# Smooth and Efficient Obstacle Avoidance for a Tour Guide Robot

# Smooth and Efficient Obstacle Avoidance for a Tour Guide Robot

Roland Philippsen and Roland Siegwart

Autonomous Systems Lab, EPFL-I$^2$S-ASL
Swiss Federal Institute of Technology Lausanne (EPFL)
{roland.philippsen, roland.siegwart}@epfl.ch

***Abstract*— We present the local path planning and obstacle avoidance method used on the autonomous tour-guide robot RoboX. It has proven its value during a 5 month operation of ten such robots in a real-world application, a very crowded exhibition. Three known approaches (DWA, elastic band, NF1) have been integrated into a system that performs smooth motion efficiently, in the sense of computational effort as well as goal-directedness. Apart from modifications to the DWA and the elastic band, we present the formulations that allow this fusion.**

## I. INTRODUCTION

A tour-guide robot has to be able to move autonomously, acquire the attention of the visitors and interact with them efficiently. Usually, the environment is known and accessible, but the visitors make it highly cluttered and dynamic.

This paper presents an implementation of path planning and obstacle avoidance for RoboX (fig. 1), an autonomous tour-guide robot developed at the Autonomous Systems Lab for Expo.02 (the Swiss national exhibition that took place from May $15^{th}$ to October $20^{th}$ 2002). RoboX's navigation subsystem comprises an embedded Power PC G3 at 380 MHz running the XO/2 real-time operating system, two SICK laser scanners, 8 contact sensors with soft bumpers and a differential drive architecture.

The Robotics pavilion where RoboX operated was visited by 400 persons per hour, at a density of 100 visitors on $300m^2$. This can be likened to a railway station at rush-hour (fig. 2). RoboX had to move with, against and across the flow of people to accomplish its tour-guiding task. And it should never stop moving, lest the visitor lose interest.

## II. AIM AND APPROACH

A tour-guide robot faces certain requirements. The collision risk must be low and the eventual effects of a collision be harmless. Smooth motion is important, as visitors anticipate movement when they follow the guide. The obstacle avoidance control loop should be fast in order to not only run in real-time, but also leave enough processing resources to other modules such as localization, sensor acquisition, web server and motor control.

Path planning and obstacle avoidance have been treated in previous works. Similar to [1], [3], [11], we evaluated and chose among existing algorithms and combined


Fig. 1. RoboX was developed at the Autonomous Systems Lab.

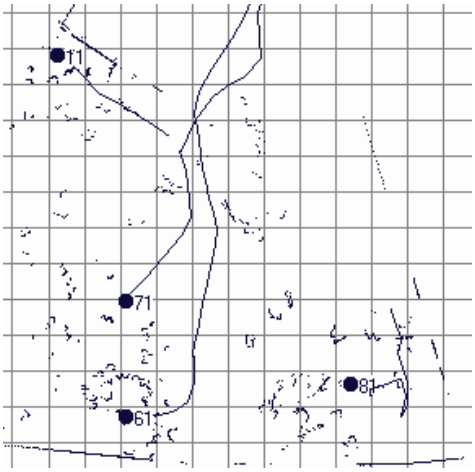


Fig. 2. Laser scanner snapshot illustrating how cluttered the Robotics pavilion is. Raw data from all robots transformed to their respective poses. Grid resolution is $1m$.

them such that their drawbacks cancel out as much as possible and their advantages cover our needs. Our main contribution here is a consistent fusion at the interfaces between the sub-tasks without compromising any component's functionality. Also, the used methods were modified to either improve a component or their interaction. For instance, real-time performance and memory usage was optimized without effect on other components by using transparently compressed look-up tables.

The task of the motion planner was divided into two layers, one to supply a path plan, the other to follow it while taking into account the exact geometry of the robot, its kinematics, and the dynamics of its actuators.

From the candidates for the reactive level [2], [4], [5], [12], [13], the dynamic window (DWA) was chosen because of its physically meaningful representations (actuator speeds and accelerations, robot geometry in work-space) and one-step calculation of rotational and translational speed.

Among the works [3], [6], [7], [9], [10] studied for the planning layer, or for their ability to solve reactive and planning problems at the same time, an elastic band approach was chosen for RoboX. Its path representation is compact and physically meaningful, smooth and designed to accommodate environment dynamics. We use the NF1 [8] for generating the initial plan and rely on the elastic band to compensate for its drawbacks (grazing of obstacles, unsmooth paths). Elastic band and NF1 taken together are referred to as local path planning on RoboX, whereas global path planning refers to searching a graph-based a-priori map.

Fig. 3 shows the flow of information and control in the motion planner. The DWA is a real-time (RT) control loop at $10Hz$, the elastic band is a non-RT loop that typically runs at $5Hz$ and the NF1 is calculated upon request and can take up to $0.5s$. Non-RT execution times depend on the path length, the clutter in the environment and the overall system load.

## III. DYNAMIC WINDOW

The DWA generates actuator commands such that the robot does not collide with obstacles, the commands do not violate the dynamic capabilities of the actuators and the robot follows the elastic band.

In our implementation, the robot shape's (polygon) is defined at startup (instead of being hard-coded). Additionally, a significant speed-up and a predictable maximum cycle time have been achieved by calculating a look-up table for the collision prediction, also during startup. This vital part of the DWA would otherwise constantly require expensive computations of varying numbers of intersections between circles and lines, up to 90000 on RoboX. A similar idea can be found in [11].

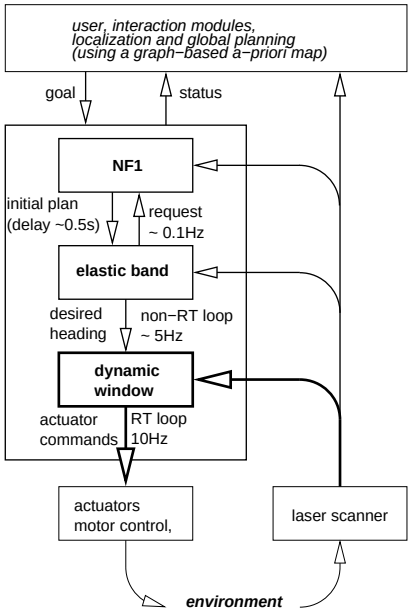


Fig. 3. Diagram of motion planning and control loops and how they are integrated into the overall navigation architecture. Deliberative levels specify a global goal position to the motion planner and can query its status. The output is in the form of actuator commands sent to the motor control level.

### A. Collision Prediction

The collision prediction in [5] calculates the distance to travel before hitting an obstacle. This is not applicable to pure rotations because any collision would seem instantaneous. We solved this problem by using the time until collision, which does not present such a singularity. As a side effect, the same geometric distance appears closer at high speeds, effectively adding a buffer distance proportional to speed.

### B. Velocity Space

RoboX is a differential drive robot. The kinematic model and its inverse are given in (1) and (2).

$$\vec{v}(\dot{\vec{q}}) = \begin{bmatrix} \dot{s}(\dot{q}_l, \dot{q}_r) \\ \dot{\theta}(\dot{q}_l, \dot{q}_r) \end{bmatrix} = \begin{bmatrix} \frac{R_{\text{wheel}}}{2}(\dot{q}_l + \dot{q}_r) \\ \frac{R_{\text{wheel}}}{D_{\text{base}}}(\dot{q}_r - \dot{q}_l) \end{bmatrix} \quad (1)$$

$$\dot{\vec{q}}(\vec{v}) = \begin{bmatrix} \dot{q}_l(\dot{s}, \dot{\theta}) \\ \dot{q}_r(\dot{s}, \dot{\theta}) \end{bmatrix} = \begin{bmatrix} (\dot{s} + \frac{D_{\text{base}}}{2}\dot{\theta})/R_{\text{wheel}} \\ (\dot{s} - \frac{D_{\text{base}}}{2}\dot{\theta})/R_{\text{wheel}} \end{bmatrix} \quad (2)$$

where $R_{\text{wheel}}$ is the radius of the drive wheels, $D_{\text{base}}$ is the wheel base, $(\dot{q}_l, \dot{q}_r)$ are the rotational speeds of the left and right wheel and $(\dot{s}, \dot{\theta})$ are the translational and rotational speeds of the robot. On RoboX, $R_{\text{wheel}} = 0.09m$ and $D_{\text{base}} = 0.521m$

Using the actuator space $\dot{\vec{q}} = (\dot{q}_l, \dot{q}_r)$ properly models the acceleration and speed limits of the actuators (fig. 4), as opposed to the usual $\vec{v} = (\dot{s}, \dot{\theta})$.

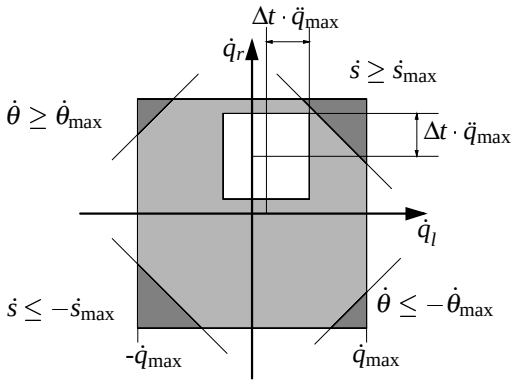


Fig. 4. Definition of the dynamic window: $(\dot{q}_l, \dot{q}_r)$ are the actuator speeds, $\dot{q}_{\max}$ is the maximum actuator speed, $\ddot{q}_{\max}$ the maximum actuator acceleration, $\Delta t$ the time-step of the control loop, $(\dot{s}, \dot{\theta})$ the robot speed in euclidean space, $\dot{s}_{\max}$ and $\dot{\theta}_{\max}$ are global speed limits. Dark gray regions are forbidden by $\dot{s}_{\max}$ or $\dot{\theta}_{\max}$, the light gray region are the available speeds and the white square is an example of reachable actuator speeds at a given moment. Admissible speeds are those inside the white square which would not lead to a collision. On RoboX, $\dot{q}_{\max} = 6.5 rad/s$, $\ddot{q}_{\max} = 6.5 rad/s^2$, $\Delta t = 0.1 s$, $\dot{s}_{\max} = 0.6 m/s$ and $\dot{\theta}_{\max} = 2.5 rad/s$

### C. Objective Functions

The DWA chooses among admissible commands by maximizing an objective function on the sampled $(\dot{q}_l, \dot{q}_r)$ space. The usual sub-objectives are used: Heading, speed and clearance.

The heading objective $w_{\text{head}}$ makes the robot follow the elastic band or, once the goal radius has been reached, orient itself along a specified direction.

The speed objective $w_{\text{speed}}$ can be switched at run-time: High objective values for high forward speeds to move forward, preferring high backward speeds make the robot reverse, and high objective values for low translational speeds to turn on the spot. This is used to follow paths based on holonomic assumptions while consistently using the DWA to generate actuator commands.

The clearance objective $w_{\text{clear}}$ tends to maximize the space between robot and obstacles. It measures by how much the collision prediction exceeds the braking time for $\dot{\vec{q}}$, see (3) and (4).

$$w_{\text{clear}}(\dot{\vec{q}}) = \begin{cases} 0 & \text{if } t_{\text{col}} \leq T(\dot{\vec{q}}) \\ \frac{t_{\text{col}} - T(\dot{\vec{q}})}{T_{\max} - T(\dot{\vec{q}})} & \text{if } T(\dot{\vec{q}}) < t_{\text{col}} < T_{\max} \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

$$T(\dot{\vec{q}}) = max(\dot{q}_l, \dot{q}_r) / \ddot{q}_{\max} \quad (4)$$

where $t_{\text{col}}$ is the collision prediction for $\dot{\vec{q}}$ given the current sensor readings, $T(\dot{\vec{q}})$ is the braking time when traveling at $\dot{\vec{q}}$ and $T_{\max}$ is the braking time at maximum speed. Speeds where $w_{\text{clear}}$ is zero are constraints: They are flagged as non-admissible because they would surely lead to collision.

The overall objective $w^*$ (5) is a weighted sum of the sub-objectives. The next motion command (6) is chosen to maximize $w^*$.

$$w^* = \alpha_{\text{clear}} w_{\text{clear}} + \alpha_{\text{speed}} w_{\text{speed}} + \alpha_{\text{head}} w_{\text{head}} \quad (5)$$

$$\dot{\vec{q}}^* = \arg(\max_{\dot{\vec{q}}}(w^*)) \quad (6)$$

where $\alpha_{\text{clear}}$, $\alpha_{\text{speed}}$ and $\alpha_{\text{head}}$ define the relative weights of the sub-objectives. At Expo.02, we used $\alpha_{\text{clear}} = 0.5$, $\alpha_{\text{speed}} = 0.1$ and $\alpha_{\text{head}} = 0.1$.

### D. Look-Up Tables

Two layers of look-up operations are involved (fig. 5). Clearance-lookup objects are instantiated only for those cells in the obstacle grid which actually can lead to collision predictions that are inferior to $T_{\max}$. Other cells, as well as those contained within the robot outline, contain null pointers and don't use up RAM.

The maximum memory requirements for this structure on RoboX are $3.4 Mb$ for the combination of obstacle grid and clearance look-up, plus $7.2 Mb$ for the quantizer tables. Less than half of this is actually used due to the mentioned null-pointers and the fact that very few quantizer tables need more than 100 entries of the 256 available.

Depending on the number of obstacles and the current speed, calculating one iteration of the DWA on RoboX can take up to $22 ms$, with a mean value of $7.5 ms$.

The center points of the cells are used to calculate the time until collision, which can lead to over-estimations if the actual point lies closer to the outline. So, the robot outline is grown by half the cell diagonal to counter this. Remaining collision prediction errors are always under-estimations of the time until collision: In the worst case, the robot stops too early, but never too late.

## IV. ELASTIC BAND

The elastic band is responsible for path representation, adapting the plan to the robot's movement and changes in the environment.

To reduce the computational load, we use euclidean distances and linear forces. This is acceptable because the DWA ensures the dynamic, kinematic and geometric constraints. More importantly, we mask out certain obstacles to make the algorithm more stable in highly dynamic environments, where movement far from the robot can "snap" the elastic band unnecessarily.

### A. Obstacle Masking

The elastic band is composed of bubbles $b_i$ which represent the path and associated free space. They are circles defined by center $\vec{c}_{bi}$, radius $r_{bi}$ and masking distance $D_{mi}$. The latter has a linear relationship to the bubble's position $L_i$ along the path, see (7) and (8).

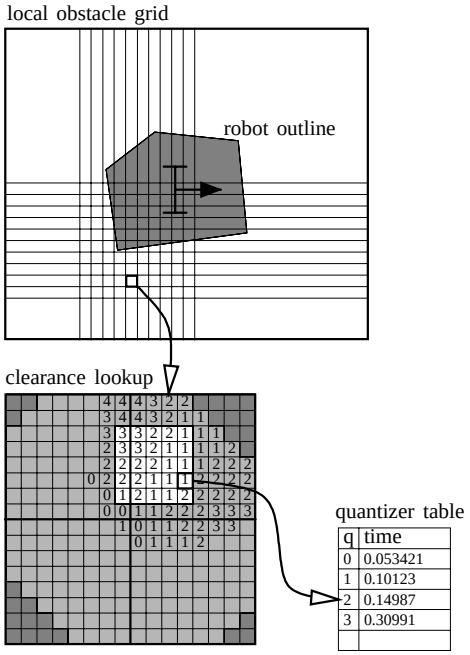


Fig. 5. Look-up operations in the dynamic window. Laser points are entered into a two-dimensional local obstacle grid. For each occupied cell we iterate over its associated clearance look-up, a two-dimensional table corresponding to the dynamic window. The clearance look-up contains 8-bit indexes into an associated quantizer table storing the actual collision times. Indexes are generated using a Lloyd-Max algorithm.

$$L_i = \sum_{j=1}^{i} \| \vec{c}_{bj-1} - \vec{c}_{bj} \| \tag{7}$$

$$D_{mi} = D_{m,\max} \cdot \begin{cases} 0 & \text{if } L_i \le L_{\min} \\ 1 & \text{if } L_i \ge L_{\max} \\ \frac{(L_i - L_{\min})}{L_{\max} - L_{\min}} & \text{otherwise} \end{cases} \tag{8}$$

where $L_{\min}$ and $L_{\max}$ define the cumulative path lengths over which $D_{mi}$ stretches and $D_{m,\max}$ is the maximum distance at which readings can be ignored. At Expo.02, we used values of $L_{\min} = 2.0m$, $L_{\max} = 8.0m$ and $D_{m,\max} = 8.5m$.

Obstacles are modeled as points $\vec{p}_j = (x_{pj}, y_{pj})$. Each $b_i$ has an associated set of masked obstacles $\{\vec{p}_{m,ij}\}$ defined in (9). The obstacle $\vec{p}_i^*$ closest to $b_i$ is found (10) and determines $r_{bi}$ (11).

$$\{\vec{p}_{m,ij}\} = \{\vec{p}_j : \| \vec{c}_{bi} - \vec{p} \| > D_{mi}\} \tag{9}$$

$$\vec{p}_i^* = \arg\Big(\min_{\vec{p} \in \{\vec{p}_{m,ij}\}} \| \vec{c}_{bi} - \vec{p} \|\Big) \tag{10}$$

$$r_{bi} = \min_{\vec{p} \in \{\vec{p}_{m,ij}\}} \| \vec{c}_{bi} - \vec{p} \| \tag{11}$$

### B. Linearisations

The expressions for the internal (12) and external (13) forces are linear. They determine the iterative movement $\vec{c}_{bi,t+1} = \vec{c}_{bi,t} + \Delta\vec{c}_{bi}$ of the bubbles (14). The first bubble follows the robot's position and the last bubble is immobile at the goal.

$$\vec{f}_{\text{int},ij} = \alpha_{\text{int}} \cdot \begin{cases} 0 & \text{if } \| \vec{c}_{bi} - \vec{c}_{bj} \| \le \varepsilon \\ \frac{\vec{c}_{bj} - \vec{c}_{bi}}{\| \vec{c}_{bi} - \vec{c}_{bj} \|} & \text{otherwise} \end{cases} \tag{12}$$

$$\vec{f}_{\text{ext},i} = \alpha_{\text{ext}} \cdot \begin{cases} 0 & \text{if } r_{bi} \le \varepsilon \text{ or } r_{bi} \ge r_{\lim} \\ \frac{r_{\lim} - r_{bi}}{r_{bi}} (\vec{c}_{bi} - \vec{p}^*{}_i) & \text{otherwise} \end{cases} \tag{13}$$

$$\Delta\vec{c}_{bi} = \alpha_{\text{tot},i} \cdot (\vec{f}_{\text{int},i,i-1} + \vec{f}_{\text{int},i,i+1} + \vec{f}_{\text{ext},i}) \tag{14}$$

$$\alpha_{\text{tot},i} = \begin{cases} 1 & \text{if } r_{bi} > r_{\lim} \\ \frac{r_{bi}}{r_{\lim}} & \text{otherwise} \end{cases} \tag{15}$$

where $r_{\lim}$ is a parameter that defines the distance at which the elastic band starts to react to obstacles, $\alpha_{\text{int}}$ and $\alpha_{\text{ext}}$ are parameters that define how strong these forces are, $\alpha_{\text{tot},i}$ makes smaller bubbles move smaller amounts and $\varepsilon$ is used to avoid divisions by zero. During Expo, we used $r_{\lim} = 1.6m$, $\alpha_{\text{int}} = 0.1$, $\alpha_{\text{ext}} = 0.1$ and $\varepsilon = 10^{-9}$.

### C. NF1

In order to generate a new path plan, the NF1 grid is initialized to the current situation, its wavefront-propagation calculated and the cell centers leading from robot to goal transformed to the global frame of reference (fig. 6). Only current laser readings are used to initialize the grid obstacles, which are enlarged by the robot radius prior to wavefront-propagation. If no path is found, replanning is tried again with a larger width.

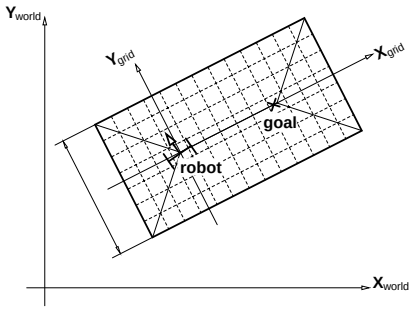


Fig. 6. Positioning of the NF1 grid. Its width is a parameter, its length adapts to the situation.

## V. INTEGRATION

### A. Perception

The main sensor input for the motion planner comes from the laser scanners. Bumpers are also used, but only to stop the robot if it touches something. The robot also stops if a laser reading indicates an object inside the robot's outline.

All calculations are based on the latest sensor data only, without any memory effects. This is acceptable in the Robotics pavilion because the layers above the motion planner provide subgoals which lie close to each other along a topologically feasible path and most obstacles are visitors and tend to move out of the robot's way sooner or later.

Having more than one robot in the same space adds the problem of how robots can detect each other. Each RoboX has the laser scanners at the same height. This is problematic because even if another robot is detected, it's not seen as the actual outline. We solved this by attaching reflector bands in the blind zone between the laser scanners. The intensity of the laser data is used to detect the reflectors on other robots, which are then injected as virtual "ghost" points into the laser data to represent the approximate shape of surrounding robots. An example can be seen in fig. 7.

### B. Motion Planner

The motion planner provides a convenient interface for configuring and using the local path planning and obstacle avoidance system. It orchestrates the components to achieve the required overall behavior by querying the elastic band about the best heading and feeding this information to the DWA. If no band exists, it falls back to a purely local algorithm using the direct line from robot to goal. It is also the motion planner's task to switch between pure rotation and forward movement. Replanning on the other hand is initiated independently by the elastic band.

When the elastic band cannot guarantee sufficient clearance along the path, it snaps and a new plan has to be generated. We want RoboX to do so without halting its movement. Here, we take advantage of environment properties: The band usually snaps because visitors move into it from both sides, and a couple of updates later the band can become valid again if they move on. So, RoboX fires a background replanning thread but continues to use a snapped band, confident that the DWA keeps it from colliding. As soon as the new plan is available, it is used instead of the broken band. This process is illustrated in fig. 8.

## VI. RESULTS

RoboX moves smoothly through crowded exhibitions. Some collisions still happen. They are mostly due to objects that are out of the laser's field of view (especially feet and protruding parts of buggies and wheelchairs), in which case the robot simply stops as soon as contact is detected. Two main failure modes remain. One can appear when moving RoboX from the main exhibition hall into a narrow corridor, the other stems from the simplistic detection and modeling of other robots.

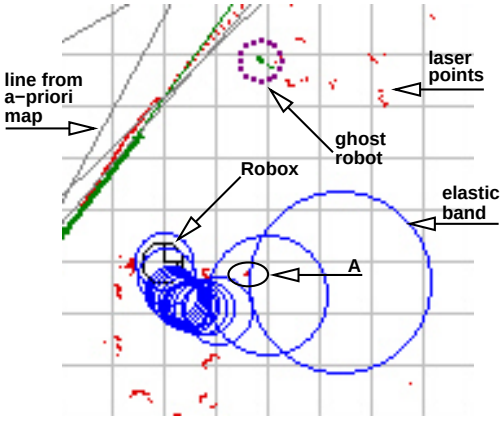


Fig. 7. An image generated on the robot shows laser scanner data, a "ghost robot" constructed around a reflector of another RoboX, the elastic band and lines from the map. The points in region A are masked, otherwise the second to last bubble would be smaller.

Moving into a corridor, RoboX sometimes gets stuck at a corner. This seems to be due to the approximation of holonomic movement in the path representation, combined with the position of the blind zone between the sensors. The path leads RoboX towards the corner such that it disappears in the blind zone. Further along the path, the corner sometimes reappears inside the modeled robot outline, which makes it stop.

Robots can also get stuck when two of them move towards each other while not detecting the reflectors, which can be hidden behind visitors. When the reflectors are later detected, the resulting ghost robots appear instantly, often inside the outline and make RoboX stop.

During Expo.02, the ten RoboX have accumulated an operation time of 13313$h$, of which 9415$h$ were spent in movement for a total distance of 3315$km$ and met approximately 686000 visitors. The maximum speed of the robot was set to a relatively low 60$cm/s$ such that visitors would not feel threatened. No harm was done to any visitor, exhibit or robot. The above mentioned failure modes are the only cases where the robots got stuck due to deficiencies of the motion planner. Other blocked situations arose from localization failures making the robot seek goals inside walls or sensor failures at the beginning of Expo.02 (the latter were subsequently detected automatically and made the robot perform an emergency stop).

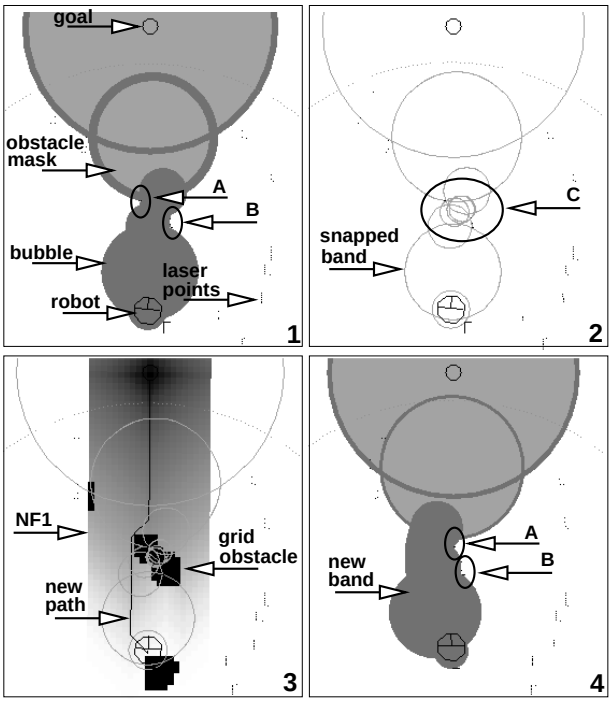


Fig. 8. Typical replan sequence (simulated, takes up to 0.5*s* at Expo). 1: An intact band is squeezed by two visitors A and B approaching each other. 2: The band snaps in C, a replanning thread is started. The robot continues to update and use the snapped band to keep its movements smooth. 3: The unsmooth NF1 path is an intermediate step of the replanning thread. 4: The new path has been translated into a new elastic band.

## VII. FUTURE WORK

The current implementation takes advantage of the specific environment. Ongoing research at the Autonomous Systems Lab aims at making the motion planner more generally applicable.

From time to time, the lack of sensory memory causes an oscillating replanning behavior. This was not a problem at Expo.02, but future applications might be less forgiving in this aspect. Integrating a memory along the lines of the local perceptual space mentioned in [7] should alleviate this problem.

Another issue concerns the simplifications used for the elastic band and how the robot can get stuck when turning into narrow corridors. Here, the non-holonomic nature of the differential drive should be taken into account in the plan representation. A solution which does so only at critical points along the elastic band seems promising, as we aim to preserve the computational efficiency of the simplified elastic band.

## VIII. CONCLUSION

We have presented a novel combination of known algorithms for mobile robot path planning and control. It was shown that our combination performs well enough to be deployed in a challenging long-term real-world application.

Using a time-based clearance measure solves a singularity present in the original DWA. We base the dynamic window in physically meaningful representations and use the speed objective as a means to safely switch between overall robot behaviors.

Relying on the DWA, the elastic band could be simplified to a point where it becomes computationally efficient. The main speedup comes from using euclidean distances. Heuristically masking some obstacles reduces the frequency of replan requests to improve the overall performance of the motion planner in the context of highly cluttered and dynamic environments.

## IX. ACKNOWLEDGMENTS

The authors would like to thank Björn Jensen and Nicola Tomatis for their help developing and refining the approach presented in this paper and the EPFL research council for funding.

## X. REFERENCES

[1] K.O. Arras, J. Persson, N. Tomatis, and R. Siegwart. Real-time obstacle avoidance for polygonal robots with a reduced dynamic window. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2002.

[2] J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–88, June 1991.

[3] O. Brock and O. Khatib. High-speed navigation using the global dynamic window approach. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1999.

[4] H.J.S. Feder and J.-J.E. Slotine. Real-time path planning using harmonic potentials in dynamic environments. In *IEEE International Conference on Robotics and Automation*, April 1997.

[5] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, March 1997.

[6] M. Khatib, H. Jaouni, R. Chatila, and JP. Laumond. Dynamic path modification for car-like nonholonomic mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1997.

[7] K. Konolige. A gradient method for realtime robot control. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.

[8] J.-C. Latombe. *Robot motion planning*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1991.

[9] J. Minguez, L. Montano, T. Simeon, and R. Alami. Global nearness diagramm navigation (gnd). In *IEEE International Conference on Robotics and Automation*, May 2001.

[10] S. Quinlan and O. Khatib. Elastic bands: connecting path planning and control. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1993.

[11] C. Schlegel. Fast local obstacle avoidance under kinematic and dynamic constraints for a mobile robot. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.

[12] S. Sekhavat and M. Chyba. Nonholonomic deformation of a potential field for motion planing. In *IEEE International Conference on Robotics and Automation*, May 1999.

[13] R. Simmons. The curvature-velocity method for local obstacle avoidance. In *International Conference on Robotics and Automation*, April 1996.