Sampling Based Trajectory Planning for Robots in Dynamic Human Environments

Mikael Svenstrup - Department of Electronic Systems, Aalborg University, Denmark

Introduction

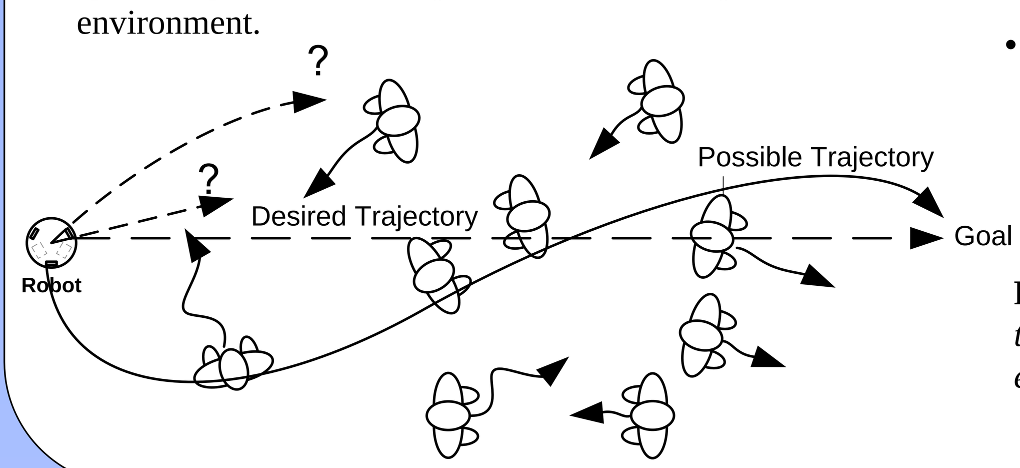
Open-ended human environments, such as pedestrian streets, hospital corridors, train stations etc., are places where robots start to emerge. Hence, being able to plan safe and natural trajectories in these dynamic environments is an important skill for future generations of robots.

A new approach for adaptive motion control is developed based on the human to human proxemics described by Hall [1] (see Fig. 2). This control strategy is inspired by [2] and utilizes a cost function centred in each person. Summing the cost function for all persons gives a potential field landscape through which a trajectory must be generated.

A modified Rapidly exploring Random Tree (RRT) algorithm is used to minimise the cost of traversing the potential field landscape.

Objectives

The objective is to plan a trajectory through a dynamic human environment, where the goal is not a specific point, but to get forwards through the environment.



- The trajectory has to be safe and natural for the humans in the environment.
- The trajectory must take into account the dynamics of the environment, i.e. the predicted human motion.
- The trajectory must be adjusted online if the environment changes.

Fig. 1: *The objective is to generate a robot trajectory, through a dynamic human environment.*

Methods

Potential field derivation:
A sum of four person centred bi-variate Gaussian distributions is used to represent the desired spatial motion of the robot, with respect to each human (see [4]). The distributions represent a cost function (potential field) for where the robot should be. The cost function is derived using Hall's proxemic zones [1] (see Fig. 2), which indicates where humans likes others to be. The potential field around a person can be seen in Fig 3. Summing over all persons in an environment, yields a potential field as shown in Fig. 4, where three potential robot trajectories are also shown.

Algorithm 1 Modified RRT for human environments

```
RRTmain()
Tree = seed(oldBestTrajectory)
while (Nnodes < maxNodes) and (t < tMax) do
    q.target = SampleTarget()
    q.nearest = NearestVertex(Tree , q.target)
    q.new = CalculateControlInput(q.nearest,q.target)
    if PruneNode(q.new) == false then
        Tree.add(q.new)
    end if
end while
return BestTrajectory(Tree)
SampleTarget()
if Rand < GoalSamplingProb then
    return q.goal
else
    return RandomConfiguration()
end if
```

Trajectory cost minimisation:
To minimise the cost of traversing the potential field, a modified version of a Rapidly exploring Random Tree (RRT) algorithm is used [3]. Given the dynamic nature of the problem, robotic kinodynamic and nonholonomic constraints must also be considered. The standard RRT is shown in Algorithm 1 to the left, where the modified parts are indicated with red. The standard RRT trajectory planner is modified in the following way:

- The planner runs in configuration-time (C - T) space, where moving obstacles are static.
- Person motion models are used to predict trajectories of persons into the future
- When expanding the RRT, a 2. order dynamic motion model of the robot is used to simulate the robot motion and calculate a control input, which drives the robot towards the sampled vertex.
- RRT vertices are pruned where the cost is too high.
- A "best trajectory" is selected based on the cost of traversing the trajectory and not based on reaching a goal.
- Using a Model Predictive Control scheme, a new "best trajectory" is calculated on-line, while executing the current planned trajectory.
- After a short time period, the current trajectory is replaced by the new "best trajectory". And again the new RRT is initialised by seeding with the new "best trajectory".
- The environment potential changes over time according to the desired destination, which also make the algorithm robust to local minima.

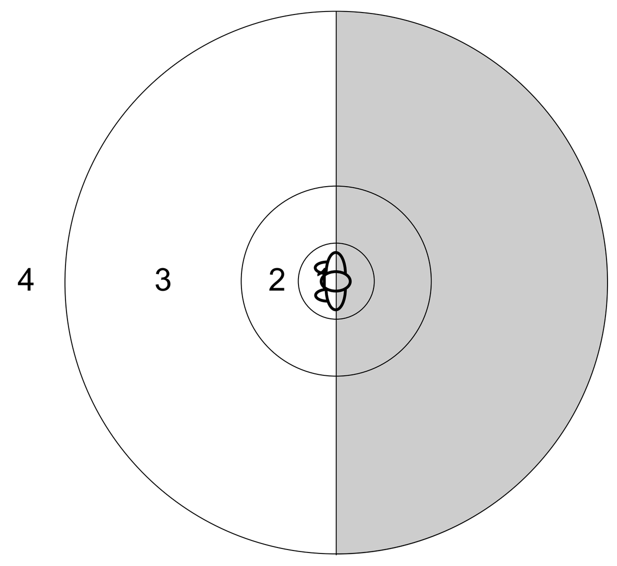


Fig. 2: *The Hall zones in the area of a human:*
Zone 1: Intimate Zone, < 0.45 m
Zone 2: Personal Zone, 0.45 – 1.2 m
Zone 3: Social Zone, 1.2 – 3.6 m.
Zone 4: Public Zone, > 3.6 m

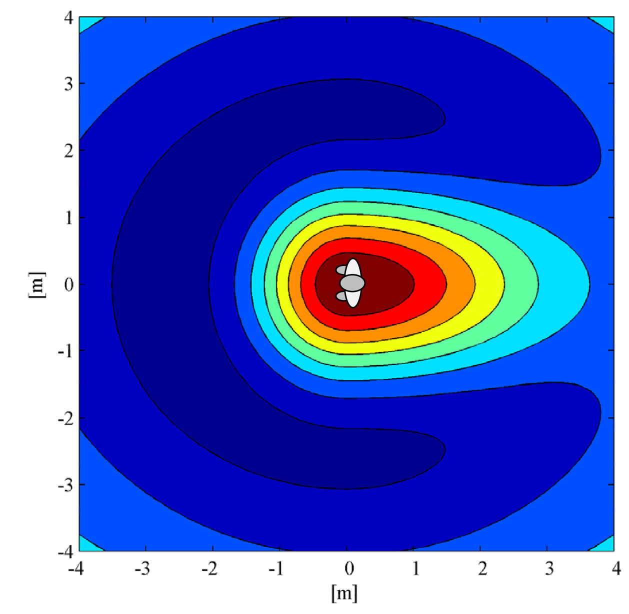


Fig. 3: *Potential field around a person. The robot is not allowed to towards the red areas. This is for example directly behind a person. The robot is attracted towards the dark blue areas.*

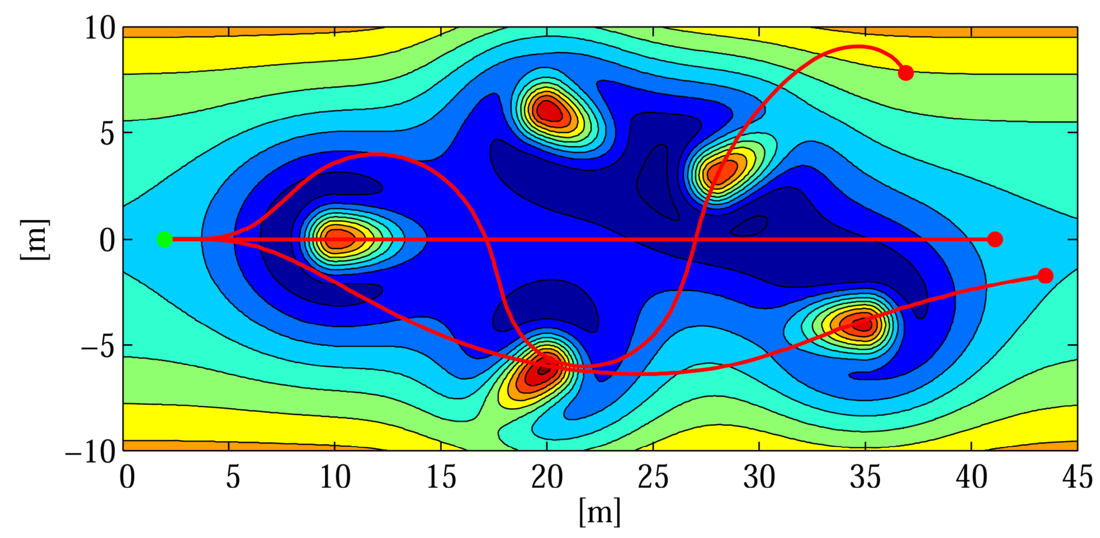


Fig. 4 *Illustration of the potential field with five persons. Three possible trajectories are shown. Even though the trajectories goes through persons, it might not necessarily be bad trajectories, since the persons might have moved away when the robot gets there.*

Simulations and Results

In Fig. 5 an example of an RRT through the environment from Fig. 4 is shown. It is seen that the RRT covers the configuration space well, and especially in the near area. After 10-15 m the density of the tree is lower, but since a new trajectory is already calculated when the robot gets this far, it is less important. The green line on the figure shows the best, i.e. the minimum cost trajectory.

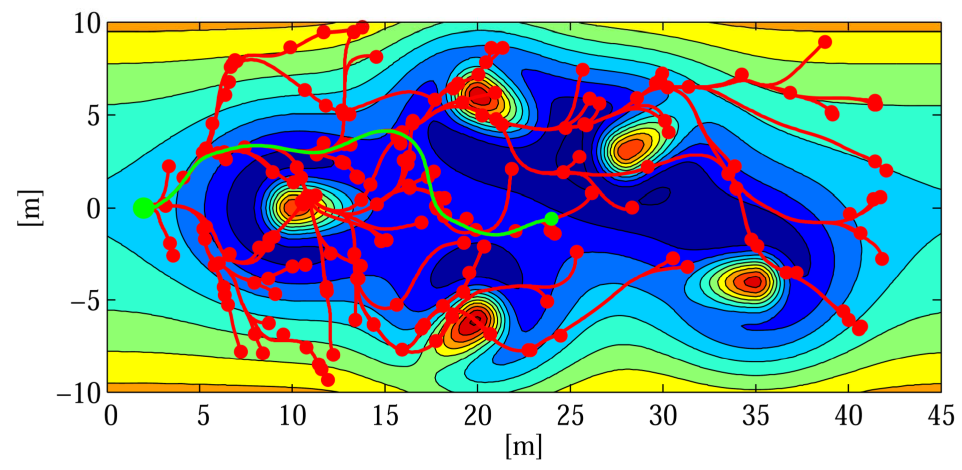


Fig. 5: *An RRT for a robot starting at (2, 0) The vertices are the red dots, and the lines are the simulated trajectories. The green trajectory is the minimum cost trajectory.*

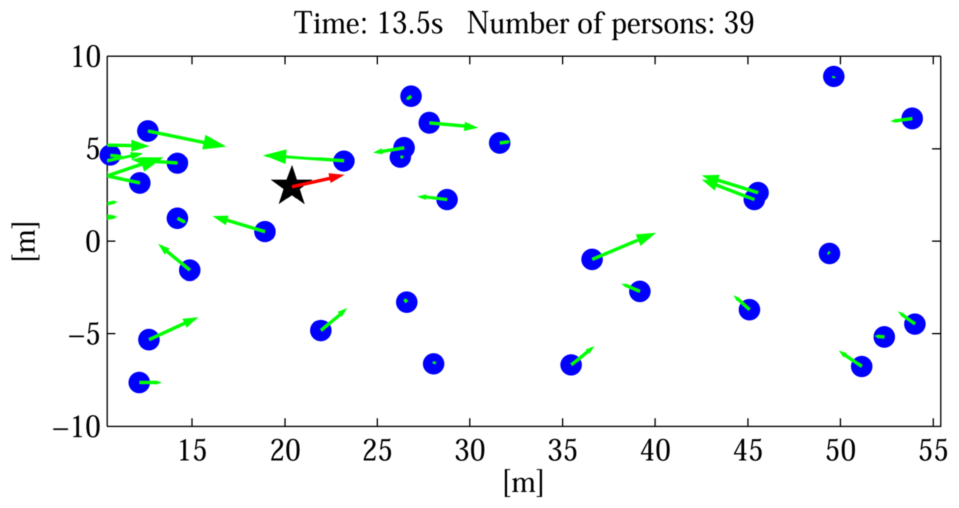


Fig. 6: *A scene from a simulation. The blue dots are persons, with their corresponding current velocity vectors. The star square is the robot.*

The RRT algorithm has been implemented and demonstrated in an experiment, where the robot plans the trajectory through a simulated pedestrian street. The environment has been simulated with a Poisson distributed number of persons entering and leaving the environment and with the persons not taking into account the motion of the robot.

Fig. 6 shows a scene from the simulation. Through 50 simulations of a one minute period, the robot never runs into any persons. Fig. 7 shows in which zone the closest person is how much of the time. It can be seen that in approximately 98% of the time the robot keeps at least 1,2m to the nearest person, which is the boundary between the personal and the social zone according to Hall's social distances [1]. This demonstrates that the algorithm is able to plan a trajectory, which is safe and natural, through an uncertain open-ended human environment.

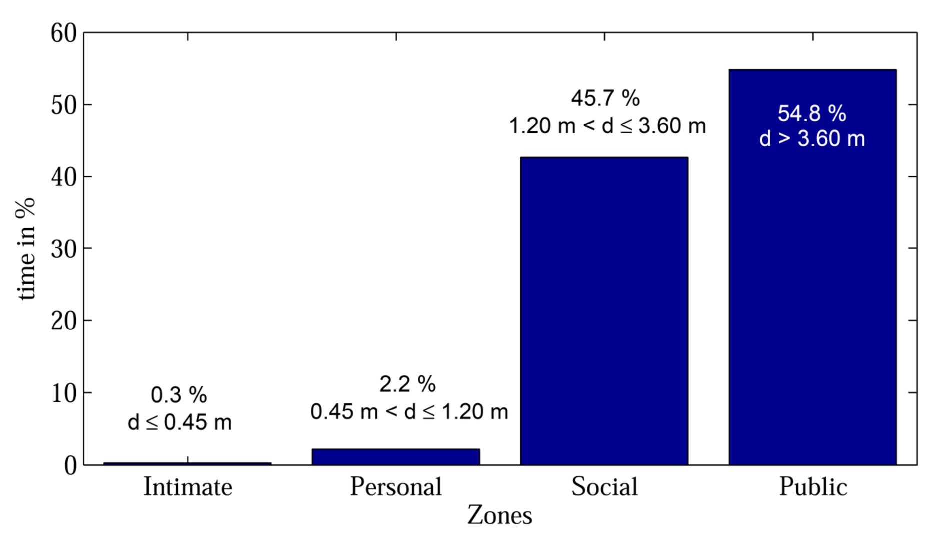


Fig. 7: *The figure shows how large a part of the time the closest person to the robot has been in each zone. The robot should try to stay out of the personal and intimate zones, which is done most of the time.*

Discussion and Conclusions

The simulations of the trajectory generation algorithm have shown that the robot is capable of navigating in a highly dynamic environment, which in this case is populated with humans. Together with a dynamic model of the robot, an MPC scheme is used to enable the planner to continuously plan a safe and reachable trajectory on an on-line system.

The algorithm is challenged when the environments become very densely populated, but so are humans. Humans react by mutual adaptation and allowing violation of the social zones. This is not done here, where the robot takes on all the responsibility for finding a safe trajectory. But even so, the robot manages to avoid human contact.

Future work include real life experiments, and incorporation of human to human motion correlation into the algorithm.

References

[1] E. T. Hall. A system for the notation of proxemic behavior. American anthropologist, 65(5):1003-1026, 1963
[2] E. A. Sisbot, R. Alami, T. Simeon, K. Dautenhahn, M. Walters, S. Woods, K. L. Koay, and C. Nehaniv. *Navigation in the presence of humans.* In IEEE-RAS International Conference on Humanoid Robots, Tsukuba, Japan, December 2005.
[3] S. LaValle. Planning algorithms. Cambridge Univ Pr, 2006.
[4] M. Svenstrup, S. T. Hansen, H. J. Andersen, and T. Bak, *Pose estimation and adaptive robot behaviour for human-robot interaction.* In, IEEE International Conference on Robotics and Automation, Kobe, Japan, May 2009.

AALBORG UNIVERSITY