This is an accepted version of the article: Kulich, M., Kozák, V., and Přeučil, L. (2015). **Comparison of Local Planning Algorithms for Mobile Robots**. *Modelling and Simulation for Autonomous Systems (Second International Workshop, MESAS 2015)*, which has been published in a final form at https://doi.org/10.1007/978-3-319-22383-4_15.

Comparison of local planning algorithms for mobile robots

Miroslav Kulich, Viktor Kozák, and Libor Přeučil

Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Technicka 2, 166 27 Prague 6, Czech republic
kulich@labe.felk.cvut.cz
viktor-kozak@seznam.cz
preucil@labe.felk.cvut.cz
http://imr.felk.cvut.cz

Abstract. Local planning algorithms are an essential part of today's mobile robots and autonomous vehicle control. While the global planning decides the route of the robot based on initial data given to the planner, the local planning is a real-time motion control, based on the feedback from sensors. Its purpose is to keep the robot on an optimal track, following the global planner and to avoid unexpected obstacles, making it a fundamental part for safe robot navigation. The paper aims to compare three local planning algorithms: the Dynamic Window Approach, Enhanced Vector Field Histogram and Smooth Nearness-Diagram. The comparison was made on various maps in the Player/Stage system and with a real robot in SyRoTek (System for Robotic e-learning). More than 20 000 simulation runs and 120 hours of experiments with a real robot on SyRoTek were made at first to find best configurations for the particular planning algorithms and the robot used. After that, another set of experiments with the found parameters was conducted to gain the results needed for comparison of the algorithms.

Keywords: local planning, mobile robots, obstacle avoidance

1 Introduction

When designing an autonomous robotic system, key attention has to be paid to planning functionalities. These functionalities heavily influence behavior of the system and enable it to operate effectively. Planning is typically employed in several layers of a deliberative robotic control architecture [5]. *Mission planning* at the highest level interacts with a human operator and provides consecutive goals to be fulfilled in order to accomplish the whole mission. The intermediate *path planner* obstacle-free path based on an actual model of the working environment (either provided by the operator or built autonomously during the mission). This path is than converted into a sequence of control commands to be performed by robot's actuators. This works if the environment is static, its model is precise enough, and movement of the robot is noise-free. In the other case, *local planner/obstacle avoidance module* aims to detect obstacles around the robot with some sensors and modify the path to avoid collisions with obstacles.

The paper aims to compare performance of three reactive local planning algorithms that are probably mostly used nowadays : the Dynamic Window Approach (DWA) [3], Vector Field Histogram Plus (VFH+) [8] and Smooth Nearness-Diagram (SND) [2] algorithms. The comparison was made on various maps in simulation on the Player/Stage system [4] and with a real robot in the SyRoTek system (System for Robotic e-learning) [6].

The rest of the paper is organized as follows.

2 The compared methods

2.1 Vector Field Histogram Plus

Vector Field histogram Plus (some authors call it Enhanced Vector Field Histogram) [8] is an improvement of Borenstein's and Koren's Vector Field Histogram (VFH) [1]. VFH employes a polar histogram grid for representation of robot's surrounding environment. This grid is built from a two-dimensional certainty grid updated from sensor readings taken with a ranging sensor (sonar, laser range-finder). The polar histogram is a vector that moves with the robot. Each element of the histogram corresponds to a circular sector for which information about amount and distance of obstacles in the form of a weighted sum is stored. The histogram after smoothening (by a simple low-pass filtering) has typically "peeks", i.e. sectors with high values and "valleys", sectors with low values. Any valley containing sectors with values below a certain threshold can be called a candidate valley. If more than one candidate valley is detected, the best one that most closely matches the direction to the target is selected. Finally, the most suitable sector within the selected valley is chosen as the next goal and the robot is navigated towards it. The whole process is repeated whenever new sensor data are gathered (or in predefined time steps) until the final goal is reached.

VFH+ enhances the original algorithm in several ways. First, threshold hysteresis is used to suppress alternating between several goals in narrow openings resulting in robot movement in the close vicinity of obstacles. Moreover, robot's size is taken into account by enlarging obstacle cells by a robot diameter. Finally, dynamics and kinematics of the robot were not taken into consideration. VFH+ uses a simple approximation of currently possible robot's trajectories by a set of circular arcs with various curvatures assuming that forward and angular velocities are piecewise constant.

2.2 Smooth Nearness Diagram

Another approach was presented by Minguez and Montano [7] as a geometry-based implementation of the reactive navigation method design. The approach called Nearness-Diagram (ND) navigation introduces “gaps” – discontinuities in the nearness of obstacles to the robot which indicate potential paths into occluded areas of the environment. By the pairs of consecutive gaps “regions” can defined, navigable regions are then “ valleys”. After assembling all the valleys surrounding the robot, all the gaps are compared against the heading provided by the global planner. The next subgoal and control to it is determined based on position of two closes obstacles and the width of the valley containing the gap with the heading that best matches the heading to the goal.

Smooth Nearness Diagram (SND) [2] differs from ND in the way the next subgoal is determined. SND measures a threat possessed by each of the obstacles (an obstacle is considered a threat if it lies within the safety distance of the robot) – the treat measure increases as the obstacle gets closer to the robot. Deflection from the desired heading is computed based on threat measurements of each obstacle. The experiment show that oscillatory patterns were suppressed leading to method’s performance improvement in narrow corridors.

2.3 Dynamic Window Approach

The Dynamic Window Approach (DWA), introduced by Fox et al. [3], searches control commands directly in the space of velocities and takes limitations of the velocities and accelerations of the robot into account. The approach consists of two parts: search space is generated at the first, followed by determination of the optimal path int the generated search space.

A two dimensional search space of valid linear and angular velocities is computed directly from the limitations of the velocities and accelerations of the robot. The origin of the space lies in the point representing the current robot’s linear and angular velocities. The space is then discretized based on computation power of the robot and the requested precision, i.e. a number of possible pairs of translational and rotational velocities is chosen from the interval between the maximal and minimal velocities. Each velocity pair(v, w) is represented by a circular arc with the starting point in the center of the robot. It’s radius is calculated as $r = \frac{v}{w}$ and the length of the arc is set as v. This representation is called the dynamic window. All curvatures outside this dynamic window cannot be reached by the robot in the next step, and thus are not considered for obstacle avoidance. Each trajectory is then compared with readings from the laser range finder. The trajectory is considered safe if the robot is able to stop before colliding with any object along the path.

Selection of the optimal trajectory is based on three basic attributes, which differ for each velocity pair: angular, velocity, and clearance. The angular attribute expresses favors directions trajectory similar to the global goal position, velocity prefers high linear velocities, while clearance favors trajectories going away from obstacles. The pair from the search space with the highest weighted sum of values of the attributes is chosen as the next control command. Behavior of the algorithm can be easily continuously modified (from safe to fast) by adjusting the weights of the particular attributes.

3 Experimental evaluation

Performance of the approaches described in the previous sections has been evaluated in several types of environments in simulations using the Player/Stage framework [4], see section 3.1 and with a real robot in the SyRoTek system [6], which is described in section 3.2. VFH+ and SND drivers from Player/Stage and our own implementation of DWA (due to its unavailability in Player/Stage) were used for comparison.

3.1 Simulation

The experiments in Player/Stage were performed in five environments differing in width of passages (see Fig. 2), so performance of the algorithms in narrow corridors, wide passages, open space and combination of these can be evaluated. Size of all maps is 3.5×3.8 m and a model of the SyRoTek S1R robot equipped with Hokuyo laser range finder was used. The robots model was restricted to maximum speed 0.3 ms/s, maximum turn-rate 1 rad/s, maximum acceleration 0.05 m/s and maximum turn-rate acceleration 0.05 rad/s. Dimensions of the robot and thus the model are (length $\times$ width $\times$ height): $174 \times 162 \times 180$ mm.

To ensure a fair comparison 3.1 best settings of the particular methods was determined first, results in the simulator are described in 3.1, while section 3.2 is devoted to experiments in the SyRoTek system.

Parameters settings Best settings of the VFH+ and DWA algorithms this was done in several steps. The algorithms were tested manually at first trying to find configurations with satisfactory results. After that, a broad range of parameter configurations was made in the neighborhood of the parameter values found by hand. These configurations are shown in Table 1.

The SND algorithm has been heavily used at CTU for several years and therefore configuration that is currently used and has been determined as most reliable was chosen for further evaluation.

Table 1: Tested parameters. 320 configurations were tested for DWA and 108 configurations for VFH+.

DWA

Parameter	Min	Max	Step
robot_radius	0.10	0.13	0.01
ValAngle	5	20	5
ValVelocity	5	20	5
ValObstacle	0	20	5

VFH+

Parameter	Min	Max	Step
cell_size	0.01	0.02	0.01
window_diameter	10	20	10
safety_dist_0ms	0.05	0.09	0.02
weight_desired_dir	5	9	2
free_space_cutoff_0ms	$2 \cdot 10^6$	$4 \cdot 10^6$	10^6

The configurations for the DWA and VFH+ algorithms in Table 1 were tested on the *Arena 0* map (Fig. 2a). 10 test runs were made for each configuration and best 60 configurations for DWA and 9 for VFH+, which achieved 100% success rate, were taken for the final round of experiments described in Section 3.1. Note that search for ideal parameter configurations for the DWA algorithm is made on a larger scale than for the VFH+ algorithm. That is because the DWA algorithm was developed as a part of the described work and it is desirable to promptly test it's full functionality.

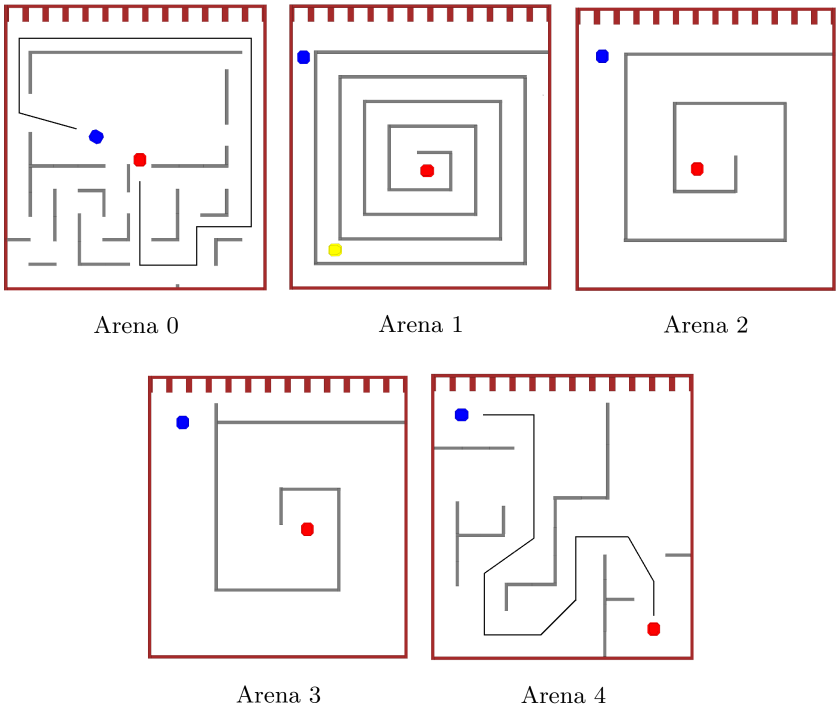


Fig. 2: The testing environments. The narrowest passage of *Arena 0* and *Arena 1* is 28 cm, *Arena 2* and *Arena 4* have the narrowest passage 60 cm, whole *Arena 3* 85.5 cm. The starting position is represented by the blue icon of the robot, and the goal position is represented by the red dot. Some of the maps contain a secondary starting point represented by a yellow icon or a black line representing the desired trajectory of the robot.

Results In the final evaluation, 50 test runs were made for each configuration of every algorithm (DWA, VFH+, SND) on every map (Fig. 2).

Tables 3–8 display resulting statistic from the experiments. The values presented in the tables are: t_{exp}, t_{min}, t_{max}, v_{exp} and the success rate. The t_{exp} is the average time in which the robot reached the goal, t_{min} is the minimal time and t_{max} is the maximal time over all runs. The v_{exp} is the average speed of the robot and the success rate is a percentage value of successful runs.

The tables present the results for the best configuration of each algorithm on the particular map (*DWA* and *VFH*) and the results for the best configuration of the DWA and VFH+ algorithms overall (DWA_{best} and $VFH{+}_{best}$). The best results are chosen at first by the success rate and then by the average time.

As *Arena 1* simulates a passage through a very narrow corridor (the width of the corridor is less than twice the diameter of the robot). This map has considerably lower success rate than other maps. Thanks to that it was possible to push the navigation algorithms to their limits and see the difference in their reliability. In order to achieve a higher success rate and thus larger quantity of times for the computation of more reliable data, additional experiments were made on this map, using a different starting point as shown in Fig. 2. Computed data are thus split into two sets of results.

The SND algorithm proved to be the most reliable amongst the three algorithms tested. The results form the *Arena 1* show a great difference between the success rate of the SND algorithm and the success rate of other two algorithms. This algorithm also proved to have similar time results as the other algorithms on most of the maps. The DWA algorithm proved to be less reliable, when used in long narrow passages in *Arena 1*, aside from this map, its performance was mostly equal or superior to the performance of the SND algorithm. The VFH+ algorithm proved to be less reliable than the other two and it's time results were worse than the results of the other two algorithms.

Particular situation occurred with the VFH+ driver whenever the robot got stuck close to an obstacle and couldn't go as planed, it seemed to start its "escape function". The escape function appears to have a simple system, the robot keeps turning counter-clockwise, looking for an open space in which it could continue it's motion. This function deals with the problem of a robot getting stuck near an obstacle, but has one weakness, the robot always starts turning counter-clockwise regardless of the position of the target. That sometime causes the robot to lose a lot of time, just by turning in the opposite direction than the direction of the target. This function makes it unfavorable for the algorithm to be used in environments containing any narrow turns in the clockwise direction. If this function would be fixed to select the turning direction accordingly to the position of the target, it could significantly improve the performance of the VFH+ controller.

3.2 Experiments with a real robot

Evaluation with a real robot was performed a similar way as in simulation. The SyRoTek S1R robot equipped with Hokuyo laser range finder operated in the SyRoTek Arena of size 3.5×3.8 m. Robot's maximal speed was restricted to $0.45\ \mathrm{ms}^{-1}$ and its maximal turn-rate to $1\ \mathrm{rad\,s}^{-1}$.

A huge set of configurations for all three algorithms was tested first. Based on this, several most promising configurations were chosen for which a detailed evaluation was make.

Table 3: Results for Arena 0.

Algorithm	t_{exp}	t_{min}	t_{max}	v_{exp}	success rate
DWA	80.842	77.364	91.326	0.1263	100%
SND	80.621	78.812	81.654	0.1247	100%
VFH+	153.701	145.762	164.423	0.0775	32%
DWA_{best}	82.693	77.629	108.207	0.1247	100%
$\text{VFH+}_{\text{best}}$	168.896	155.660	182.886	0.0709	22%

Table 4: Results for Arena 1 when using the first starting point.

Algorithm	t_{exp}	t_{min}	t_{max}	v_{exp}	success rate
DWA	268.092	239.299	305.973	0.1153	18%
SND	211.084	208.311	213.881	0.1429	66%
VFH+	305.431	289.636	322.115	0.1042	20%
DWA_{best}	253.711	233.581	304.169	0.1228	10%
$\text{VFH+}_{\text{best}}$	303.762	286.719	324.883	0.1057	14%

Table 5: Results for Arena 1 when using the second starting point.

Algorithm	t_{exp}	t_{min}	t_{max}	v_{exp}	success rate
DWA	146.402	143.614	150.819	0.116	56%
SND	131.035	127.802	133.538	0.127	92%
VFH+	191.199	177.542	212.584	0.095	60%
DWA_{best}	146.402	143.614	150.819	0.116	56%
$\text{VFH+}_{\text{best}}$	189.953	175.381	209.555	0.095	58%

Table 6: Results for Arena 2

Algorithm	t_{exp}	t_{min}	t_{max}	v_{exp}	success rate
DWA	84.189	80.400	88.307	0.1715	100%
SND	85.318	84.542	85.982	0.1656	100%
VFH+	115.864	108.242	121.508	0.1425	100%
DWA_{best}	86.129	83.206	89.810	0.1691	100%
$\text{VFH+}_{\text{best}}$	115.864	108.242	121.508	0.1425	100%

Parameter settings While several simulations could be performed at the same time, the experiments on the real robot could only be made "one at the time" that made the experiments quite time consuming. Therefore the initial experiments to determine the best configurations were made on a smaller scale than during the simulations. After that 20 test runs were made for three final configurations and for the best configuration from the simulations.

Table 7: Results for Arena 3

Algorithm	t_{exp}	t_{min}	t_{max}	v_{exp}	success rate
DWA	56.485	54.863	59.866	0.1761	100%
SND	57.641	56.442	58.428	0.1688	100%
VFH+	56.976	50.263	67.827	0.2201	100%
DWA_{best}	56.949	55.276	60.168	0.1756	100%
$\text{VFH+}_{\text{best}}$	56.976	50.263	67.827	0.2202	100%

Table 8: Results for Arena 4

Algorithm	t_{exp}	t_{min}	t_{max}	v_{exp}	success rate
DWA	48.548	46.269	51.362	0.1628	100%
SND	65.310	64.239	66.157	0.1106	100%
VFH+	77.051	65.769	120.463	0.1352	80%
DWA_{best}	51.523	47.455	78.900	0.1572	100%
$\text{VFH+}_{\text{best}}$	73.684	66.366	112.436	0.1382	66%

Initial experiments were made on two maps, see Fig. 4: (1) a small set of experiments made on the *Map 1*, with a passage through an environment containing a narrow corridor and (2) a larger set of experiments on *Map 2* containing a passage through an environment with a lot of narrow corridors. Two test runs were made on for *Map 1* for each configuration from Table 9 for every algorithm and the best three configurations were chosen accordingly to the success rate and the average speed. Two test runs on *Map 2* were made for each configuration from Table 10 for every algorithm and the best 10 configurations were chosen for the second round of experiments by the success rate and the average speed. In the second round 10 runs were made for each configuration and the three best configurations with highest success rate and average speed were chosen.

Results The three most suitable configurations of each algorithm from the initial testing were chosen for further experiments together with the best configuration gained from the simulation data. 20 runs were made then for each of these configurations and for both maps.

Tables 11-13 present the results for the best configurations of each algorithm on the particular map and the results for the configuration gained from simulation data. Moreover, five-point summaries are depicted in Fig. 5a.

Since the VFH+ algorithm wasn't able to complete the whole *Map 1*, an additional starting point has been placed after the narrow part of the passage (see Fig. 5c). The data are thus split into two sets of results. Table 11 presents values when the first starting point was chosen and it contains only the results of the DWA and SND algorithms. Table 12 then contains results for all three algorithms on a trajectory starting from the second starting point.

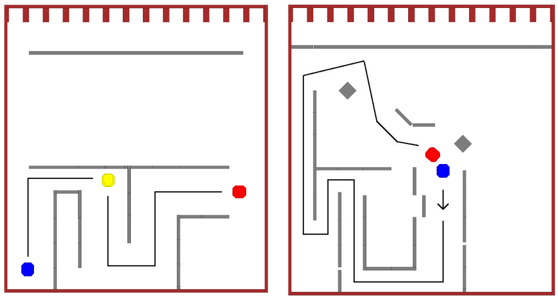


Fig. 4: The real environments in the SyRoTek arena. The narrowest passage of *Map 1* is 28 cm (and 60 cm when using the second starting point). The narrowest passage of *Map 2* is also 28 cm.

Table 9: Tested parameters for Map 1.

Driver	Number of combinations	Parameter	Min	Max	Step
DWA	36	robot_radius	0.08	0.12	0.02
		ValAngle	10	20	10
		ValVelocity	10	20	10
		ValObstacle	0	20	10
VFH	32	cell_size	0.01	0.02	0.01
		window_diameter	10	20	10
		safety_dist_0ms	0.05	0.09	0.04
		weight_desired_dir	5	9	4
		free_space_cutoff_0ms	$2 \cdot 10^6$	$3 \cdot 10^6$	$1 \cdot 10^6$
SND	27	robot_radius	0.04	0.08	0.02
		min_gap_width	0.12	0.16	0.02
		obstacle_avoid_dist	0.04	0.08	0.02

The SND algorithm is more reliable and faster in narrow passages than the DWA driver. On contrary, DWA driver is faster than the other two approaches when used in open areas. The VFH+ controller was successfully used only in environments, which don't contain narrow passages. In wider environments VFH+ achieved results similar to the other two controllers.

3.3 Discussion

In general, the SND driver has proved to be the most reliable both in simulations and experiments. The DWA algorithm had equal results only when used in environments wider than twice the diameter of the robot.

Table 10: Tested parameters for Map 2.

Driver	Number of combinations	Parameter	Min	Max	Step
DWA	48	robot_radius	0.07	0.10	0.01
		ValAngle	10	20	10
		ValVelocity	10	20	10
		ValObstacle	0	20	10
SND	27	robot_radius	0.04	0.08	0.02
		min_gap_width	0.12	0.16	0.02
		obstacle_avoid_dist	0.04	0.08	0.02

Table 11: *Map 1.* Results for the real robot, when using the first starting point. A, B, C are the best configurations from the initial testing, sim stands for the best configuration from simulation evaluation.

Algorithm	Configuration	t_{exp}	t_{min}	t_{max}	success rate
DWA	**A**	53.478	52.171	54.896	100 %
	B	53.860	51.830	56.792	100 %
	C	58.337	55.530	60.868	95 %
	sim	57.833	55.928	60.781	100 %
SND	A	61.441	59.680	67.977	100 %
	B	60.937	56.670	65.888	100 %
	C	63.321	61.117	68.186	100 %
	sim	63.055	57.268	67.715	100 %

The choice of the algorithm with best achieved times depends on the environment. In narrow environments the SND controller would be the best choice while the DWA algorithm could be recommended for environments with wider passages.

Both the SND and DWA algorithms should be easy to use. They have only a few important parameters and since the main parameters are connected with the robot radius, it is easy to determine how to set particular parameters, even for an unskilled user.

The VFH+ algorithm is inferior to other two controller in both reliability and speed. Its parameters in Player/Stage were originally set for a different robot and the description of the parameters is insufficient for a proper recalibration by someone unfamiliar to the driver.

4 Conclusion

The paper presents a comparison of three obstacle avoidance algorithms in a simulated environment and with a real robot in the SyRotek system. Over 20000

Table 12: *Map 1*. Results for the real robot, when using the second starting point. A, B, C are the best configurations from the initial testing, sim stands for the best configuration from simulation evaluation.

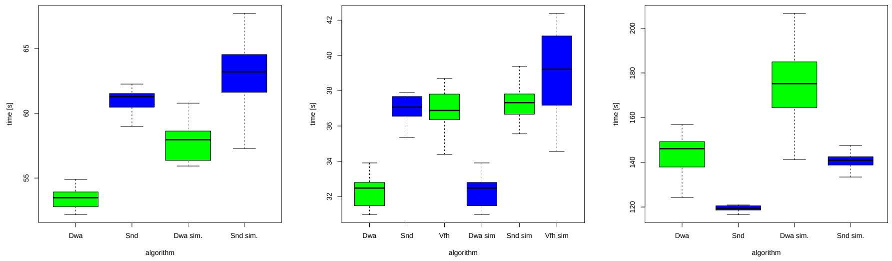
Algorithm	Configuration	t_{exp}	t_{min}	t_{max}	success rate
DWA	A	33.058	30.561	34.543	95 %
	B	33.456	30.418	35.402	100 %
	C	32.340	30.939	34.405	100 %
	sim	32.321	30.972	33.907	100 %
SND	A	37.602	35.856	39.933	95 %
	B	37.690	36.370	39.404	100 %
	C	37.071	35.358	37.884	100 %
	sim	37.249	35.557	39.390	100 %
VFH+	A	37.707	34.508	41.228	100 %
	B	37.502	34.106	40.962	100 %
	C	36.469	25.398	40.708	100 %
	sim	46.867	34.561	182.435	90 %

Table 13: *Map 2*. Results for the real robot.

Algorithm	Configuration	t_{exp}	t_{min}	t_{max}	success rate
DWA	**A**	143.153	124.298	156.942	95 %
	G	144.866	111.429	197.131	95 %
	B	153.975	131.620	167.875	90 %
	sim	174.318	141.168	206.722	80 %
SND	**A**	119.462	116.555	123.733	100 %
	G	120.788	117.771	129.183	100 %
	D	119.949	117.774	126.956	100 %
	sim	140.640	133.426	147.559	95 %

simulations in a simulator and 120 hours of experiments with a real robot of the SyRoTek system were made to find the best configurations for the drivers and to make a fair comparison of the algorithms.

After the best parameter configurations were found, a set of experiments was made to gain the results needed for the comparison of the controllers. The comparison proved the SND driver to be the most reliable one. The SND algorithm also achieved the best time results in narrow environments. The DWA controller was slightly inferior to SND in narrow environments, but in wider environments it proved to have equal results in reliability and equal or superior time results. The VFH+ controller was shown to be insufficient for the use in narrow environments. It doesn't offer any advantages over the other two controllers and it's use was accompanied with several difficulties.



(a) (b) (c)

Fig. 6: Graphic comparison of times achieved by each algorithm. (a) *Map 1*, the first starting point, (b) *Map 1*, the second starting point, (c) *Map 2*.

Acknowledgments.

This work has been supported by the Technology Agency of the Czech Republic under the project no. TE01020197 "Centre for Applied Cybernetics".

References

1. Borenstein, J., Koren, Y.: The vector field histogram-fast obstacle avoidance for mobile robots. Robotics and Automation, IEEE Transactions on 7(3), 278–288 (Jun 1991)
2. Durham, J., Bullo, F.: Smooth nearness-diagram navigation. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. pp. 690–695 (Sept 2008)
3. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. Robotics Automation Magazine, IEEE 4(1), 23–33 (Mar 1997)
4. Gerkey, B.P., Vaughan, R.T., Howard, A.: The Player/Stage project: Tools for multi-robot and distributed sensor systems. In: Proc. of the 11th Int. Conf. on Advanced Robotics. pp. 317–323 (2003)
5. Ingrand, F., Ghallab, M.: Robotics and artificial intelligence: A perspective on deliberation functions. AI Commun. 27(1), 63–80 (Jan 2014), `http://dl.acm.org/citation.cfm?id=2594611.2594619`
6. Kulich, M., Chudoba, J., Kosnar, K., Krajnik, T., Faigl, J., Preucil, L.: Syrotek – distance teaching of mobile robotics. Education, IEEE Transactions on 56(1), 18–23 (Feb 2013)
7. Minguez, J., Montano, L.: Nearness diagram (nd) navigation: collision avoidance in troublesome scenarios. Robotics and Automation, IEEE Transactions on 20(1), 45–59 (Feb 2004)
8. Ulrich, I., Borenstein, J.: Vfh+: reliable obstacle avoidance for fast mobile robots. In: Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on. vol. 2, pp. 1572–1577 vol.2 (May 1998)