SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 10, No. 1, February 2013, 59-72

UDK: 621:007.52]:004.021 DOI: 10.2298/SJEE1301059C

An Algorithm for Formation Control of Mobile Robots

Aleksandar Ćosić[1], Marko Šušić[1], Stevica Graovac[2], Duško Katić[1]

Abstract: Solution of the formation guidance in structured static environments is presented in this paper. It is assumed that high level planner is available, which generates collision free trajectory for the leader robot. Leader robot is forced to track generated trajectory, while followers' trajectories are generated based on the trajectory realized by the real leader. Real environments contain large number of static obstacles, which can be arbitrarily positioned. Hence, formation switching becomes necessary in cases when followers can collide with obstacles. In order to ensure trajectory tracking, as well as object avoidance, control structure with several controllers of different roles (trajectory tracking, obstacle avoiding, vehicle avoiding and combined controller) has been adopted. Kinematic model of differentially driven two-wheeled mobile robot is assumed. Simulation results show the efficiency of the proposed approach.

Keywords: Formation control, Fuzzy control, Mobile robots.

1 Introduction

Formation control is an important field in multi-robot coordinated control, which has recently triggered great interest of the research community. Group of mobile robots show obvious advantages over single autonomous vehicle, including greater flexibility, adaptability and robustness. Team of mobile robots can efficiently solve tasks such as space exploration, transportation of large objects, security tasks, group hunt, etc.

Approaches in formation control can be divided into several categories: *behaviour based approaches, virtual structures methods, leader – follower approaches, potential fields and generalized coordinates methods* [1]. In behaviour based methods, group behaviour or mission consists of a number of primitive decentralized actions (subtasks), synthesized in order to achieve global goal [2], while the control action is obtained as a combination of these primitives. In leader – follower approaches, one of the robots in formation is

[1] Mihajlo Pupin Institute, University of Belgrade, Volgina 15, 11000 Belgrade, Serbia;
E-mails: aleksandar.cosic@pupin.rs, marko.susic@pupin.rs, dusko.katic@pupin.rs

[2] School of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, 11000 Belgrade, Serbia;
E-mail: graovac@etf.rs

designated as a leader, while the others are designated as followers and they are forced to track the leader, maintaining a specified geometric arrangement [3]. In virtual structures, robots are considered as particles, inserted into rigid virtual structure, which represents the whole formation [4]. Hence, whole formation is considered as a single rigid structure. General approach to modelling and control of mobile robot formations based on generalized coordinates is given in [5]. This method ensures asymptotic convergence of the realized trajectory to the desired one, even if the trajectory is curved, keeping the desired formation shape. Another decentralized approach is based on potential fields [6]. In this method, various virtual forces are assigned to individual robots, obstacles and desired formation shape, and they are combined and used to move robot to its desired position in the formation.

Leader – follower approach is adopted in this paper. One of the robots is designated as a leader, which is forced to track the given collision free trajectory, generated by the high level planner, given in [7]. The followers should track the leader, keeping the desired formation shape. Desired trajectories of the followers are generated based on the trajectory realized by the leader robot. In order to obtain smooth trajectories, approach based on curvilinear coordinates is utilized, given in [9]. Formation has to be dynamic, i.e. it must be able to change its shape, depending on environmental conditions and obstacle presence. This approach can be easily implemented in practice. Comparing to virtual structure methods, leader – follower approach can realize time varying formation shape, which is very important property in complex environments with narrow passages. Due to decentralized control, stability of the formation can be guaranteed even when the uncertainties and disturbances are significant. Consequently, this method is more suitable for practical applications than generalized coordinates.

2 Kinematic Model of Mobile Robot

Schematic model of the two-wheeled mobile robot is shown on Fig. 1. World coordinate frame is denoted by $\{X, O, Y\}$, while $\{x_l, COM, y_l\}$ denotes local coordinate frame, attached at the robot. Origin of the local coordinate frame is placed at the robot centre of mass (COM). State variables are position and orientation of the robot, i.e., COM position (x, y) and angle φ between x axes of the world and local coordinate frame, while ω_L and ω_D denote angular velocities of the left and right side wheels of the robot, respectively, and represent control inputs. Linear velocity of the robot is denoted by v_c. It can be noted that velocity vector coincides with the x_l axis in the absence of slipping. Derivation of kinematic equations is given in [8]. Kinematic model of the robot is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\varphi & \frac{r}{2}\cos\varphi \\ \frac{r}{2}\sin\varphi & \frac{r}{2}\sin\varphi \\ -\frac{r}{2b} & \frac{r}{2b} \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_D \end{bmatrix}. \tag{1}$$

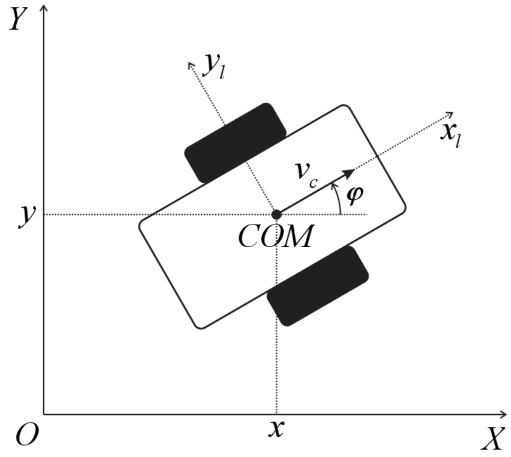

Fig. 1 – *Kinematic model of the two-wheeled mobile robot.*

3 Generation of the Followers' Desired Trajectories

Formation control task is solved using leader – follower approach, i.e. leader robot moves in space with obstacles, avoiding them, while the followers should track the leader, keeping the desired formation shape. If the formation cannot be maintained, i.e. if the collision between follower and obstacle is probable, formation switching must happen (e.g., formation has to change its shape to convoy), in order to ensure safe passing through obstacles. When all vehicles come to safe area, formation has to be maintained again.

Ideal differentially driven robot can turn on spot, therefore, it can realize arbitrarily curved trajectory. Unfortunately, formation cannot be turned on spot. Hence, perfect formation cannot be maintained during turning, therefore, concession must be made. One approach is to maintain formation in curvilinear, rather than in the original rectilinear coordinates, as given in [9]. This situation is depicted on Fig. 2. If the reference point of the whole formation is leader's COM, position of every follower can be described by two parameters: p_i – distance from leader to i-th follower along the leader's path and q_i – normal distance from follower to leader's path. Position and orientation of the leader is denoted by (x^L, y^L, φ^L), whereas the position and orientation of the *i*-th follower by $(x^{F_i}, y^{F_i}, \varphi^{F_i})$. Let the current time instant be denoted by *t* and time instant when the leader was at the distance p_i away from its current position along the trajectory by t_f. Current position and orientation of the *i*-th follower can be evaluated using the following formula:

$$\begin{aligned} x^{F_i}(t) &= x^L(t_f) + q_i \sin\varphi^L(t_f) \\ y^{F_i}(t) &= y^L(t_f) - q_i \cos\varphi^L(t_f) \\ \varphi^{F_i}(t) &= \varphi^L(t_f) \end{aligned} \tag{2}$$

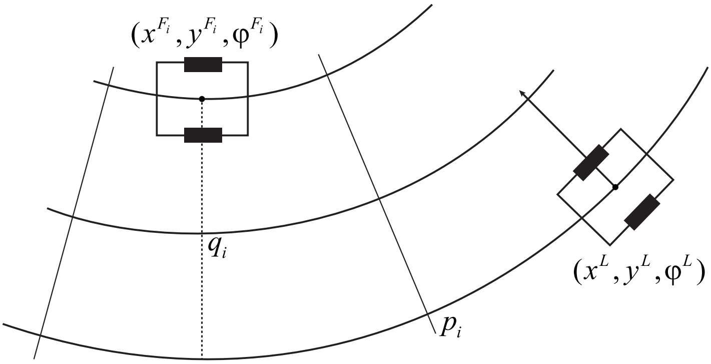

Fig. 2 – *Simple formation during turning.*

4 Control System Structures

In the previous step, desired trajectories of the followers are generated, based on trajectory realized by the leader robot. Formation control structure will be proposed in this section. Control system of each follower has the following controllers: trajectory tracking (TTC), obstacle avoiding (OAC), vehicle avoiding (VAC) and combined controller (CC). Each controller has a different function. TTC must provide tracking of the reference trajectory, OAC and VAC become active when the robot comes close enough to the object of the environment (obstacle or another vehicle), while the CC has to made concession between individual control actions, depending on current situation in the environment. That means that every individual controller generates its own control action, while the CC combines them, depending on situatuion in the environment, into single control. Control system of the leader does not take care of the followers, i.e. it does not have VAC part. TTC is nonlinear proportional-integral (PI) controller, because controllers of this type are widely used in industrial practice. OAC and VAC are fuzzy logic based controllers (FLCs).

4.1 Trajectory tracking controller (TTC)

Tracking controller should generate control action which tries to direct robot to the desired trajectory. This action is mainly achieved by the proportional term. Also, controller should have integral term in order to decrease an error in stationary state. Let (x, y, φ) denote robot position and orientation, while desired position in the same time instant is denoted by (x^*, y^*, φ^*), and desired velocity by $\boldsymbol{v}^*$. Velocity generated by controller is denoted by $\boldsymbol{v}_z$ and can be written as:

$$\boldsymbol{v}_z = \boldsymbol{v}^* + \Delta\boldsymbol{v}_z ,$$

$$\Delta\boldsymbol{v}_z = \begin{cases} \dfrac{K_p\left(\|\boldsymbol{e}_z\|/k - d\right)}{\|\boldsymbol{e}_z\|}\left(\boldsymbol{e}_z + \dfrac{1}{T_i}\boldsymbol{e}_{zi}\right), & \|\boldsymbol{e}_z\| > d, \\ 0, & \|\boldsymbol{e}_z\| \le d, \end{cases} \tag{3}$$

$$d = d_0 - \min\left\{d_0, \|\boldsymbol{v}^*\|/k\right\},$$

where $\Delta\boldsymbol{v}_z$ is velocity correction, k is positive gain and d is the dead-zone size, dependent on $\boldsymbol{v}^*$. Tracking error and integral of tracking error are denoted by $\boldsymbol{e}_z = \left[x^* - x \quad y^* - y\right]$ and $\boldsymbol{e}_{zi} = \int \boldsymbol{e}_z \,\mathrm{d}t$, respectively. Proportional gain is denoted by K_p, whereas T_i stands for integral constant.

As can be seen from (3), velocity correction is chosen as nonlinear function of errors sum, i.e. dead-zone around desired point is introduced. Introducing nonlinearity is necessary, because it decreases oscillations of robot position when it comes close enough to the desired point. Dead-zone changes its size depending on desired velocity, i.e., it decreases when desired velocity increases. Parameter d_0 determines the size of the dead zone when desired trajectory approaches destination point, i.e. maximal value of tracking error when real vehicle approaches the destination point.

Simple anti-windup algorithm is adopted, i.e., integral term “freezes” on the previous value, when one of the motors saturates.

It can be seen from kinematic equation (1) that angular velocities of the motors (ω_L, ω_D) are actually weighted sums of the linear and angular velocities of the robot $(v_c, \dot{\varphi})$. So, angular velocities generated by the controller (ω_{Lt}, ω_{Dt}) are:

$$\omega_{Lt} = a_v \|\boldsymbol{v}_z\| - a_\varphi \Delta\varphi_z \text{ , } \omega_{Dt} = a_v \|\boldsymbol{v}_z\| + a_\varphi \Delta\varphi_z , \tag{4}$$

where $\|\boldsymbol{v}_z\|$ and φ_z denote magnitude and angle of the velocity vector $\boldsymbol{v}_z$ given by (3), $\Delta\varphi_z$ approximates derivative of the φ_z, whereas weights a_v and a_φ are control parameters, which have to be adjusted experimentally and weight straight line and turning capabilities. First order difference is used as a derivative approximation $\Delta\varphi_z$:

$$\Delta\varphi_z = \begin{cases} \dfrac{\varphi_z - \varphi}{T_s}, & \|\boldsymbol{e}_z\| > d, \\ \dfrac{\varphi^* - \varphi}{T_s}, & \|\boldsymbol{e}_z\| \le d, \end{cases} \tag{5}$$

where T_s denotes the sampling time. It can be seen from (5) that controller tends to align orientations of the real and virtual robot when they are close enough, i.e., if their distance is less or equal d.

4.2 Obstacle avoiding controller (OAC)

Due to uncertainties and measurement noise, tracking is never perfect. Therefore, it is not still ensured that robot will pass from starting to destination point safely and additional controller which will be active in the close neighborhood of the obstacle becomes necessary. It should generate correctional control action which moves robot away from the obstacle. For this purpose, FLC is proposed with two inputs: distance between robot and obstacle, d_{TO}, and angle at which robot sees the obstacle, α_{TO}, and one output: normalized correction of the angular velocity, $\Delta\omega'_{OAC}$. If the position and radius of the circular obstacle are denoted by (x_O, y_O, r_O), inputs of the FLC can be evaluated using the following formula:

$$\begin{gathered} d_{TO} = \left\|(x,y),(x_O,y_O)\right\| - r_O - b\sqrt{2}, \alpha_{TO} = \arctan\frac{y_O - y}{x_O - x} - \varphi, \\ \Delta\omega_{OAC} = K_\Delta^{OAC}\Delta\omega'_{OAC}, \end{gathered} \tag{6}$$

where K_Δ^{OAC} denotes the controller gain. Membership functions of the OAC inputs and output are depicted on Fig. 3, whereas the fuzzy rule base is given by **Table 1.**

Table 1
Fuzzy rule base of the OAC.

		angle α_{TO}					
		Right Back	*Right*	*Right Front*	*Left Front*	*Left*	*Left Back*
distance d_{TO}	*Close*	Small Positive	Medium Positive	Large Positive	Large Negative	Medium Negative	Small Negative

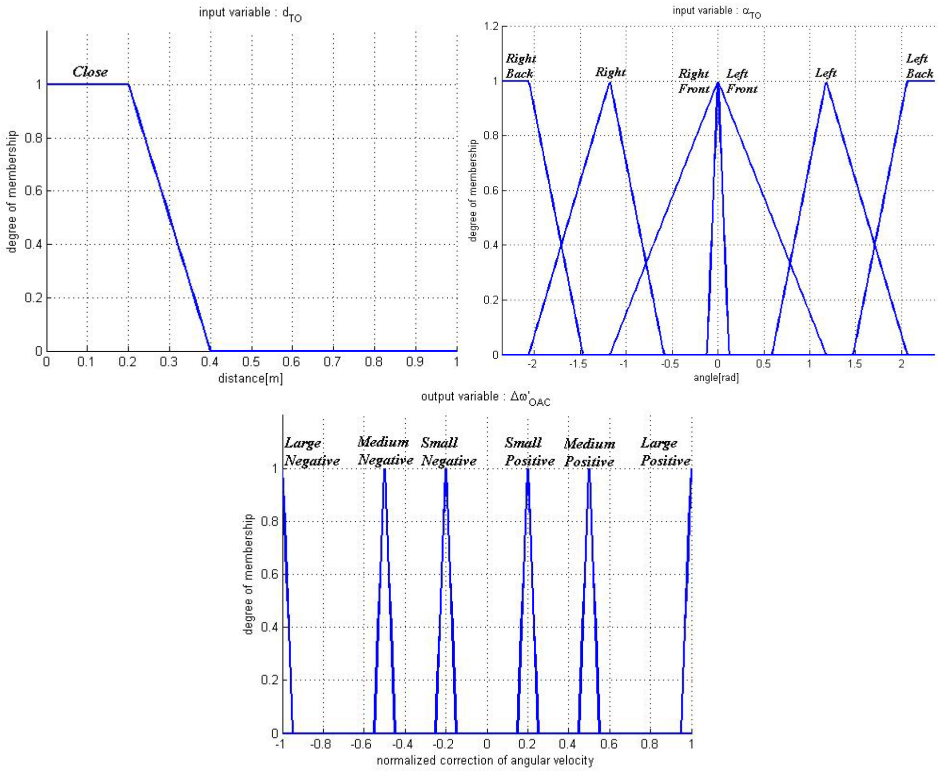

Fig. 3 – *Membership functions of the* OAC *input and output variables.*

4.3. Vehicle avoiding controller (VAC)

The purpose of the VAC is similar to the OAC, i.e., it becomes active when two robots are close enough to each other, generating correctional control action which moves robots away from each other, reducing the possibility of the collision. FLC is also used for this purpose with three inputs: distance to the closest robot, $d_{TV,}$ angle at which robot sees its neighbour, α_{TV}, and closing velocity, $v_{TV} = -\dot{d}_{TV}$, whereas the output is normalized correction of the angular velocity $\Delta\omega'_{VAC}$. If the (x_V, y_V) denotes current position of the closest robot, these inputs can be evaluated as:

$$d_{TV} = \left\|(x,y),(x_V,y_V)\right\| - 2b\sqrt{2}, \alpha_{TV} = \arctan\frac{y_V - y}{x_V - x} - \varphi,$$
$$\Delta\omega_{VAC} = K_{\Delta}^{VAC}\Delta\omega'_{VAC}, \tag{7}$$

where K_{Δ}^{VAC} denotes controller gain. Membership functions of the VAC inputs and output are depicted on Fig. 4, while the fuzzy rule base is given by **Table 2**.

Table 2
Fuzzy rule base of the VAC.

angle d$_{TV}$ = *Close*							
		angle α_{TV}					
		Right Back	*Right*	*Right Front*	*Left Front*	*Left*	*Left Back*
closing velocity v_{TV}	*Large*	Medium Positive	Large Positive	Large Positive	Large Negative	Medium Negative	Small Negative
	Medium	Small Positive	Medium Positive	Large Positive	Large Negative	Medium Negative	Small Negative

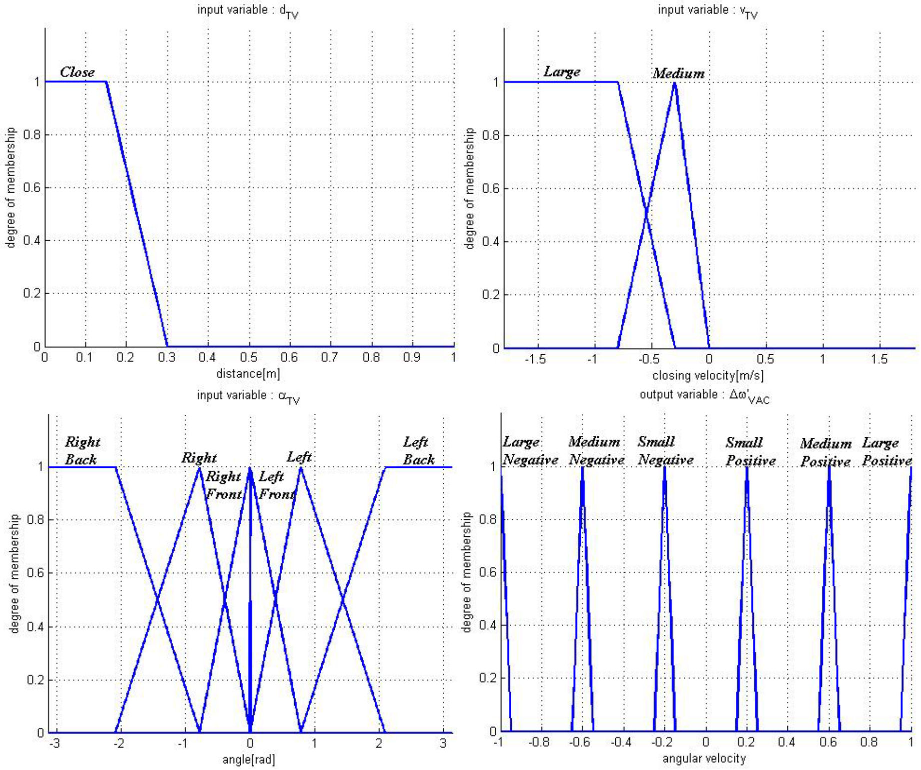

Fig. 4 – *Membership functions of the VAC input and output variables.*

4.4 Combined controller (CC)

Each of three previously presented controllers has a different function: TTC provides tracking of the reference trajectory, while OAC and VAC provide avoidance of objects in the environment, obstacles and vehicles, respectively.

The task of the CC is to combine control actions of these controllers into single control signal which is the input of the mobile robot, i.e. to mix these controls, depending on situation in the environment. When the robot is far away from the objects of the environment, tracking action should be dominant. When the robot approaches obstacle, contribution of the OAC gradually increases, while the contribution of the TTC decreases. Similar situation happens when robot approaches another vehicle, when the contribution of the VAC increases. Output of the CC can be written as a weighted sum, whose weights are variable and depend on distance between robot and objects of the environment (obstacles and other robots).

$$\begin{aligned} \omega_L &= K_1\omega_L^t - K_2\Delta\omega_{OAC} - K_3\Delta\omega_{VAC}, \\ \omega_D &= K_1\omega_D^t + K_2\Delta\omega_{OAC} + K_3\Delta\omega_{VAC}, \end{aligned} \tag{8}$$

where weights K_1, K_2 and K_3 can be evaluated using the following formula:

$$\begin{aligned} K_1 &= \min\left(\min\left(1, \frac{1-K_1^{\min}}{d_{kO}}d_{TO} + K_1^{\min}\right), \min\left(1, \frac{1-K_1^{\min}}{d_{kV}}d_{TV} + K_1^{\min}\right)\right), \\ K_2 &= \max\left(0, -\frac{K_2^{\max}}{d_{kO}}d_{TO} + K_2^{\max}\right), \\ K_3 &= \max\left(0, -\frac{K_3^{\max}}{d_{kV}}d_{TV} + K_3^{\max}\right), \end{aligned} \tag{9}$$

whereas $K_1^{\min}$ denotes minimal value of the weight K_1, $K_2^{\max}$ and $K_3^{\max}$ maximal values of the weights K_2 i K_3, while d_{kO} and d_{kV} represent minimal distances to obstacle and vehicle when OAC and VAC become active, respectively. Choice of these parameters is critical. It is recommended to choose $K_1^{\min}$ between 0.4 and 0.5, $K_2^{\max}$ and $K_3^{\max}$ should be between 0.4 and 0.6, while choice of parameters d_{kO} and d_{kV} depend on robot and obstacle size.

5 Simulation Results

Proposed solution to formation control in environment with known and static obstacles is simulated in MatLab package. Although scenario with circular obstacles is adopted, proposed solution can be applied in scenario with arbitrarily shaped obstacles. It is assumed that high level planner is available, which provides reference trajectory for the leader robot. One of the solutions is proposed in [8]. Trajectories of the followers are generated based on trajectory realized by the leader robot.

It is assumed that the dimensions of all robots are the same, i.e., $b = 15$ cm and $r = 6$ cm. Maximal angular velocities of the motors for the leader and followers are $\omega_L^{\max} = 15\ \text{rad/s}$ and $\omega_F^{\max} = 27\ \text{rad/s}$, respectively. The initial position of the virtual leader is (–1, 1), while its destination is (11, 3). Real leader starts its motion from the close neighborhood of the virtual one, i.e., initial conditions of the real leader are $(x_0^L, y_0^L, \varphi_0^L) = (-1.05, 1, \pi/4)$. In order to be applicable in real world conditions, control algorithm must provide dynamic formation switching if the collision with obstacles may occur. Herein, simple algorithm is adopted, i.e. formation has to switch to convoy when any of the virtual robots collide with obstacle.

Triangular formation is adopted, whose parameters are $P = \begin{bmatrix} p_1 & p_2 \end{bmatrix} = \begin{bmatrix} 0.8 & 0.8 \end{bmatrix}\mathrm{m}$ and $Q = \begin{bmatrix} q_1 & q_2 \end{bmatrix} = \begin{bmatrix} -0.4 & 0.4 \end{bmatrix}\mathrm{m}$. Initial positions of the real followers can be arbitrarily chosen, i.e., they are $(x_0^{F_1}, y_0^{F_1}, \varphi_0^{F_1}) = (-2, 0, 0)$ and $(x_0^{F_2}, y_0^{F_2}, \varphi_0^{F_2}) = (-2, 2, 2\pi/3)$, for the first and second follower, respectively. Starting points of the followers are chosen such that they can collide during early stages of formation establishing.

Parameters of the controllers are adjusted experimentally, as a compromise between tracking performance and object avoiding capability. The parameters of the control structure with PI TTC for the leader robot are chosen as follows:

$$\begin{gathered}
\mathrm{TTC}: \quad K_p = 5, \quad T_i = 0.5, \quad k = 10, \quad d_0 = 0.05\mathrm{m}, \\
a_v = 66.68, \quad a_\varphi = 9.33, \\
\mathrm{OAC}: \quad K_\Delta^{OAC} = 7.5, \\
\mathrm{CC}: \quad K_1^{\min} = 0.5, \quad K_2^{\max} = 0.7, \quad d_{kO} = 0.6\mathrm{m},
\end{gathered} \tag{10}$$

while the parameters of the followers' controllers are the same for both robots and chosen as:

$$\mathrm{TTC}: \quad K_p = \begin{cases} \dfrac{-\left(K_p^{\max} - K_p^{\min}\right) d_{\min} - K_p^{\min} d_{c1} + K_p^{\max} d_{c2}}{d_{c2} - d_{c1}}, & d_{c1} \le d_{\min} \le d_{c2}, \\ K_p^{\max}, & d_{\min} \le d_{c1}, \\ K_p^{\min}, & d_{\min} \ge d_{c2} \end{cases}$$

$$\begin{gathered}
d_{\min} = \min\left(d_{TO}, d_{TV}\right), \quad d_{c1} = 0{,}225, \quad d_{c2} = 0.6, \quad K_p^{\min} = 2, \quad K_p^{\max} = 8, \\
\mathrm{OAC}: \quad K_\Delta^{OAC} = 21, \\
\mathrm{VAC}: \quad K_\Delta^{VAC} = 27, \\
\mathrm{CC}: \quad K_1^{\min} = 0.5, \quad K_2^{\max} = 0.8, \quad K_3^{\max} = 0.8, \quad d_{kO} = 0.4\mathrm{m}, \quad d_{kV} = 0.6\mathrm{m}.
\end{gathered} \tag{11}$$

It can be seen from (11) that proportional gain of the follower's PI depends on vehicle to object distance $d_{\min}$, i.e., it decreases when distance to closest object increases. It is experimentally observed that this variable gain is better solution than the constant one, i.e. avoiding capability is improved. Proportional gain K_p, as well as gains a_v and a_φ have major effect on tracking performance. Increase of these parameters (as well as $K_1^{\min}$) improves tracking performance, but degrades avoiding capability. Avoiding capability can be improved by increasing the parameters K_Δ^{OAC} and K_Δ^{VAC}, as well as $K_2^{\max}$ and $K_3^{\max}$. It is the rule for all controllers, i.e. tracking performance can be improved by increasing the TTC gains, whereas avoiding capabilities can be improved by increasing the gains that belong to VAC and OAC. These two goals are opposite, i.e. improvement of tracking quality usually leads to degradation of avoiding capability and vice versa. Hence, chosen values of parameters are compromise between these two opposite goals.

Figs. 5, 6 and 7 show relevant variables when the complete control structure is adopted. State variables of the robots in formation (COM position and orientation of the robot), together with tracking errors are shown on Fig. 5, control signals are shown on Fig. 6, while the two-dimensional plot of the formation motion is given on Fig. 7, where snapshots are taken on every 5s.

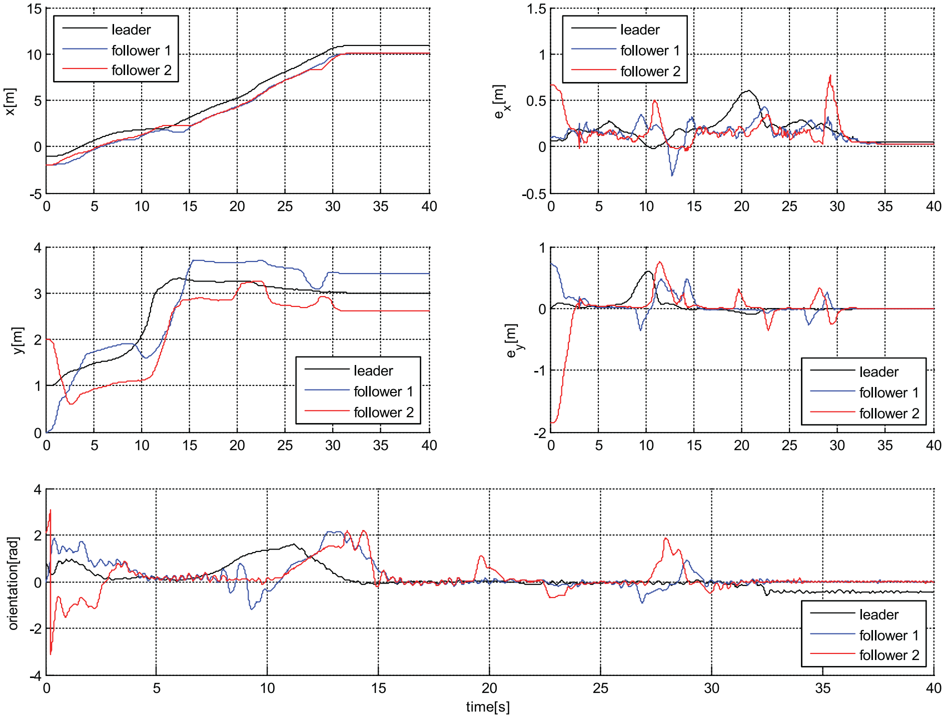

Fig. 5 – *Trajectory tracking performance.*

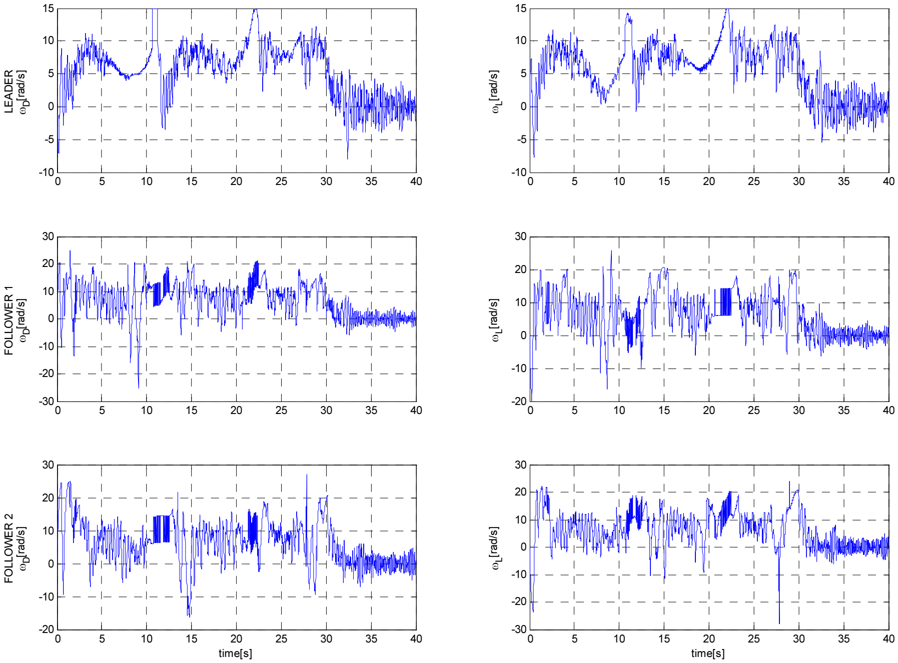

Fig. 6 – *Control signals of the leader and followers.*

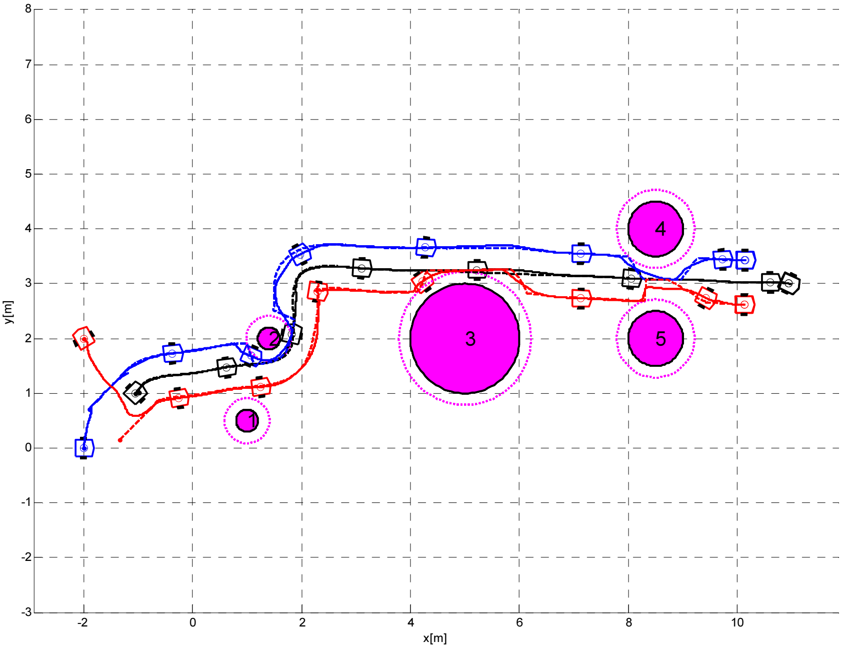

Fig. 7 – *Two-dimensional view of robots' motion.*

It can be observed that motion of all robots in the formation is slightly oscillatory, as can be seen from orientation plot on Fig. 5. Filled pink circles depict obstacles, while dashed pink lines represent borders of the obstacles, enlarged by the robot dimension. Hence, paths realized by robots have to be outside regions enclosed by these lines. Leader is depicted by black color, while the first and the second follower are depicted by blue and red color, respectively. Dashed lines represent desired paths, while the solid lines represent paths realized by the real robots. The color of the path corresponds to the color of the robot. Tracking performance is satisfactory, as well as collision avoidance with vehicles. Unfortunately, obstacle avoidance can be unsatisfactory in some cases, when sharp maneuver is required. It can be seen from Fig. 7 that the first follower slightly touches obstacle 2. Formation switching is executed successively.

6 Conclusion

An approach for formation control in known and static environment is presented in this paper. It is assumed that high level planner is available, which provides reference trajectory for the leader robot, while the desired trajectories of the followers are generated based on trajectory realized by the leader. Formation switching is also provided, in cases when formation cannot be maintained. In order to make manoeuvres in space with obstacles successively, every robot in formation is equipped with three different controllers: trajectory tracking, obstacle avoiding and vehicle avoiding controller, whose actions are coordinated, depending on situation in the environment. Simulation results show the efficiency of the proposed approach. Proposed approach can be improved further, using online path planning algorithm, when proposed control structure can be suitable for real world, dynamic environments.

7 Acknowledgement

This paper is supported by Serbian Ministry of Science under the grants TR-35003 and III-44008 and particularly by the project “CARE-robotics” supported by SNSF SCOPES IP program under the grant IZ74Z0_137361/1.

8 References

[1] R.W. Beard, J. Lawton, F.Y. Hadaegh: A Coordination Architecture for Spacecraft Formation Control, IEEE Transactions on Control Systems Technology, Vol. 9, No. 6, Nov. 2001, pp. 777 – 790.

[2] T. Balch, R.C. Arkin: Behavior-based Formation Control for Multirobot Teams, IEEE Transaction on Robotic Automation, Vol. 14, No. 6, Dec. 1998, pp. 926 – 939.

[3] J. Huang, S.M. Farritor, A. Qadi, S. Goddard: Localization and Follow-the-leader Control of a Heterogeneous Group of Mobile Robots, IEEE/ASME Transactions on Mechatronics, Vol. 11, No. 2, April 2006, pp. 205 – 215.

[4] W. Ren, R.W. Beard: Formation Feedback Control for Multiple Spacecraft via Virtual Structures, IEE Proceedings – Control Theory Application, Vol. 151, No. 3, May 2004, pp. 357 – 368.

[5] S. Spry, J.K. Hedrick, J.K: Formation Control using Generalized Coordinates, 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas, 14 – 17 Dec. 2004, Vol. 3, pp. 2441 – 2446.

[6] F.E. Schneider, D. Wildermuth: A Potential Field based Approach to Multi Robot Formation Navigation, IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, Changsha, Hunan, China, 08 – 13 Oct. 2003, Vol. 1, pp. 680 – 685.

[7] M. Šušić, A. Ćosić, A. Ribić, D. Katić: An Approach for Intelligent Mobile Robot Motion Planning and Trajectory Tracking in Structured Static Environments, IEEE 9th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, 08 – 10 Sept. 2011, pp. 17 – 22.

[8] A. Ćosić, M. Šušić, D. Katić: Advanced Algorithms for Mobile Robot Motion Planning and Tracking in Structured Static Environments using Particle Swarm Optimization, Serbian Journal of Electrical Engineering, Vol. 9, No. 1, Feb. 2012, pp. 9 – 28.

[9] T.D. Barfoot, C.M. Clark: Motion Planning for Formations of Mobile Robots, Robotics and Autonomous Systems, Vol. 46, No. 2, Feb. 2004, pp. 65 – 78.