Preprints of the 2012 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, July 3-6, 2012

A Novel Algorithm for Following of Moving Target in Outdoor Mobile Robot Environment Based on Inverse Matching

Harun Šiljak and Jasmin Velagić
Faculty of Electrical Engineering, Zmaja od Bosne bb, 71000 Sarajevo, Bosnia and Herzegovina
Email: hs14938@etf.unsa.ba, jasmin.velagic@etf.unsa.ba

***Abstract*—This paper proposes a new algorithm for following a mobile robot target in an outdoor environment. This approach combines a fuzzy tracking control law and an inverse matching based algorithm for finding and following of the moving target. The inverse matching algorithm comprises local maps of a robot environment to identify the target and generate desire path toward it. The localization of the mobile robot is done by the triangulation method based on GPS measurements. In addition, previously developed software system for the mobile robot enables GPS guidance and obstacle avoidance which makes it suitable for creation of mobile convoys. Finally, the experimental verification is performed to confirm the effectiveness and practical value of the proposed approach.**

***Index Terms*—Mobile robot, Robot convoy, Inverse matching, Fuzzy control, Target following.**

I. Introduction

The estimation of positions of the moving target and its following in unstructured environments are challenging tasks in mobile robotics. Two separate approaches can be considered for solving these problems: based on measurements of distance sensors[1] and use of vision [2]. Controllers used for following of the moving target by the mobile robot in each of these two approaches may be various. The fuzzy logic based approaches assert as potential effective ways to cope with uncertainties which are occurred in sensor measurements and accurate estimation of the moving target location [3]. Recently, the fuzzy control has been used very successful for tracking multiple moving objects with a mobile robot and for a robotic convoy formation [4].

The mobile robot localization and mapping in outdoor environments is a challenging task within navigation system, and various solutions for these tasks have been proposed [5]. It is well known that the concept of matching is often used in the mobile robot localization [6], but in our paper it is used in reverse to determine changes in the environment. This is known in mobile robotics under the name novelty detection [7], [8]. The novelty detection method exploits results of localization and mapping in solving the problem of robot based following of mobile targets and formation of mobile convoys. The robotic convoy is an array of mobile robots in a stable string led by a robot which is called the convoy leader.

In unstructured environments the target robot can exhibits unpredictable motions and the most used estimation techniques, such as EKF (Extended Kalman Filter), become impractical [9]. The second problem is appeared when the target robot is not visible and out of sensor's range. The particle filters have been applied with great success to visual tracking of the moving target [10]. Unfortunately, they exhibit a poor tracking performance when the moving target is not visible by used visual system or there is unpredictable motions such as camera shake [11]. In this paper we present an approach for tracking the target robot which attempts to overcome these problems. This approach is suitable for the mobile robot convoy formation.

Since the proposed following algorithm is used for formation of the mobile robot convoy, there was a need for development of the robotic convoy leader able to move autonomously toward a fixed target. Its location is given in terms of longitude and latitude via GPS while avoiding obstacles in its way. Also, it has to be capable to move with manual controls, since in the phase of system testing it has to investigate options of tracking avoidance (another important concept in this area of mobile robotics [12]) and verify the quality of the tracking algorithm.

Our approach combines the fuzzy logic based tracking algorithm and inverse matching method for estimation of a moving target location and its following within unstructured outdoor environments. This approach is capable to follow the moving target which is outside of mobile robot sensors readings and therefore is suitable for robots without sensors parallel to their heading (such as Pioneer 3DX). We introduce the inverse matching method to determine changes in the environment in order to improve tracking capabilities. This method ensures that the robot convoy formation is enabled even without a communication channel for members of the team.

The paper is organized as follows. Section II introduces the convoy software and hardware platform which realizes their specific capabilities relevant for the following of moving target concept. In Section III more details about the new fuzzy tracking algorithm are given. Section IV presents the new inverse matching algorithm for the following of moving target. Experimental results are discussed in Section V. Some

978-1-4673-2529-5/12/$31.00 ©2012 IEEE

concluding remarks and future directions are given in Section VI.

II. System Description

The overall proposed target following system for outdoor environments is realized through client-server architecture ilustrated in Fig. 1. Taking into account that all these functionalities do not require a server, but a centralized server-client architecture may be a good feature for certain tasks - surveillance, not for control [13]. In our previous paper [14] we have already developed hardware architecture in Fig. 1 and some appropriate software components. The software modules were realized using Matlab and Aria packages for Pioneer 3DX mobile robot platform.

In this paper the system depicted in Fig 1. is expanded by adding new functionality which enforced this system to deal with problems of following the moving target and the robot convoy formation. The features of the software implemented on clients and the optional server are presented in Fig. 2. All communication routes in this system are completely optional and every team member can act autonomously without information or instructions from the server or other group members, assuming a task has been assigned to it. Task-specific diagram for clients (i.e. mobile robots) is shown in Fig. 3.

As far as development process of the software is concerned, spiral concept has been used, in which features were added gradually. First feature was the mobile target following algorithm, which has been tested and validated on simple trajectories of another robot as the object of following. Next step was the development of simulator for [14] which has taken the role of convoy leader in tests to come. After making sure the follower performs its task well in an environment without obstacles, multiple improvements on the algorithm (such as following the closest object) were tried out in order to make the algorithm operate in a populated environment. The only successful extension was inverse matching concept. This whole process is shown in Fig. 4.

As the final step of the spiral development process, leader and follower programs were integrated into one GUI and tested. Results will be given in Section V.

A. GPS Targeting

The hardware and software structure described in [14] makes it possible to obtain GPS data for the mobile robot in regular time intervals via Cinterion (former Siemens) module with integrated GPRS, which allows sending the data to the centralized server and manipulating them both on the server and client computer. This allows the server to give out instructions (i.e. target point) and observe robotic movement, while the client makes sure the targeting algorithm is being executed properly.

The algorithm itself is rather simple - in case no obstacles are in the way, the mobile robot directs itself straight to the target point and moves on the line. In case obstacles appear, the algorithm pauses until the obstacles are avoided and so on.

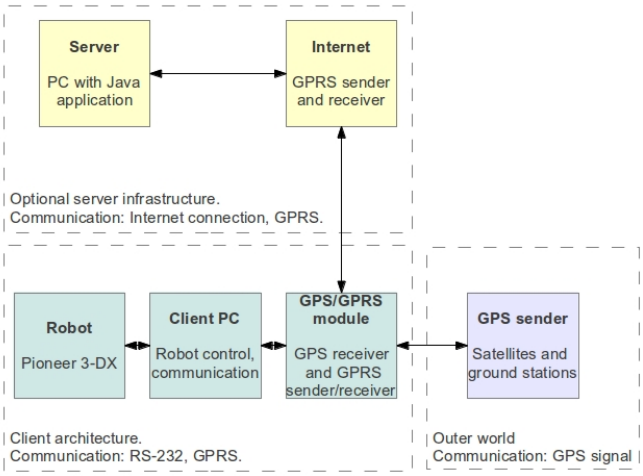


Fig. 1. Client-server hardware and communication architecture.

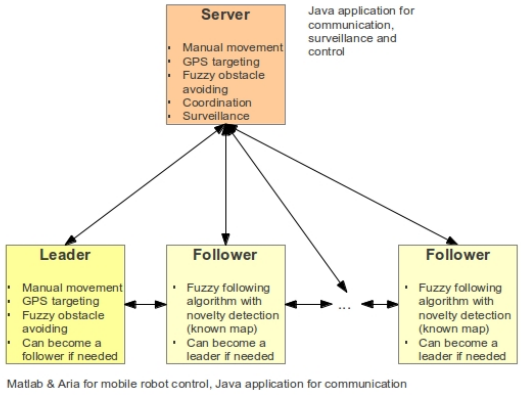


Fig. 2. Client and server functionalities with possible communication routes.

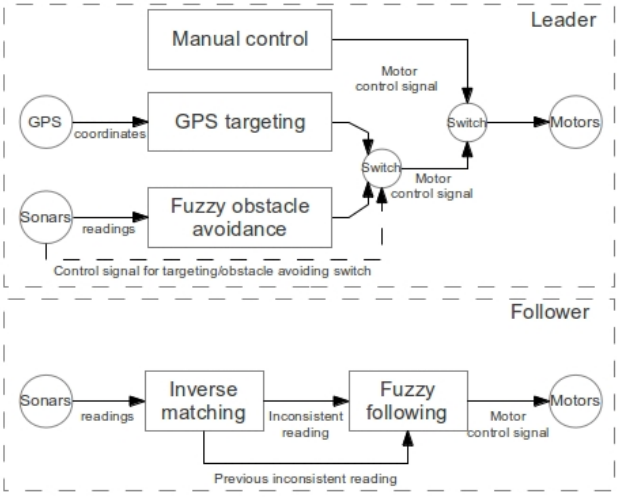


Fig. 3. Client functionalities block scheme.

Robotic movement is followed on a satellite image taken from Google Earth. As an addition to system developed in [14], current system now features a GPS simulator, using

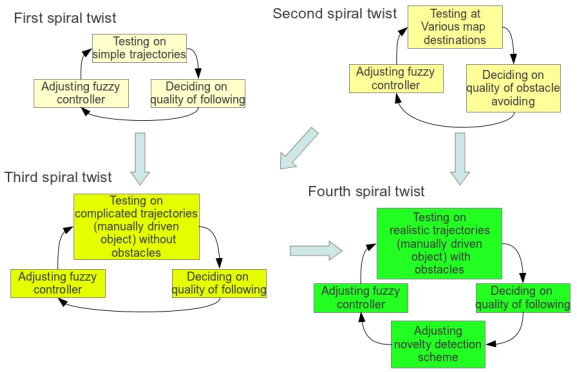


Fig. 4. Spiral approach illustration.

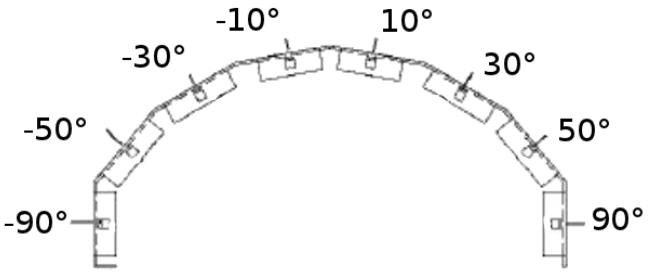


Fig. 5. Sonar placement on Pioneer 3 mobile robot.

odometry to simulate GPS and custom-made line maps to simulate the environment.

B. Obstacle Avoidance

The obstacle avoidance algorithm is based on the Mamdani fuzzy controller with only two inputs are used from sonar sensors - namely two central ones shown in Fig. 5. It is a significant change compared to [14] where four sensors were used. Although the set of input is reduced, quality of obstacle avoiding is still on a high level, and combined with previously presented GPS targeting makes it possible for the robot to travel to even the hardest target points, just like the one in Fig. 6.

The target point is placed deliberately behind a concave obstacle which becomes a trap for the robot, but even this simple algorithm makes its way out of it and continues its traverse towards the target point. The screenshot in Fig. 6 is just a part of the robot GUI made in Matlab which controls it all. The whole GUI representation can be seen in Figs. 11 and 12.

The principle is a well known one - the closer the obstacle is, the sharper the turn, with an option of going backwards in case of sudden very close encounters. Two sensors make it possible to choose which side will the robot turn. Full functionality is made possible with 7 rules, although that rule set could have been reduced as well.

Fig. 6. Movement of mobile robot towards a goal point (blue circle - top right is the start, green circle - bottom right is the target position, red crosses showing the GPS readings of robot position).

C. Moving Target Following

Target following process will be described in more details in following sections, here we only note that it is implemented in the same GUI as the previous features, therefore every robot can be both convoy leader and a plain member (follower).

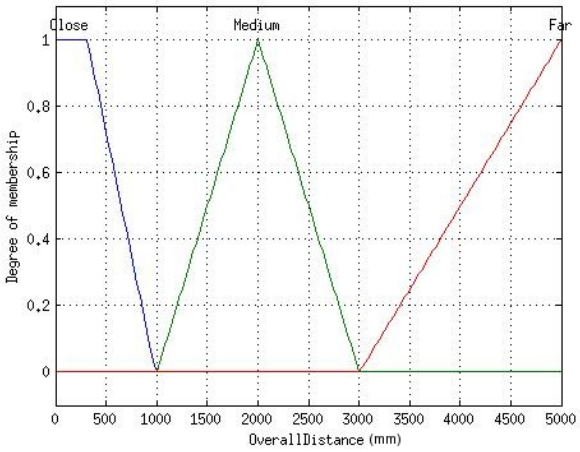
III. Fuzzy Tracking Algorithm

The traditional form of the fuzzy algorithm for the mobile target following is well known and can be described in terms of two parallel sensors placed on two sides of robot's axis of symmetry. Velocity of the follower is adjusted according to the distance reading on the sensors - while both sensors register the followed object. When only one sensor senses the object, robot makes a turn in that direction [3].

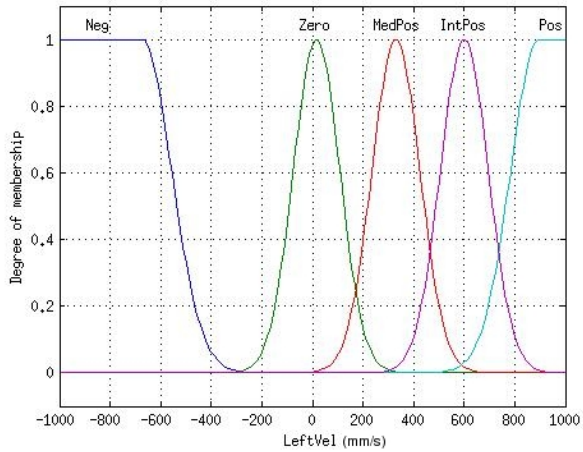
Indeed, this simple concept is applicable for custom-made mobile robots with onboard sensors. On the other hand, Pioneer 3DX has a disadvantage in its sensors placement. Two central sensors leave 20° angle between them and we can expect that most of the time followed object remains in that area, which is shown in Fig. 5.

Therefore, this algorithm had to introduce new corrective input, since there is no other way to adjust the velocity of follower with the velocity of leader. Tests have shown that in most applications this corrective input is good enough to ensure good following, with performance close to performance of other following algorithms on more suitable hardware platforms.

The fuzzy controller is Mamdani, with nine inputs and wheel velocities as outputs. First eight inputs are current sensor readings while the last one is most recent measurement under 5000 millimeters (that is the limit of measurements for these sensors). While first eight inputs control the turns - farther the sensor is from the axis, faster the turn - last input controls the robot velocity while moving straight ahead. Eleven single-variable rules were enough for the first version, while more rules on the speed control may be imposed so better discretization of velocity is achieved. It is worth mentioning that rules on the speed control are weighted by 1, while turning rules are weighted by 0.1. Reasoning behind it is simple - that way the turning is customized in spirit of velocity control.



(a)



(b)

Fig. 7. Membership functions for following. a - membership function for *OverallDistance* input, b - membership function for velocity output.

TABLE I
FOLLOWING FUZZY RULES

IF	THEN LeftVel	RightVel	Weight
OverallDistance is Close	Zero	Zero	1.0
OverallDistance is Medium	MedPos	MedPos	1.0
OverallDistance is Far	Pos	Pos	1.0
Sonar1 is InSight	Neg	Pos	0.1
Sonar2 is InSight	Zero	MedPos	0.1
Sonar3 is InSight	MedPos	IntPos	0.1
Sonar4 is InSight	IntPos	Pos	0.1
Sonar5 is InSight	Pos	IntPos	0.1
Sonar6 is InSight	IntPos	MedPos	0.1
Sonar7 is InSight	MedPos	Zero	0.1
Sonar8 is InSight	Pos	Neg	0.1

The exact values for weights were determined through trial and error process.

Fig. 7 depicts respective membership functions of inputs and outputs (membership functions of sensor inputs 1-8 are not shown since they are boolean - the only set is named *in sight* and means the input is less than 4900 mm). If it isn't *in sight*, it is ignored, as one can see in Table I where the rules are presented. Else, it is considered as a trigger for turning of the robot. Depending on the magnitude of *overallDistance* which represents the least measurement from the last sensor reading vector velocities of wheels vary, either in turning mode, either while going straight. That is a consequence of good weight placement.

This algorithm cannot make a difference between the object being followed and obstacles in the environment. In order to overcome that we introduce the inverse matching algorithm for recognizing the real target in presence of various static obstacles. This algorithm is presented in the next section.

IV. INVERSE MATCHING ALGORITHM FOR FOLLOWING OF MOBILE TARGET

The concept of *regular* matching is simple: the robot compares readings of its sensor from the environment with the memorized map, or parts of map. As soon as this matching gives a satisfying result, that gives a location for the robot, according to the place on the map his readings match [6].

Unlike this approach, we are doing the opposite, and assume that the localization using GPS measurements has given us correct robot position (this assumption will be dealt with later). Using those coordinates, the robot can compare its sensor readings to those expected at that location in the respective part of map. If the readings match within certain error margin, we can consider that no foreign entities are sighted, as done in the more general *novelty detection* [8].

In case that readings show a significant difference in one or more directions, then we can declare a foreign entity/intruder/leader has been spotted. Different names stand for different purposes of application. In the case that our robot is doing a surveillance check, foreign object is intruder. In situation when it is in a robotic convoy, the foreign object by default is the group leader or one of other group members which should be followed.

The mathematical explanation of this novelty detection variant which we call *inverse matching* is rather simple and elementary, knowing the properties of Pioneer 3DX robot: its sensors are placed as in Fig. 5, with respective coordinates (70,130), (100,100), (130,70), (170,20), (170,-20), (130,-70), (100,-100), (70,-130) from the left to the right in a coordinate system with origin in robot's center. Abscissa (x) being the axis of symmetry of robot and the sensor belt and ordinate (y) being perpendicular to it in robot's plane - coordinates taken in millimeters.

Let us denote ith sensor's angle from the Fig. 5 with θ_i and its coordinate pair as (x_{si0}, y_{si0}), for $i = 1, 2, \ldots, 8$. Furthermore, denote the current robot position (in a coordinate system with origin in robot's starting position and coordinates given in millimeters) with (x_c, y_c) and the current heading

with θ_c (for convenience, we will take this angle in degrees to match the angles in Fig. 5).

Straightforward calculation yields the formula for location of ith sensor in that case:

$$x_{si} = x_c + \sqrt{x_{si0}^2 + y_{si0}^2} \cos\left(\theta_c + \arctan\frac{y_{si0}}{x_{si0}}\right) \quad (1)$$

$$y_{si} = y_c + \sqrt{x_{si0}^2 + y_{si0}^2} \sin\left(\theta_c + \arctan\frac{y_{si0}}{x_{si0}}\right) \quad (2)$$

Sensor rays are lines (rays are not the same as lines, but for now we will break that convention) defined with

$$\frac{y - y_{si}}{x - x_{si}} = \tan(\theta_c + \theta_i) \quad (3)$$

It is necessary to find out the location of intersections of these rays with closest obstacles. Assume that map is stored in form of line segments, defined with two endpoints. Take one jth such line segment and denote its endpoints with (x_{1j}, y_{1j}) and (x_{2j}, y_{2j}). Then the line equation is

$$\frac{y - y_{1j}}{x - x_{1j}} = \frac{y_{2j} - y_{1j}}{x_{2j} - x_{1j}} \quad (4)$$

Now it is possible to find intersection of that line and the sensor ray. Denoting $k_{si} = \tan(\theta_c + \theta_i)$ and $k_{12j} = \frac{y_{2j} - y_{1j}}{x_{2j} - x_{1j}}$ hence the intersection points are given by:

$$x_{ij} = \frac{\frac{y_{1j} - y_{si} - k_{12j} x_{1j}}{k_{si}} + x_{si}}{1 - k_{12j}/k_{si}} \quad (5)$$

$$y_{ij} = \frac{y_{1j} - k_{12j}(x_{1j} - x_{si} + y_{si}/k_{si})}{1 - k_{12j}/k_{si}} \quad (6)$$

After calculating all relevant intersections in the map part, there are open questions that have to be answered: what if the intersection is outside the line segment but on the line, what if there are more line segments intersecting the ray and what if the obstacle is behind the robot, but using formulas given above we calculated a valid intersection?

Intersection outside the line segment has a distinctive property of its distance from one of the end points being larger than the entire line segment. Checking the length hence suffices in settling this question. As far as the second question is considered, quite logically - the closest interception point is the one registered by sensors. If we don't have an interception point closer than 5000 millimeters, the sensor reading is set to 5000, due to its limit distance of 5 meters.

The third question (distinction between line and ray) is settled via coordinate system transform. First step of the transform is to translate the origin in robot's center. Second step is rotation of axes in such manner that ordinate coincides with robot's axis of symmetry. That way transformed points of intersection are visible (i.e. on the right side of the ray) if their ordinates are positive. This condition is reduced to a simple rule:

$$(x_{ij} - x_c)\cos(\theta_c + \theta_i) + (y_{ij} - y_c)\sin(\theta_c + \theta_i) > 0 \quad (7)$$

Finally, we have to consider the aspect of errors in localization, sensor readings and map discretization. Two options were tested: moving average filtering of sensor results, as well as waiting for two consecutive detections of foreign entities. Still, in the experiment presented in this paper, only error margin has been used while detection is not reported until error is larger than some predefined nonzero value. Inverse matching for novelty detection is being made in more than one point in neighborhood of reported GPS position. On the practical side, it has to be noted that GPS is combined with local odometry between two GPS reports.

In summary, the inverse matching algorithm can be presented in pseudocode as follows:

loop
  $lastReceived(value, index) \leftarrow (5000, 4)$
  $count \leftarrow 0$;
  while $count < MultipleAckNumber$ **do**
Input: $location \leftarrow GetGPS()$
Input: $readings(1 : noOfSensors) \leftarrow GetSensor()$
    $mapPart \leftarrow GetMapPart(location)$
    for $i = 1 \rightarrow noOfSensors$ **do**
      $expect(i) \leftarrow GetIntersect(mapPart, location, i)$
      if $expect(i) - readings(i) \geq errorMargin$ **then**
        $count \leftarrow count + 2$
        $anomaly(value, index) \leftarrow (readings(i), i)$
        break
      end if
    end for
    $count \leftarrow count - 1$
Output: $last \leftarrow lastReceived$
  end while
Output: $current \leftarrow anomaly$
  $lastReceived \leftarrow anomaly$
end loop

MultipleAckNumber is the number of consecutive anomalous readings which are enough to claim that the foreign object is placed there (it is not necessarily 1, due to possible measurement errors). Expect(i) is a term containing the expected value of i-th sensor reading. Change of the variable count may seem strange, but the principle is as follows: every new acknowledgement of anomaly increments the count by one, but if the new iteration does not include an anomaly, the count does not go down to zero instantly, it decrements by one instead. That is much more natural, since readings in one, two, three iterations may be flawed due to errors. As far as the outputs named last and current are concerned, last is the one doing the distance control, while current is one controlling the turn (Fig. 3 shows both outputs as fuzzy controller inputs). Since current is only active after the entire loop is executed, and last is perpetually active, the pseudocode notation is partly abused to emphasize that fact.

V. Experimental Results

In order to demonstrate the effectiveness and usefulness of the proposed approach, a series of simulations and experiments

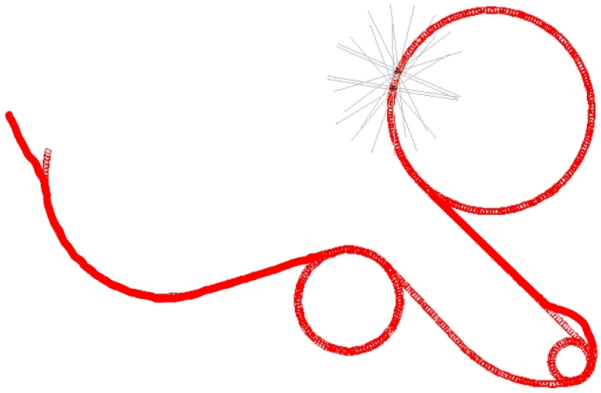


Fig. 8. Simple trajectory of the leader in obstacle-free environment (full line is the follower's trajectory, dashed line is the leader's).

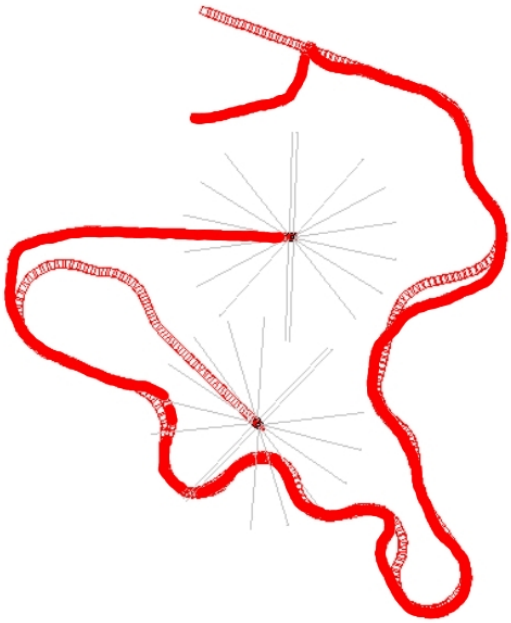


Fig. 9. More complex trajectory in obstacle-free environment (full line is the follower's trajectory, dashed line is the leader's).

has been conducted. The first scenario considers following of the moving target in simulated obstacle free environment. In the case of simple trajectories of the leader (first testing loop in Fig. 4) in which velocities are kept constant tracking is almost ideal, as shown in Fig. 8.

When the leader was manually driven through the same environment and velocities deliberately changed so it could escape (third testing loop), it would evade the pursuit, but only after several such maneuvers, as shown in Fig. 9.

Experimenting with the algorithms in populated environments (fourth, final testing loop) has shown the trade off between the robustness and quality of following. If we choose more deterministic inverse matching in novelty detection, false

Fig. 10. Trajectories of the leader and follower robots.

alarms at sharp corners make it more difficult for the followers to keep up with the leader, if the leader does not put in additional effort to lead them properly. On the other hand, if we choose more probabilistic approach to inverse matching, detection of leader's position is becoming worse.

The question of probabilistic inverse matching goes out of the scope of this paper and may be a topic for the future research. The probabilistic manner solves one of problems that are likely to appear in practice - the problem of small environment changes due to different unpredictable factors.

Depending on the particular application, these *security levels* can be set up in the robot's operating code. For this demonstration we used maximal deterministic inverse matching with a manually driven leader. In this situation the velocity is changed from time to time, to test the ability of follower to keep track.

Trajectory is deliberately chosen through a corridor, so we may be sure that the robot doesn't mistakenly follow the nearby wall instead of the convoy leader. As can be seen in Fig. 10, the trajectory is not a simple one (moreover, take into account artificial velocity changes), but achieved results seem to be excellent, as shown in Figs. 11 and 12. This configuration has been chosen as a representative one, but it does not mean it is the most complicated one this algorithm can deal with. The future work may consider testing of the algorithm on different environment configurations.

There could have been more robots in this experiment, but two are enough to demonstrate the quality of following provided. For reference, dimensions of the map shown in Fig. 10 are 124.05 m $\times$ 61.22 m and the maximum velocity achieved by both robots was 1 m/s.

VI. Conclusion

The paper deals with problem of moving target following by mobile robot in outdoor environments. The obtained simulation and experimental results indicate the effectiveness and usefulness of proposed algorithms. Considering the fact that mobile robot used for following obviously wasn't designed for this kind of task, one can tolerate it not being as optimal as some other designs may be. Nevertheless, certain improvements will be considered in the future.

In our future work, we will compare our approach and its superiority regarding the other approaches, especially in terms

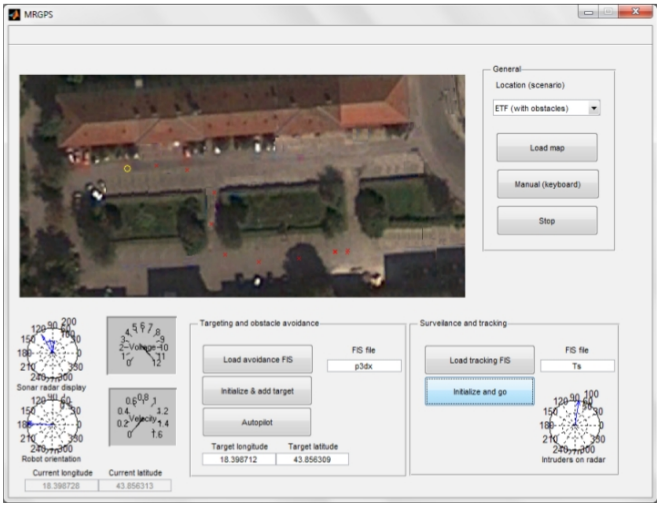


Fig. 11. Same trajectory shown in our robot software, follower's view (yellow circle - top right is the start, red crosses showing the GPS readings of robot position).

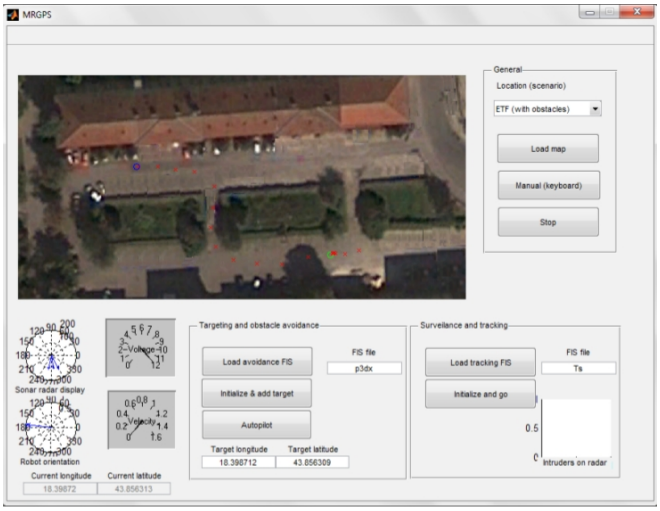


Fig. 12. Same trajectory shown in our robot software, leader's view (blue circle - top right is the start, green circle - bottom left is the target position, red crosses showing the GPS readings of robot position).

of quantitative measurement of performance, time complexity, etc.

In this paper it is important to address a certain question that may arise. First of all - is the request for all followers in a mobile convoy to have a map of the area they are moving in too hard to achieve?

It is not, even if the environment is initially unknown to the convoy's members. The convoy leader moves through the environment first. The sensor measurements of obstacles in its vicinity may be transmitted via GPRS or ZigBee to the followers and that way they may form a partial map of local area, which is then used for matching and locating the convoy leader and/or other convoy members. As an alternative, map can be learned through SLAM.

Furthermore, the possibility of applications of this system is also important. Beside the classical applications of mobile target following and convoy forming mobile robot systems, this particular solution may be used for surveillance because of the novelty detection concept.

REFERENCES

[1] A. Korodi, A. Codrean, L. Banita and C. Volosencu, "Aspects Regarding the Object Following Control Procedure for Wheeled Mobile Robots", *WSEAS Trans. on Systems and Control*, Vol 3 pp. 537-546, 2008.

[2] L. Freda and G. Oriolo, "Vision-based interception of a moving target with a nonholonomic mobile robot", *Robotics and Autonomous Systems* 55, pp. 419-432 2007.

[3] I. Ullah, F. Ullah and Q. Ullah, "Real-time object following fuzzy controller for a mobile robot", *ICCNIT Proceedings*, pp. 241-244, 2011.

[4] C. Santos, F. Espinosa, D. Pizzaro, F. Valdes, E. Santiso and I. Diaz, "Fuzzy Decentralized Control for guidance of a convoy of robots in non-linear trajectories", *IEEE Conference ETFA Proceedings* pp. 1-8 2010.

[5] J. A. Castellanosand J. D. Tardos, *Mobile Robot Localization and Map Building - A Multisensor Fusion Approach*, Springer Verlag, Berlin, Germany, 2000.

[6] J. Minguez, F. Lamiraux and L. Montesano, "Metric-Based Scan Matching Algorithms for Mobile Robot Displacement Estimation", *IEEE International ICRA Conference Proceedings*, pp. 3557-3563, 2005.

[7] P. Chakravarty, A.M. Zhang, R. Jarvis and L. Kleeman, "Anomaly Detection and Tracking for a Patrolling Robot", *Proceedings of ACRA*, 2007.

[8] M.F. Miskon and R.A. Russell, *Mapping normal sensor measurement using regions*, Proceedings of IEEE ICIT, 2009.

[9] H. Min, N. Papanikolopoulos, C.E. Smith and V. Morellas, "Feature-based Covariance Matching for a Moving Target in Multi-robot Following", *19th Mediterranean Conference on Control and Automation (MED), Corfu, Greeece*, pp. 163-168, 2011.

[10] D. Schulz, W. Burgard, D. Fox and A.B. Cremers, "Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association", *IEEE International ICRA Conference Proceedings*, pp. 1665-1670, 2001

[11] M. Pupilli, A. Calway, "Real-time Visual SLAM with Resilience to Erratic Motion", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1244 1249, June 2006.

[12] T. Chung, G. Hollinger and V. Isler, "Search and Pursuit-Evasion in Mobile Robotics", unpublished manuscript.

[13] B. P. Gerkey, R.T. Vaughan and A. Howard, "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems", *ICAR Conference Proceedings*, pp. 317-323, 2003.

[14] J. Velagic, N. Osmic, F. Hodzic and H. Siljak, "Outdoor navigation of a mobile robot using GPS and GPRS communication system", *ELMAR Proceedings*, pp. 173-177, 2011.