has been tested with over 100 hours of testing

Questions?

State Transition Graph