2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Receding Horizon Control in Unknown Environments: Experimental Results

Markus Deittert, Arthur Richards and George Mathews

Abstract— **This paper presents a novel approach to navigation in an a priori unknown, GPS-denied environment. The aim is to combine dynamic path planning with the ability to learn about the environment. The vehicle is tasked with autonomous travel from an uncertain initial position to an uncertain target, without prior mapping information. The environment is modelled using linear segments that represent boundaries between the estimated traversable and non-traversable regions. The approach integrates Receding Horizon Control (RHC) and Simultaneous Localisation and Mapping (SLAM). The control problem is formulated as a mixed integer linear program (MILP) and explicitly includes the obstacles and vehicle dynamics. We present the results of our experiments using Pioneer robots, as well as, simulation results which clearly demonstrated the impact of each component of the system.**

I. Introduction

This paper discusses an integration of Receding Horizon Control (RHC) [1] and Simultaneous Localisation and Mapping (SLAM) [2], aiming to solve the problem of autonomous navigation in an unknown, GPS-denied, "semi-urban environment", in which, a vehicle travels in an open 2-D space, but encounters barriers it cannot traverse, e.g. walls, ditches or buildings in a village. However, the question of terrain estimation is not considered. The vehicle has a priori no knowledge of its surroundings, but can sense the boundaries, provided they are within range and not obstructed by any other barrier, e.g. with LIDAR. The target can also be sensed in this way. The target location, as well as the vehicle's initial position and orientation, are known approximately, in some global frame of reference. All obstacles are assumed to be static. The vehicle must discover its surroundings, to the extend that, finding the route to the intended target becomes feasible. The object is to reach the target as quick as possible.

Previous publications have presented the initial implementation of this approach [3] and discussed efficient obstacle avoidance strategies [4]. This paper presents experiments conducted at BAE Systems' demonstrator facility, using Pioneer Robots [5]. In addition, the sensitivity of the system performance to removal of its components is investigated.

A variety of approaches have been researched for problems of this type. Distinguishing features of the approach in this paper are the use of global optimisation for path planning, instead of sampling [6], [7] and a purely feature-based map representation, instead of a grid [8], [9].

Drawing on existing work on path-planning in complex, uncertain environments [10], the planner combines a pre-computed straight-line far-term path approximation with a Mixed-Integer Linear Programming (MILP) near-term detailed planner in a receding horizon framework [11]. Using MILP to solve the near-term detailed planning, allows efficient implementation of nonconvex piecewise linear environments, while taking linearised vehicle dynamics into account. Furthermore, commercial solvers, such as GLPK or CPLEX are available. The approximate linear dynamics model [12], safety constraints [13] and integration of the cost-to-go [11] are all drawn from existing work. Although MILP optimisation has been developed, a number of innovations are required for the new setting considered.

First the wall representation of the world demands boundary representation, not area representation as in previous work [11], [12]. Hence, a modified piecewise affine world representation [14], [15] is used. Second, obstacles are revealed when they are visible to the vehicle's sensors, yielding only linear aspects of complex obstacles. The visible obstacle free space, or isovist [16], around the vehicle is approximated by a union of triangles and is guaranteed to be free of obstacles. The planner is constrained, such that the detailed plan remains inside the isovist, thereby preventing the plan from colliding with either known or yet invisible obstacles. The isovist's triangles are subdivided, such that transitions between neighbouring cells remain inside the isovist [3].

Recursive feasibility [17] is achieved by including a constraint in the optimisation, such that the last way point of the detailed plan is stationary and remains inside the isovist. Thus, if replanning fails or is not desired, the vehicle follows its old plan until the stop is reached. This prevents the combination of near-term and far-term plan from committing the vehicle to a cause of action which makes a collision with an obstacle unavoidable.

Section II reviews the navigation system's components, while experimental results are presented in Section III. Simulation results are discussed in Section IV.

II. Navigation System Components

Fig. 1 gives an overview of the components comprising the navigation system. These are discussed in detail below.

M. Deittert and A. Richards are with the Aerospace Department, Faculty of Engineering, University of Bristol, Bristol, United Kingdom markus.deittert@bristol.ac.uk arthur.richards@bristol.ac.uk

George Mathews is with the Advanced Technology Centre, BAE Systems, Bristol, United Kingdom george.mathews@baesystems.com

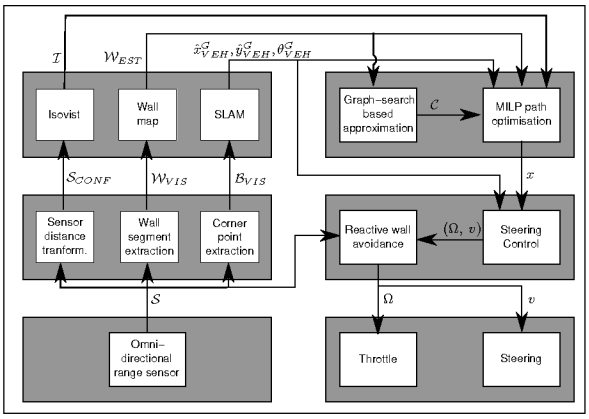

Fig. 1. Navigation system components

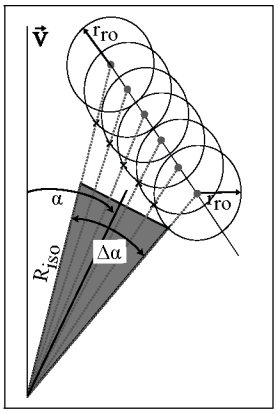

Fig. 2. Isovist triangle components. Measurement (dotted line), isovist cell (shaded area), corrected measurement (x)

A. Vehicle dynamics

The vehicle's dynamics are assumed to be of the Dubin's car type [18]:

$$\begin{aligned} \dot{x}^G_{VEH} &= v\sin\theta + w_x \\ \dot{y}^G_{VEH} &= v\cos\theta + w_y \\ \dot{\theta}^G_{VEH} &= \Omega + w_\Omega \end{aligned}$$

where (x^G_{VEH}, y^G_{VEH}) is the vehicle's position in the global reference frame and θ^G_{VEH} represents the vehicle's heading. The vehicle accepts two control inputs: speed, v, and turn rate, Ω. The maximum turn rate, $\Omega_{\max}$, is limited such that:

$$|\Omega_{\max}| = \frac{v_{\max}}{R_{\min}},$$

i.e. at maximum speed, the vehicle has a minimum turn radius of $R_{\min}$. The state evaluation is subject to noise, w_x, w_y and w_Ω, . The noise is assumed Gaussian with zero mean, standard deviations, σ_x, σ_y and σ_Ω and is uncorrelated.

B. Wall representation

The environment is modelled as a list of linear obstacles, i.e. walls. Define a wall map $\mathcal{W} = (\mathcal{P}_W, \mathcal{L}_W)$ where $\mathcal{P}_W = \{p_1, \ldots, p_{N_{points}}\}$ is a list of points in 2-D and $\mathcal{L}_W = \{(i^a_1, i^b_1), \ldots, (i^a_{N_{walls}}, i^b_{N_{walls}})\}$ is a list of point-pairs. Then there are N_{walls} walls in the map and wall n runs from point $p_{i^a_n}$ to point $p_{i^b_n}$. Define the estimated map as $\mathcal{W}_{EST}$ and the instantaneous map of visible wall segments as $\mathcal{W}_{VIS}$. $\mathcal{W}_{EST}$ is updated with $\mathcal{W}_{VIS}$ after each sensor reading, by a simple combination check. This process is covered in more detail in [3].

C. Feature extraction

Raw sensor data is delivered by the laser range finder in terms of range and bearing pairs, $S_i = (r_i, \beta_i)$, where all ranges are equal to, or less than, the sensors maximum range, $r \leq r_{max}$. Define a sensor scan $\mathcal{S} = \{S_1, \cdots, S_{N_{sense}}\}$ as a list of N_{sense} measurement pairs. Wall segments are extracted from raw range finder data, by an algorithm given in Ref. [19], where straight line segments are fitted to measurement points using a nonlinear regression algorithm [20], resulting in $\mathcal{W}_{VIS}$.

Corners are extracted from the identified wall segments, $\mathcal{W}_{VIS}$, using visibility information to differentiate between true corners and overlapping wall sections. Corners are given in terms of their polar coordinates, $B_i = (r_{C_i}, \beta_{C_i})$. These are obtained by computing the intersection point of known wall segments. The corners are used as beacons by the SLAM algorithm. Define $\mathcal{B}_{VIS} = \{B_1, \cdots, B_{N_{beacon}}\}$ as a list of N_{beacon} currently visible beacons.

D. Isovist

During the near-term planning horizon, the vehicle is constrained to remain inside the isovist. This is converted into an equivalent movement of the vehicle's centre of gravity, in the spirit of the configuration space [21], by placing a virtual circle, of radius r_{ro}, at the location of each reading, S_i, in the sensor scan, $\mathcal{S}$, and re-sampling the distances at the same bearings, β_i, as the original sensor scan (Fig. 2). This is achieved by finding the closest point of intercept between the circles and the ray in direction β_i, giving a new sensor scan, $\mathcal{S}_{CONF}$.

The isovist is based on the union of isosceles triangles of equal angular width, centred at the vehicle. Each isovist triangle has a length of its equal sides, R_{iso}, bearing, α and opening angle, $\Delta\alpha$ (Fig. 2). A number of sensor readings, S_i, must be collected into an isovist triangle, as otherwise, the isovist would contain too many triangles, resulting in a large optimisation problem, which cannot be solved in real time. The length, R_{iso}, thus equals the smallest sensor reading in the triangle's bearing range, $\Delta\alpha_i$.

The isovist's triangles are subdivided further into triangular cells, such that neighbouring cells share a common edge (Fig. 3). Let $\mathcal{I} = (\mathcal{L}_I, \mathcal{P}_I)$ be a map of triangle points, $\mathcal{P}_I = \{p_1, \cdots p_{N_{points}}\}$ and a list of point triplets, $\mathcal{L}_I = \{(l^a_1, l^b_1, l^c_1), \cdots, (l^a_{N_{trig}}, l^b_{N_{trig}}, l^c_{N_{trig}})\}$, which defines the isovist cells by their corner points. This list of isovist cells is used in (6).

Let $T_i = (C_a, C_b)$ be a pair of isovist cell indexes from $\mathcal{P}_I$, with a common edge between cell a and cell b. Then $\mathcal{T} = \{T_1, \cdots, T_{N_{trans}}\}$ is a list of N_{trans} permitted transitions between isovist cells, which is used in (7) (Fig. 3).

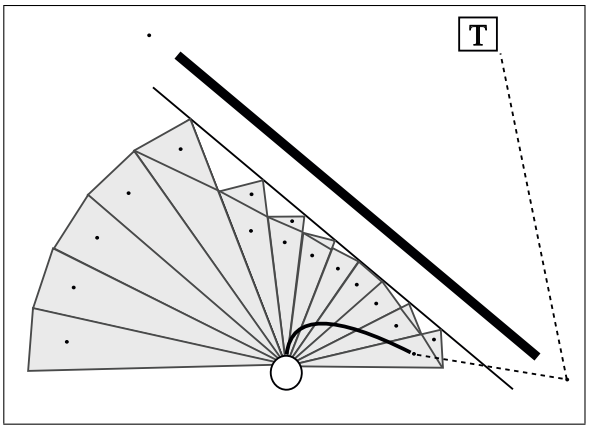

Fig. 3. Planner components: robot (circle), isovist (shaded area), estimated walls (thick line), near-term plan (thin line), far-term plan (dashed line), navigation nodes (black dots) and target (T)

E. Graph-based far-term approximation

The far-term path approximation proceeds by constructing a set of nodes at locations the vehicle is likely to pass and then constructing the visibility graph [21] between these nodes. The visibility graph is formed in a naïve manner: first constructing an arc for all pairs of nodes and then removing those arcs that intersect walls in $\mathcal{W}_{EST}$. Although this simple approach could be improved upon, it has so far not been shown to limit the achievable performance.

Previous work used obstacle corners for node locations [11] and these have been found to be good choices. Thus, nodes are placed at the ends of wall sections and corners, where two nodes are placed, one in the convexity and one in the concavity [3]. Fig. 3 shows a straight line example. Additional nodes are placed at the centroid of the isovist's cells.

Visibility of the first far-term node from the end of the near-term plan is enforced in the optimisation. Initial experiments showed that checking the visibility of every cell against every node is computationally expensive. Hence, in the current implementation only nodes in isovist cells are marked as visible from *within* their respective cells. The node visibility constraint is thereby transformed into a location constraint. Let $V_i = (C_j,\, N_k)$ be a cell and node pair, with a clear line of visibility between any location inside cell *j* to node point *N*. Let $\mathcal{V} = \{V_1, \cdots, V_{N_{pairs}}\}$ be a list of N_{pairs} cell to node visibility pairs.

Once the visibility graph is constructed, value iteration [22] is then used to determine the shortest distance along the graph to the target from each node. The pairs of nodes $\mathcal{N} = \{n_1, \cdots, n_{N_{nodes}}\}$, where each n_i is the location in 2-D, and the associated cost-to-go values $\mathcal{C} = \{c_1, \cdots, c_{N_{nodes}}\}$ are subsequently used in the MILP detailed path planner.

F. Simultaneous Localization and Mapping

A "standard" Simultaneous Localisation and Mapping (SLAM) solution [23] has been implemented, in this case using the beacons, $\mathcal{B}_{VIS}$, from Section II-C and vehicle dynamic and process noise from Section II-A. Estimation is performed using an Extended Kalman Filter (EKF) and association of features is done using Munkres' assignment algorithm [24] with the Mahalanobis distances as costs [2]. Having found the assignment of features to measurements achieving the lowest summed cost, those assignments within a specified distance threshold are accepted. Measurements not assigned to an existing feature are candidates for new beacons, $\mathcal{B}_{CAN}$. A periodic "tidy-up" is performed to seek and recombine duplicate features.

The outputs of the SLAM algorithm are estimates of the vehicle configurations $\hat{x}^G_{VEH}, \hat{y}^G_{VEH}$ and $\hat{\theta}^G_{VEH}$, corresponding to the model in Section II-A. The SLAM filter also maintains estimates of the feature locations, $\mathcal{B}_{EST}$, but currently, these are not used any further.

G. Constraint tightening

The vehicle's position is uncertain due to the tracking error of the low level controller, modelling errors of the linearised dynamics and uncertainty in the sampling period length. To reduce the likelihood of collision with obstacles, constraint tightening [25] is used. In this implementation, the isovist, $\mathcal{I}$, is scaled down for consecutive time steps, so that, the planner must place the plan points further away from the obstacles at later time steps. Thus $\mathcal{P}_I(k=0) = \mathcal{P}_I$, $\mathcal{P}_I(k=1) = (1-\alpha)\,\mathcal{P}_I$, and $\mathcal{P}_I(k \geq 2) = \left(1 - \frac{3}{2}\alpha\right)\,\mathcal{P}_I$. Hence, the permissible area shrinks but retains its shape and cells.

H. Receding Horizon Control

The MILP optimization employs a linear system approximation of the dynamics presented in Section II-A. The estimated configuration from SLAM is converted into the linear state:

$$x_0 = \begin{pmatrix} \hat{x}^G_{VEH} \\ \hat{y}^G_{VEH} \\ v_{LAST} \sin \hat{\theta}^G_{VEH} \\ v_{LAST} \cos \hat{\theta}^G_{VEH} \end{pmatrix}$$

where v_{LAST} is the speed before the planning. Then the MILP optimization is as follows:

$$\min_{N_{\Delta t}, u(\cdot), x(\cdot), j(\cdot), i_{VIS}} \left\{ \begin{array}{l} W_N\; N_{\Delta t} \\ +W_F\; \|[I\;0]x(N) - p_{i_{VIS}}\| \\ +W_F\; c_{i_{VIS}} \end{array} \right\}$$

subject to

$$\begin{aligned} N_{\Delta t} &\leq N_{\max} && (1)\\ x(0) &= x_0 && (2)\\ x(k+1) &= Ax(k) + Bu(k) && (3)\\ ([0\;I]\,x(k), u(k)) &\in \mathcal{L} && (4)\\ [0\;I]\,x(N_{\Delta t}) &= 0 && (5)\\ [I\;0]\,x(k) &\in \mathcal{I}\,(j(k),\, k) && (6)\\ (j(k+1), j(k)) &\in \mathcal{T} && (7)\\ (j(N_{\Delta t}), i_{VIS}) &\in \mathcal{V} && (8) \end{aligned}$$

Fig. 4. Photo of one of the experiments using arena layout (f) of Fig. 5. The Pioneer travels from the centre top corner towards the cardboard box.

where $[I\ 0]\,x(k)$ denote a sequence of near-term way point, $[0\ I]\,x(k)$ are the associated velocities and $u(k)$ denotes a sequence of acceleration controls for the linear system. The variables $j(k)$ and i_{VIS} represent choices of cell for each time step and of visible cost-to-go node, respectively, while $p_{i_{VIS}}$ represents the coordinates of the first far-term node point. $N_{\Delta t}$ are the number of time steps in the near term plan, while W_N and W_F are weightings for the near- and far-term plan respectively. In order to prevent near-term plans of zero activity, W_N and W_F are set such that, the far-term plan is more "expensive".

Constraint (1) limits the length of the near-term plan and (2) fixes the starting point at the current state. (3) and (4) represent linearised model and constraints from [12]. Recursive feasibility is ensured by (5). (6) requires the plan to remain inside the isovist cells picked from the set $\mathcal{I}\,(j(k),k)$, while (7) limits transitions between cells to the set of permitted transitions, $\mathcal{T}$. Constraint (8) enforces visibility between the last way point of the near-term plan and the first node of the far-term plan.

I. Low level control

A simple low level feedback controller is used to convert the plan, $x(k)$, and acceleration inputs, $u(k)$, into turn rate, Ω and speed, v, commands for the vehicle. The low level controller aims for the first way point of the plan ($k = 1$) using a simple proportional feedback steering controller together with open-loop speed control, thus:

$$\begin{aligned} \Omega &= -K_P\,(0-\beta_{WP}) \\ v &= \|\,[0\ I]\,x(1)\| \end{aligned}$$

where β_{WP} is the vehicle relative bearing to the way point and K_P is a proportional feedback constant.

Although constraint tightening is intended to bestow the system with some robustness, we cannot guarantee collision avoidance. Hence, additional low level avoidance behaviour is included. This initiates a full stop followed by a stationary turning motion if an obstacle is sensed at a distance, r, less than the avoidance threshold, r_{trig}, i.e. if $\min\,(r(\delta)|-\pi\leq\delta\leq\pi)\leq r_{trig}$.

III. Practical Experiments

27 experiments where conducted using 6 different arena layouts (see Fig. 5) with 3-5 runs per layout. The layouts are of increasing difficulty, with additional walls being placed between the robots initial position and the target. BAE Systems' demonstrator facility allows arenas measuring up to 7.1m by 5.4m, with walls created from movable barriers which are 1.8m long and 0.8m high. Fig. 4 shows a typical arena layout.

A. Robot configuration

The Pioneer robot was equipped with a SICK laser range finder. This sensor gives the robot a field of view of 180°, angular resolution of 0.5° and a maximum range of approximately 8m, taken at a rate of 2 Hz. The Pioneer robot has been artificially limited to forward speeds up to 0.2 m/s and maximum turn rates of 0.2 rad/s, reversing being not allowed due to the limited field of view. At full speed, the robot's turn radius was 1 metre.

The navigation system has been implemented using Matlab, while the MILP optimisation has been implemented using AMPL and solved with CPLEX. The system has two feedback loops of different speed. The fast loop, which provides the low level control feedback, operates at a nominal rate of 2 Hz, while the slow loop, which is closed by replanning the path, nominally operates at 1 Hz. Both loops use the position estimate generated by the sLAM and all computation, including the MILP optimisation, has to be completed in between sensor scans. Because Matlab is not a hard real-time environment, the sample interval of 500 ms could not be guaranteed and occasional overruns of up to 100 ms were observed.

B. Results

Experiments during which the robot arrives within 0.8 metres of the target, with a clear line of sight, where counted as successful. The experiments were stopped if the robots was deemed to be at risk of collision by the supervisor walking behind (Fig 4), if an emergency stop was initiated by the robot due to a planner malfunction or after time-out occurred.

Of the 27 experiments 22 did reach the target and 5 failed. The full list can be found in Tab. I. The failures, which are indicated by circles, are restricted to arena layouts (d) and (e).

Two failure modes were observed. First, the planner fails if the vehicle moves closer to a wall than r_{ro}. In this case, there is zero space in the isovist, the optimisation problem becomes infeasible and the planner cannot recover from this failure. It is likely that the failure is caused by inadequate constraint tightening. For simplicity, the constraint tightening shrinks the isovist's radius by a factor, in our current implementation. However theory [26] demands that the isovist must be shrunk by an absolute margin, corresponding to the position uncertainty. This type of wall proximity failure was observed during experiments 20 and 22, with the robot getting too close to wall segments 1 or 2 (Fig 5(e)).

The second failure mode is a time out due to errors in the vehicle's internal world model, arising from inconsistencies

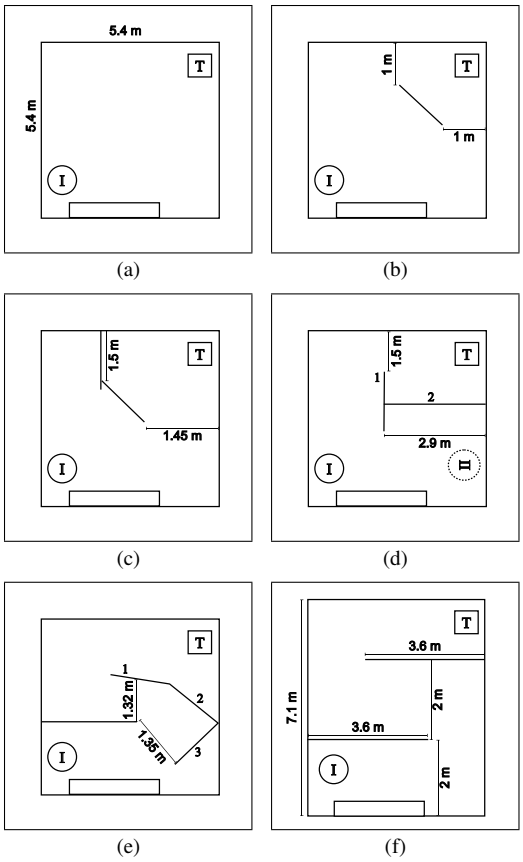

Fig. 5. Arena layouts of the experiments. T indicates the target location while I indicates the initial position. The dotted circle in Fig. 5(d) denotes an alternative starting location.

TABLE I
EXPERIMENTS AND RESULTS

Arena*	Time steps to goal
(a)	36, 35, 36
(b)	63, 49, 51, 53
(c)	61, 65, 57, 54
(d)I	(83), (180), 66, (77)
(d)II	98, 98, 66
(e)	161, (92), 123, (101), 118
(f)	143, 129, 138, 121

circled numbers indicate experiments where the target was not reached
* see Fig. 5

in the SLAM's state estimate [27]. In this case, the target may become obscured by false walls or false gaps may open in the wall model, where solid walls should be. This type of failure was mostly observed during experiments 12, 13 and 15 (Fig 5(d)), in which a sharp turn in confined space let to loss of SLAM consistency.

IV. SIMULATIONS

The results of the practical experiments have been used to tune sensor and process noise levels of a MATLAB simulation, such that the observed behaviour is replicated. The simulator has then been used to simulated the systems behaviour with individual features missing. For each of the scenarios, simulations have been repeated 50 times, using the arena layout of Fig. 5(f).

Fig. 6 shows the results of these simulations. The base line scenario shown in this figure, is a simulation of the full system with all features being active. For comparison, the following features have been deactivated in the other simulations:

A. Dynamic constraints

Removing the dynamic constraints (3), (4) from the optimisation allows the planner to place plan points within the isovist without having to respect acceleration limits. Cell transition constraints are still enforced. It can be seen from Fig. 6 that many runs still reach the goal, although in longer times than the baseline and with more interaction with the walls. Furthermore, more runs end in timeout without reaching the goal. These results illustrate that while the low level avoidance behaviour and the visibility graph offer some capability for autonomous navigation, the performance is degraded through the absence of dynamics constraints. In particular, since the vehicle turning restrictions are ignored, the plans are often hard to follow, leading to close contact with walls.

B. Visibility graph

In this scenario the visibility graph has been replaced by a cost-to-go based only on straight line distance between the node points and the target, ignoring any intersecting walls. This eliminates considerable computation, but as seen in Fig. 6, leads to significant degradation compared to the baseline. Observations suggest that the robot now adopts a wall-following approach attracted by local minima, leading to many more interactions and slower progress, as seen in the figure.

C. Constraint tightening

It can be seen from Fig. 6 that the removal of the constraint tightening has no significant effect on the performance. This supports the indication from the practical experiments, Section III-B, that the constraint tightening did not increase the system robustness as expected. Hence the the improvements suggested in Section III-B, including a change to use absolute margins, will be considered in future implementations.

V. CONCLUSIONS

The navigation approach has been demonstrated to work in real-time on a Pioneer robot, showing repeatable success in a variety of environments. The success rate of the experiments is dependent on the environment's layout, with tight corridors and corners causing problems. The system works satisfactorily as long as the robot has enough room to manoeuvre.

The simulations show the implementation of the navigation system to be efficient, in the sense that, removing any one of its major features impairs performance. In particular, omitting the linear dynamic model increases the time

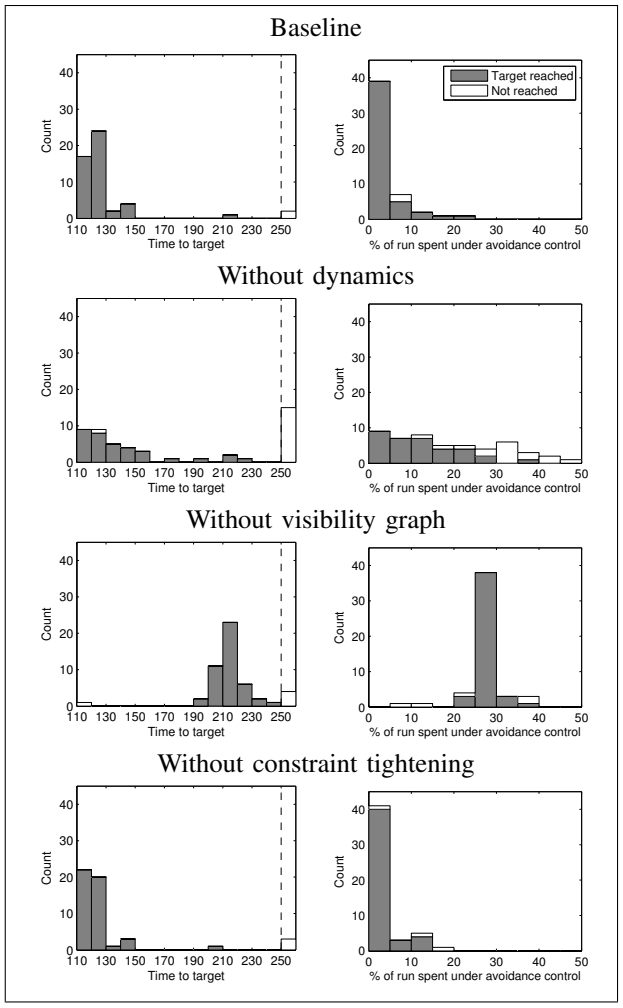

Fig. 6. Simulation results, dashed line: time-out limit. The time passed until the target is reached is given in sampling intervals.

required to reach the target and also increases the time spent under low level avoidance control. The simulations have also shown the current implementation of the constraint tightening to be ineffective and an implementation based on a reduction of the isovist area by an absolute distance instead of relative scaling is required in a future version.

VI. ACKNOWLEDGMENTS

The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence, grant nos. IF049 and AA016.

The authors would like to thank Keffin Barnaby and James Parkin from BAE Systems, for their help with the experiments.

REFERENCES

[1] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2002.

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (slam): Part i the essential algorithms," *IEEE Robot. Automat. Mag.*, vol. 13, pp. 99–110, Jun. 2006.

[3] A. Richards, "Adaptive navigation in an uncertain urban environment," in *SEAS DTC 3^{rd} Annual Tech. Conf.*, Edinburgh, UK, Jun. 2008.

[4] M. Deittert and A. Richards, "Effcient planning for adaptive navigation," in *SEAS DTC 4^{th} Annual Tech. Conf.*, Edinburg, UK, Jul. 2009.

[5] "Mobile robots Inc." 2008. [Online]. Available: http://www.activrobots.com/robots.html

[6] T. Howard, C. Green, and A. Kelly, "State space sampling of feasible motions for high performance mobile robot navigation in highly constrained environments," in *Int Conf. on Field and Service Rob.*, Jul. 2007.

[7] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. Contr. Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, Sep. 2009.

[8] S. Kolski, D. Ferguson, M. Bellino, and R. Siegwart, "Autonomous driving in structured and unstructured environments," in *IEEE Int. Veh. Symp.*, Tokyo, Japan, Jun. 2006.

[9] C. Harris, "Strategies for visual exploration of buildings," in *SEAS DTC Tech. Conf.*, Edingburgh, UK, 2007.

[10] Y. Kuwata, A. G. Richards, T. Schouwenaars, and J. P. How, "Distributed robust receding horizon control for multi-vehicle guidance," *IEEE Trans. Contr. Syst. Technol.*, vol. 15, no. 4, pp. 627–641, Jul. 2007.

[11] J. S. Bellingham, A. G. Richards, and J. P. How, "Receding horizon control of autonomous vehicles," in *Proc. American Control Conf.*, 2002.

[12] A. G. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proc American Control Conf.*, 2002.

[13] T. Schouwenaars, J. P. How, and E. Feron, "Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees," in *Proc. AIAA Guidance, Navigation & Control Conf.*, no. AIAA-2004-5141. Providence, Rhode Island: AIAA, 2004.

[14] G. Keith, J. Tait, and A. Richards, "Efficient path optimization with terrain avoidance," in *AIAA Guidance, Navigation and Control Conf.*, 2007.

[15] M. P. Vitus, V. Pradeep, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Tunnel-milp: Path planning with sequential convex polytopes," in *AIAA Guidance, Navigation, and Control Conf.*, 2008.

[16] M. L. Benedikt, "To take hold of space: isovists and isovist fields," *Environment and Planning B*, vol. 6, no. 1, pp. 47–65, 1979.

[17] T. Schouwenaars, J. P. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *Proc. American Control Conf.*, 2004.

[18] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. Vol 79 No 3, pp. 497–516, 1957 1957.

[19] R. Taylor, *Active Sensors for local Path Planning*. World Scientific Publishing Company, Ltd., 2001, ch. Extracting Lineas and Curves from Optoelectronics Range Data, pp. 223–238.

[20] A. Diosi and L. Kleeman, "Uncertainty of line segments extracted from static SICK PLS. laser scans," in *Australian Conf. on Robotics*, 2003.

[21] J.-C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[22] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2001.

[23] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, pp. 229–241, Jun. 2001.

[24] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc.of Ind. and Appl. Math.*, vol. 5, no. 1, pp. 32–38, Mar. 1957.

[25] L. Chisci, J. A. Rossiter, and G. Zappa, "Systems with persistent disturbances: Predictive control with restrictive constraints," *Automatica*, vol. 37(7), pp. 1019–1028, 2001.

[26] A. Richards, "Robust model predictive control for time-varying systems," in *Europ. Control Conf.*, 2005.

[27] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended kalman filter based slam," *IEEE Trans. Robotics*, vol. 23, no. 5, pp. 1036–1048, 2007.