MULTI-ROBOT MOTION PLANNING: A MODIFIED RECEDING HORIZON APPROACH FOR REACHING GOAL STATES

José M. Mendes Filho, Eric Lucet
02/09/2015
{josemagno.mendesfilho, eric.lucet}@cea.fr
**CONTEXT**

- Indoor environment with static obstacles
- Multi-robot system composed by nonholonomic mobile robots
- Robots’ limited perception of the environment
- Robots’ limited communication range
OBJECTIVES

Development of a real-time motion planning algorithm for a multi-robot system

- Real-time generation of collision-free trajectories
- Precise reaching of goal configuration
- Minimization of travel time
Distributed approach over centralized approach
- Drawbacks: less optimal
- Advantages: computation time, security, communication

Local planning over global planning

Base algorithm:

Constrained optimization problems numerically solved

Local planning computation by using a receding horizon approach

Distributed planning performed by postponing the consideration of coupling constraints, meaning inter-robot communication and collision avoidance

Planning in the flat space rather than state space. The solution is denoted $z^*(t)$ where $z(t)$ is the flat output

Solution represented by B-splines (minimal support)
Constraints

- Kinematic model
- Initial state
- Goal state
- Bounded input (control) vector
- Obstacle avoidance
- Inter-robot collision
- Lost of communication

Coupling constraints

Objective function to be minimized

- Travel time
RECEDING HORIZON APPROACH

- Planning horizon \((T_p)\)
- Computation horizon \((T_c)\)
Complete description of the system behavior using the flat output and its derivatives
Constraints

- Kinematic model
- Initial state
- Goal state
- Bounded input (control) vector
- Obstacle avoidance
- Inter-robot collision
- Lost of communication

Coupling constraints

Objective function to be minimized

- Geodesic distance from planned configuration at $T_p$ to the goal configuration
Constraints

- Kinematic model
- Initial state
- **Goal state**
- Bounded input (control) vector
- Obstacle avoidance
- Inter-robot collision
- Lost of communication
- Deviation from $\hat{z}_{b,\tau_k}(t)$

Objective function to be minimized

- Geodesic distance from planned configuration at $T_p$ to the goal configuration
- Stop receding planning when close to the goal configuration
- Compute new time sampling and b-spline parameters
- Change NPLs to consider goal state
Constraints

- Kinematic model
- Initial state
- Goal state
- Bounded input (control) vector
- Obstacle avoidance
- Inter-robot collision
- Lost of communication

Objective function to be minimized

- Time to reach the goal state
$NLP_{b,1,1} \Rightarrow NLP_{b,2,2}$

Constraints

- Kinematic model
- Initial state
- Goal state
- Bounded input (control) vector
- Obstacle avoidance
- Inter-robot collision
- Lost of communication
- Deviation from $\hat{z}_{b,\tau_k}(t)$

Objective function to be minimized

- Time to reach the goal state
### NLP Constraints

<table>
<thead>
<tr>
<th>NLP</th>
<th>Goal Constraints</th>
<th>Coupling Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>$NLP_{b,1,1}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NLP_{b,1,2}$</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>$NLP_{b,2,1}$</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>$NLP_{b,2,2}$</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Update detected obstacles, configuration, $t_{ref} = t$, ...

Solve $NLP_{b,1,1}$

Compute conflict sets

Is conflict sets empty?

No

Exchange intended trajectories with all robots in conflict sets

Solve $NLP_{b,1,2}$

Sync with all robots in conflict sets

$\hat{q}_{\tau_k}, \hat{u}_{\tau_k}$

Wait while $t - t_{ref} < T_c$

Yes

$\hat{q}_{\tau_k}, \hat{u}_{\tau_k} \rightarrow q^*_{\tau_k}, u^*_{\tau_k}$

No

Neighborhood of goal configuration reached?

Yes

No

1

Start
Update detected obstacles, configuration, ...

Compute new $N_s, N_{knots}$

Solve $NLP_{b,2,1}$

Compute conflict sets

Exchange intended trajectories with all robots in conflict sets

Sync with all robots in conflict sets

Is conflict sets empty?

Yes

$q^*_{\tau_k}, u^*_{\tau_k} \leftarrow \hat{q}_{\tau_k}, \hat{u}_{\tau_k}$

Finish

No

$q^*_{\tau_k}, u^*_{\tau_k}

\rightarrow q^*_{\tau_k}, u^*_{\tau_k}$

Solve $NLP_{b,2,2}$
SIMULATIONS

Kinematic simulation example

- Planning horizon: 2.0 s;
- Planning horizon: 0.5 s;
- Time samples for numerically solving the NLPs: 14 s;
- Number of internal knots for B-spline representation: 4;
- 3 robots;
- 3 obstacles;
- Max velocities $[1.0 m/s, 5.0 rad/s]^T$;
Without coupling constraints ($NLP_{b,2,1}$)
With coupling constraints ($NLP_{b,2,2}$)
Algorithm parameters analysis

- Number of time samples ($N_s$)
- Number of internal knots for B-splines ($N_{\text{knots}}$)
- Planning horizon ($T_p$)
- Computation horizon ($T_c$)
- Detection radius of the robots
SIMULATIONS

“Maximum Computation Time”/ Tc (real-time hypothesis)

- An increasing number of $N_s$ increases MCT/Tc at $O(N_s)$ with SLSQP
- An increasing number of $N_{knots}$ increases MCT/Tc at $O(N_{knots}^3)$ with SLSQP

Computation cost behavior, $N_{knots} = 4$

Computation cost behavior, $N_{knots} = 6$

Computation cost behavior, $N_{knots} = 5$
Simulations run on an Intel Xeon CPU 2.53GHz processor
Obstacle penetration area $P$ (obstacle avoidance $P = 0$)

- Penetration area $P$ decreasing as the number of samples $N_s$ increases

Travel time (optimization)

- Travel time decreases with the planning horizon $T_p$
- Travel time decreases with the number of samples $N_s$
- No high influence of the obstacles detection radius $d_{sen}$
CONCLUSIONS & PERSPECTIVES

- Motion planner based on:
  - system flatness property,
  - B-spline parameterization of the flat output
  - SLSQP optimizer

- Enhancement of this cooperative multi-robot systems motion planner, with:
  - termination constraints consideration
  - circle and convex polygon representation of obstacles

- Kinematic simulation with 3 mobile robots in presence of obstacles

- Analyze of Impact of different parameters, to guarantee
  - Real time implementation
  - Obstacles avoidance
  - Travel time optimality

- Work in progress in physics simulation environment, taking into account:
  - vehicle dynamics
  - sensors models
  - communication latency

- Future tests in real conditions with monocycle robots
Thank you!