Reactive Obstacle Avoidance and Control Action Continuity. Application to the ND Algorithm

Jorge Cabrera
Daniel Hernández
Antonio Carlos Domínguez
Josep Isern
Index

• Context and Motivation
• Proposal
• Results
• Future work
Some navigation algorithms use situated action approach to apply different control rules as a function of robot detected environment.

Advantages: instead of a general control rule, specific control rules are generated. The robot can apply different “navigation skills” depending on context (corridor traversing, door crossing, ...).
Problem: control action continuity is normally not guaranteed when robot state evolves between different situations.
The authors have considered this problem introducing the ND+ version of the algorithm, that analyzes the most common transitions and tries to reduce the transition situation effect.

Minguez et. al. "A "divide and conquer" strategy based on situations to achieve reactive collision avoidance in troublesome scenarios”, ICRA, 2004
However: experimental results show that the continuity problem is still present.
The problem is not easy to solve, as a balance between reactivity/safety and smoothness must be obtained.

An example of an inadequate solution is the use of a simple open-loop smoothness filter.

\[
\text{robotComand} = \text{newCommand} \times (1-\text{smoothFactor}) + \text{previousCommand} \times \text{smoothFactor}
\]
Proposed solution: define a kind of membership function (situation certainty) and weight the new generated command and the previous action using this value.

- a) Select current situation applying the decision tree.
- b) Compute situation certainty sCert (normalized).
- c) Weight control action: 
  \[ com = \text{newCom} \times \text{sCert} + \text{prevCom} \times (1-\text{sCert}) \]
The situation certainty is obtained as a combination of fuzzy values computed for each one of the decision criterions implicated.

\[ s_{Cert} = \text{fun}(\text{crit1}, \text{crit3}, \text{crit4}) \]

(fun type multiplicative, minimum, ...)

Proposal
For this certainty computation, each criterion is fuzzified using a ND algorithm parameter, a reference constant and a fuzzification factor.
Example: the first decision criterion (security criterion) is used to determine if there are obstacles inside a safety region around the robot.
Security criterion: fuzzy value computation

![Fuzzy Value Computation Diagram](image)
For the validation of the proposal some experiments have been conducted on Player/Stage simulator using different scenarios.

Besides the graphic-visual comparison, a benchmarking function has been defined for numerical evaluation.

**Benchmarking factors:**
- Time.
- Curvature ratio.
- Accelerations (trans, rot).
Results

Graphical application with debugging tools.
Results

* Scenario 1

ND Standard

ND Modified

![Obstacles and robot trajectory](image1)

![Translation and Rotation Acceleration](image2)

![Situated actions transitions vs. commanded angle](image3)

![Action transitions vs. commanded angle](image4)
Results

* Scenario 1

<table>
<thead>
<tr>
<th></th>
<th>TimeF</th>
<th>CurvF</th>
<th>TAccelF</th>
<th>RAccelF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>0.52</td>
<td>0.39</td>
<td>0.18</td>
<td>0.72</td>
</tr>
<tr>
<td>Modified</td>
<td>0.52</td>
<td>0.41</td>
<td>0.47</td>
<td>0.85</td>
</tr>
</tbody>
</table>

**TABLE I**

**Benchmarking for Scenario 1**
Results

* Scenario 2

<table>
<thead>
<tr>
<th>ND Standard</th>
<th>ND Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results

* Scenario 3

ND Standard

ND Modified
Future work

Analysis of parameters.

Combination of several situations.

Improve benchmarking.

Test on real robots.