

Formation Control of UGVs using an UAV as Remote Vision Sensor [★]

Andres Hernandez ^{*}, Cosmin Copot ^{*}, Juan Cerquera ^{*,**},
Harold Murcia ^{*,**} and Robin De Keyser ^{*}

^{*} Department of Electrical energy, Systems and Automation, Ghent University, Belgium (Andres.Hernandez,Cosmin.Copot}@UGent.be).

^{**} Department of Automation and Industrial Control, D+TEC research group, Universidad de Ibagué, Colombia.

Abstract: A leader-follower formation control scheme based on SRV-1 UGVs and an AR.DRONE 2.0 UAV as remote vision sensor is presented in this paper. The main advantage of the proposed strategy lies on the flexibility obtained from a flight remote sensor, as it makes possible to locate the agents at larger distances between them or to extend more easily the number of agents in the formation. A full description of the internal control designed for the UGVs and the UAV is presented, including the image processing procedure implemented to robustly measure the pose of the vehicles in the formation. Finally, experimental results using a triangular formation of three ground robots illustrates the effectiveness of the proposed control scheme.

Keywords: Autonomous vehicles, Closed-loop control, computer vision, flight control

1. INTRODUCTION

In recent years, automatic control has been incorporated into all civil vehicles: on roads, on railways, in air and space and their subsequent infrastructure (Goerzen et al. (2010); Xu et al. (2010)). The current trends indicate a preference for developing more complicated and effective control systems with diagnostic functionalities, able to improve travelers safety and comfort, the overall performance of the transportation system and the interaction between vehicles with their surroundings.

One direction of research extended from mobile robots in the last ten years is multi-robot systems (MRS). One of the first implementations of such a system was done in the Swarm-bots project and resembled to an insect colony (Bonabeau et al. (1999)). Each robot in the formation is an autonomous system, specialized on developing certain tasks, enhancing the flexibility of the overall system and allowing easy adaptation to different specifications.

The application of MRS to real-world scenarios requires the consideration of many challenging details that may increase the complexity of the implementation step. The ability to interact with a dynamic, changing environment is of key importance. Robots must be able to handle various unexpected events that can disrupt the formation, thus requiring obstacle avoidance, formation repair and changes in the formation, for which higher levels of decision making become vital. Techniques from artificial intelligence that allow identification of strategies and tactics may be needed (Murray (2007)). Furthermore, robots themselves must satisfy dynamic constraints, such as velocity and acceleration bounds. Recent experimental results provide verification of formation control and show its usefulness in real-world applications (Michael et al. (2007)).

An efficient method for deployment a formation control of mobile robots was proposed in (Chiem and Cervera (2004)). In order to estimate the position and orientation of each robot relative to its leader, a color-tracking vision system is used. In (Vidal et al. (2003)), the formation control problem was expounded as separated visual tracking tasks for each follower robot. The follower uses visual information acquired with an omni-directional camera to estimate the position and the velocity of the leader. Formation tracking is the largest portion of formation control research. The goal of formation tracking is that a group of robots has to maintain a desired formation, while tracking or following a reference. This task may also include path planning, trajectory generation and motion feasibility for robot formation (Tabuada et al. (2005)).

The study of formation control emerged from the development of MRS. Formation control classification is based on the type of strategy: behavior-based, virtual-structure and leader-follower. Each of these strategies has its own advantages and disadvantages. The type of formation used in our paper is a leader-follower formation (Das et al. (2002)). The leader-follower formation can have different forms, such as a line, i.e. where each robot is a leader for the following robot. Alternatively, we can have a *triangular* formation where a single leader robot is followed by two followers.

In this paper, a leader-follower formation of UGV robots using an UAV as remote vision sensor is presented. The type of UGVs used in this research are the Surveyor SRV-1 differential tracked robots with tank-like structure. A low-cost commercial AR.Drone quadrotor and three UGVs were considered in experimental evaluation. The bottom camera of the AR.Drone is used as sensor for visual coordinating of the UAV-UGVs in the formation. Different patterns and an image processing algorithm is used in order to distinguish the robots in the formation. The pose (position and orientation) of the mobile agents in the formation is also computed based on visual information.

[★] This work has been obtained as results of the bilateral agreement between Ghent University and Universidad de Ibagué, in the framework of the Master in Control Engineering

A communication system using Wi-Fi network is well set-up in order to communicate and control all the mobile agents in real-time.

The paper is organized as follows: Section 2 presents the robot formation control architecture. Section 3 presents the plant description and the control approach of the UGV robots. Section 4 provides the description of the quadrotor and gives our approach to the control problem. Section 5 shows the experimental results and a conclusion section summarizes the main outcome of this work.

2. THE ROBOT FORMATION CONTROL

Formation tracking is the largest portion of formation control research. The goal of formation tracking is that a group of robots maintain a desired formation, while tracking or following a reference. The main advantage of using the formation control in path following is that the entire formation will slow down if the robots get out of formation and it moves towards its goal if the robots are maintaining formation. However, practical constraints arise from the characteristics of the supporting inter-robot communication network. They have to exchange the dynamic variables to each other via communication. Thus, the quality of communication channels becomes a crucial part.

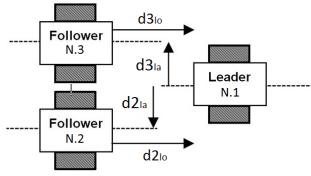


Fig. 1. Triangular leader-follower formation

In this study a *triangular* leader-follower formation is considered. On previous studies (Neamtu et al. (2011), Cristescu et al. (2012) and Chevalier et al. (2013)) the formation was successfully maintained by controlling the longitudinal d_{lo} and lateral d_{la} distances between the UGVs as depicted in figure 1. The distances are computed using the information obtained from the on-board camera located in the front of each one of the followers. Consequently, the lateral distance between the leader and the followers (e.g. d_{2la} and d_{3la}) depicted in Fig. 1 will be restricted by the size of the image obtained with the on-board camera. If the follower moves away and is not able to recognize the pattern from the robot in front, it will be impossible to keep the formation.



Fig. 2. Proposed formation control scheme using UGV SRV-1 robots and one UAV as remote vision sensor.

In this contribution the novelty is to include an UAV as remote vision sensor in the formation configuration, as depicted in Fig. 2. The UAV fully described in section 4, holds a camera mounted on the bottom of the hull, making possible to fly over the formation to obtain the lateral and longitudinal distances between the UGVs. Under this situation more flexibility is obtained as it is possible to fly high enough to see the complete

formation, or to move around in order to extend the number of UGVs in the formation.

3. UGV: THE SRV-1 ROBOTS

3.1 Plant Description

Our set up works with Surveyor SRV-1 mobile robots produced by the Surveyor Corporation. Surveyor SRV-1 is a differential tracked mobile robot with tank-like treads. The treads are rotated by 4 DC motors, two on each side. Between the shaft of the motor and the wheel that moves the tread there is a gearbox with reduction ratio of 100:1.

Different characteristics of the motors, ground conditions and status of the battery will cause the left track and the right track to rotate at different speeds in open loop, resulting in curved directions. For this reason we decided to mount on the robot optical encoders, to enable speed measurement and to control the lateral direction. The range of the speed of the robot is between 10 cm/s - 50 cm/s, it has a nonlinear characteristic with a dead-zone in the beginning due to the Coulomb friction, representing an additional challenge to the control objective.

The kinematic equations apply for this type of robots are given by:

$$v = \frac{v_l + v_r}{2}, \omega = \frac{v_l - v_r}{2l_A} \quad (1)$$

where v denotes the velocity of the robot, ω is the angular velocity, v_r and v_l represents the velocity on the right and left side of the robot respectively, and l_A is the distance between the left and the right track.

The advantage of the SRV-1 robot is that it can operate as remotely-controlled robot via Wi-Fi connection. Moreover, the SRV-1 is fully programmable for autonomous operation and can run onboard interpreted C sub-program or user-modified firmware. From the formation point-of-view, one of the challenges is the lack of communication between robots. The robots communicate with a host computer using TCP/IP protocol via wireless network.

Since the robot is a simple process, based on DC motors, a first order transfer function can be used to represent the robot's dynamics. The identification of the robot is done based on the step response, resulting in the following transfer function between the reference speed and the output speed:

$$H(s) = \frac{1.24}{0.1s + 1} \quad (2)$$

3.2 The Control Approach

The control approach consists of two distinct levels: an upper level, that controls the direction (lateral and longitudinal) of the robot based on the data extracted from image processing, and a lower level inner loop, that controls the rotation speed of the tracks of the robot. For the longitudinal and lateral control, we need to compute the distance between two robots and the offset of the follower with respect to the center of the leader. This data will be later obtained as result of the image processing of the information provided by the quadrotor section 4.2.

Speed control

The speed control compensates for the bias that causes the robot to deviate from a straight path and provides information about

the actual longitudinal speed and angular velocity of the robot. The control loop for both left and right tracks correspond to a proportional controller with feedforward action. In this system the feedback information is supplied by optical encoders placed in the wheels of the robot. As the signal from the optical encoder is noisy, we apply an efficient moving average filter. The feedforward is used to prevent a negative signal and get a faster response. To compute the feedforward, we determine the static characteristic from voltage to speed (Cristescu et al. (2012)).

Directional control

To control the direction of the robot, we combine a lateral and longitudinal controller. A common approach to decouple the lateral and longitudinal movement is adding the lateral controller to one side and subtracting it from the other side (Neamtu et al. (2011)). To compute an accurate controller for the longitudinal movement, we need a model and an identification of the model parameters. For this purpose, we need an input signal that should not be affected by the signal given by the lateral controller. Therefore, we decide to add the compensation for the lateral movement only on one side (i.e. the right side).

The lateral and longitudinal controllers are PI controllers with an anti-windup strategy to make the robot go straight, while avoiding steady-state error. Tuning of the PI controllers is done using computer-aided design, namely the Frequency Response Toolbox (FRTool) (De Keyser and Ionescu (2006)). The transfer function of the longitudinal C_{lo} and lateral C_{la} controllers are given by:

$$\begin{aligned} C_{lo}(s) &= 3.4 \left(1 + \frac{1}{1.28s} \right) \\ C_{la}(s) &= 0.4 \left(1 + \frac{1}{8s} \right) \end{aligned} \quad (3)$$

Complete control scheme

To complete the control, the velocity of the robot is computed as the average velocity of the left and right wheels and the angular velocity of the robot is computed as the difference between the left and right track velocity, divided by ℓ_A , which is the distance between the left and right wheels. The schematic representation of the complete control algorithm is shown in figure 3.

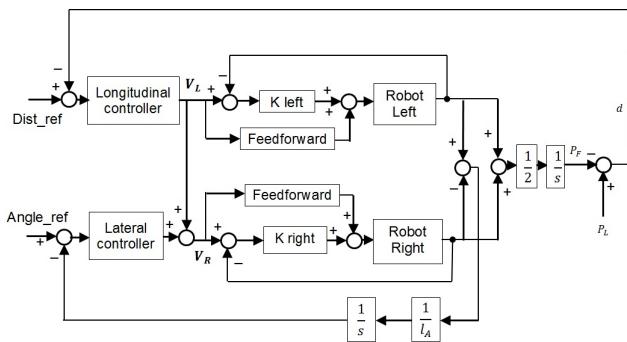


Fig. 3. Schematic representation of the control strategy embedded on each UGV.

4. UAV: THE AR.DRONE 2.0 QUADROTOR

4.1 Plant Description

The Ar.Drone 2.0 is a commercial and low-cost micro Unmanned Aerial Vehicle. The quadrotor comes with internal in-flight controllers and emergency features making it stable and safe to fly (Bristeau et al. (2011)). The only downside would be that access to the internal controller of the quadrotor is restricted. The internal software is black-box and the parameters that refer to control, motors and other calibrations are undocumented. There are 4 brushless DC motors powered with 14.5 W each from the 3 element 1000 mAh LiPo rechargeable battery which gives an approximate flight autonomy of 10-15 minutes. Two video cameras are mounted on the central hull. The front camera resolution is 1280x720 and the bottom one is 640x360 with a video stream rate of 30 FPS and 60 FPS for front and bottom cameras.

The sensors are located below the central hull and consist of a 3-axis accelerometer, a 2-axis gyroscope and 1-axis gyroscope which together form the Inertial Measurement Unit (IMU). There is one ultrasonic sensor and one pressure sensor for altitude estimation. A 3-axis magnetometer gives the orientation of the quad-rotor with respect to the command station. Communication between quadrotor and command station is done via Wi-Fi connection within a range of 30 m to 100 m for indoor and outdoor environment, respectively. The AR.Drone creates a Wi-Fi network, self-allocates a free IP address to grant access to client devices that wish to connect. For more details about internal structure of this quadrotor, check (Bristeau et al. (2011)).

A Visual Studio application in C++ establishes access to all AR.Drone communication channels, enabling functions to send commands or set configurations and also receive and store data from sensors and video stream. Thus, data can be interpreted off- or on-line for the purpose of identification, modeling and control of the quadrotor.

The quadrotor aerial movements are similar to those of a conventional helicopter. The difference is that movement is achieved by varying each of the motor speeds to obtain the desired effect. Figure 4 depicts the movement axes of the quadrotor. The six Degrees Of Freedom (DOF) of the AR.Drone give attitude and position. Movements are thus achieved on:

- Pitch - By rotational movement along transversal axis y, translational movement on x axis is made.
- Roll - By rotational movement along longitudinal axis x, translational movement on y axis is made.
- Yaw - Rotational movement along z axis.
- Throttle - Translational movement on z axis.

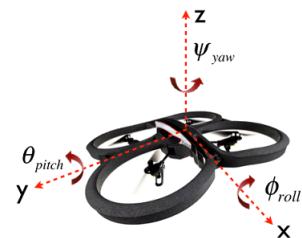


Fig. 4. Quadrotor Movement Axes

Movement is achieved by giving reference values as input to the internal, black-box controllers. The input and output relations will be discussed in the following subsection.

The control parameters given to the internal controllers are floating point values between [-2.5, 2.5] and represent the percentage of the minimum or maximum configured value for the respective movement. We denote $\{\phi, \theta, \zeta, \psi\}$ the roll angle reference value, pitch angle reference, vertical speed reference and yaw angular speed reference. We denote them by $\{\phi_{out}, \theta_{out}, \psi_{out}, \zeta_{out}, \dot{x}_{out}, \dot{y}_{out}\}$ symbolizing roll, pitch and yaw angle in radians, altitude in meters and linear velocities on longitudinal and transversal axes in m/s.

The quadrotor behaves as a Linear Time-Invariant System and parametric identification is done over pitch, roll, throttle and yaw movements using the prediction error method (Ljung (2007)). A Pseudo-Random Binary Signal (PRBS) is used to identify the dynamics of the quadrotor. A sampling time of 33 ms is chosen based on the analysis of dynamics performed on previous work Vlas (2013). The obtained transfer functions are given by:

$$\begin{aligned} H_{pitch}(s) &= \frac{x_{out}(s)}{\theta(s)} = \frac{3.65}{s(1.9s+1)} e^{-0.33s} \\ H_{roll}(s) &= \frac{y_{out}(s)}{\phi(s)} = \frac{2.67}{s(1.6s+1)} e^{-0.33s} \\ H_{throttle}(s) &= \frac{\zeta_{out}(s)}{\zeta(s)} = \frac{3.21}{s(s^2+2.67s+11.26)} e^{-0.33s} \\ H_{yaw}(s) &= \frac{\psi_{out}}{\psi(s)} = \frac{35.1}{s(0.15s+1)} e^{-0.33s} \end{aligned} \quad (4)$$

The observed time delay is due to the time it takes to receive the image from the quadrotor to the command station.

4.2 Image Processing

An important step for control is the image processing as the robots and its position are obtained based on the image captured with the bottom camera of the quadrotor.

Pattern recognition

In Fig. 5(a) is depicted the image obtained by the bottom camera of the Quadrotor with a resolution of 640x360px. The image is updated every 66ms and suffers of a communication delay of 330 ms. The image is then filtered to make it softer, and to remove noise from the patterns. Subsequently, to avoid working with color segmentation the image is converted to grayscale.

An important aspect related to the image processing is robustness, i.e., the ability to recognize the different patterns disregarding changing lighting conditions. To this end, a dynamic threshold was computed based on the histogram of the image as depicted in Fig. 5(b). A relevant aspect is to properly choose the threshold value, this is done by selecting a value in between the two peaks representing the dark and clear parts of the image, in order to obtain a black and white image (Fig. 5(c)). The resulting is a binary image of ones and zeros.

The next step consists in using contour tracing to more accurately classify the different patterns. It consists in delineate the black and white image with an ordered sequence of points distanced by one pixel. Subsequently, the contours become polygonal approximations by identifying only the squares using an exclusion analysis (for example the number of polygon sides), as depicted

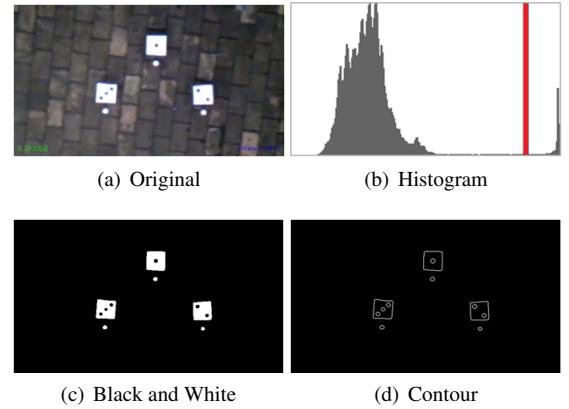


Fig. 5. Image processing procedure

in Fig. 5(d). In order to increase the robustness of the method, two procedures are followed: first, the longitude of the sides are compared, the difference between them cannot be larger than 3 cm or equivalently (20%); the second exclusion procedure is performed based on the computation of the area. It consists in neglecting the squares with an area smaller than 200cm^2 or bigger than 250cm^2 , knowing that the real area of each square is 225cm^2 . Subsequently, the number of contours within each square are counted to recognize the different robots (e.g. robot 1, robot 2, etc).

Computation of longitudinal and lateral distances

Once the robots are identified, the angle and distance between robots is computed. The procedure consists of 5 steps. First, the center of the image (CI) is computed, this basically represents the position of the quadrotor above the robots. Note that the real position of the quadrotor respectively to the ground vehicles must be corrected depending on the current pitch angle ϕ . The second step consists in calculating the center of the square that represents the ground vehicle to be followed (CR).

Next in the third step, the error between the position of the quadrotor (CI) and the position of the ground robot (CR) is calculated and converted from pixels to meters. Considering that the center of the ground vehicle has a coordinate $P_{ci} = (x_{ci}, y_{ci})$ and the orientation point $P_{oi} = (x_{oi}, y_{oi})$, where the sub-indexes c refers to center and o to orientation of the i-ground robot. The conversion factor from pixels to meters is easily found because the resolution of the camera and the real distance between P_{ci} and P_{oi} are known, for a given altitude Fig. 6(a).

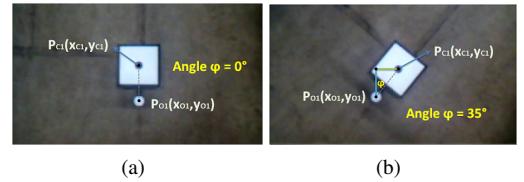


Fig. 6. Orientation of the quadrotor with respect to the ground vehicle

In the fourth step, the orientation ϕ of the ground robot with respect to the quadrotor is computed (Fig. 6(b)) as:

$$\phi = \tan^{-1} \left(\frac{x_{c1} - x_{o1}}{-(y_{c1} - y_{o1})} \right) \quad (5)$$

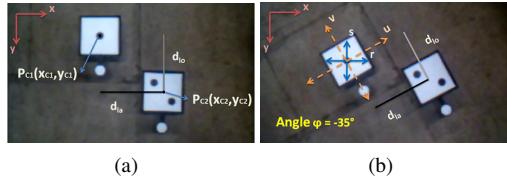


Fig. 7. Calculation of lateral and longitudinal distances

The final step consists in computing the lateral and longitudinal distances which will be later given to the controller on-board the UGVs as described on section 3.2. In the case that the formation is aligned with respect to the orientation of the quadrotor as in Fig. 7(a), the computation of the lateral distance d_{la} and longitudinal distance d_{lo} of the followers can be obtained directly from the points representing center of the robots $P_{ci} = (x_{ci}, y_{ci})$ for $i = 1, 2, \dots$ representing the subindex for each UGV.

$$\begin{aligned} d_{lo} &= y_{c2} - y_{c1} \\ d_{la} &= x_{c2} - x_{c1} \end{aligned} \quad (6)$$

However, for the case when the formation is not aligned with respect to the quadrotor orientation as depicted in Fig. 7(b), a change of coordinates is required in order to correctly compute the distances. First, a new cartesian plane (r, s) is created with coordinate $(0, 0)$ in the center of the first robot, leading to the following new P_1 and P_2 points:

$$\begin{aligned} P_{c1}(r, s) &= (0, 0) \\ P_{c2}(r, s) &= ((x_{c2} - x_{c1}), -(y_{c2} - y_{c1})) \end{aligned} \quad (7)$$

Subsequently, $P_1(r, s)$ and $P_2(r, s)$ in (7) are transformed into polar coordinates in the plane (r, s) :

$$P_{c2}(r, s) = \left(\sqrt{\Delta X^2 + \Delta Y^2}, \tan^{-1} \left(\frac{-\Delta Y}{\Delta X} \right) \right) \quad (8)$$

where $\Delta X = x_{c2} - x_{c1}$ and $\Delta Y = y_{c2} - y_{c1}$. Then $P_2(r, s)$ is rotated by φ and mapped to the plane (u, v) :

$$\begin{aligned} P_{c1}(u, v) &= (0, 0) \\ P_{c2}(u, v) &= \left(\sqrt{\Delta X^2 + \Delta Y^2}, \tan^{-1} \left(\frac{-\Delta Y}{\Delta X} + \varphi \right) \right) \end{aligned} \quad (9)$$

Finally, the lateral and longitudinal distances d_{la} and d_{lo} are obtained from (9):

$$\begin{aligned} d_{la} &= \sqrt{\Delta X^2 + \Delta Y^2} \cos \left(\tan^{-1} \left(\frac{-\Delta Y}{\Delta X} + \varphi \right) \right) \\ d_{lo} &= \sqrt{\Delta X^2 + \Delta Y^2} \cos \left(\tan^{-1} \left(\frac{-\Delta Y}{\Delta X} \right) \right) \end{aligned} \quad (10)$$

The final result of the image processing is presented in Fig. 8, which includes the following information: the calculation of distances, the simulation time (on the bottom-left), the altitude obtained from the ultrasonic sensor and the one based on the image (at the bottom-right), the leader robot being followed (at top-left), the orientation (φ in degree) of each robot with respect to the quadrotor and the lateral and longitudinal distances are represented by the black lines with origin in the center of each follower robot.

4.3 Position Control

In order the quadrotor to follow the formation, a position control was designed based on the identified dynamics (4). The tuning of the different controllers is done using computer-aided

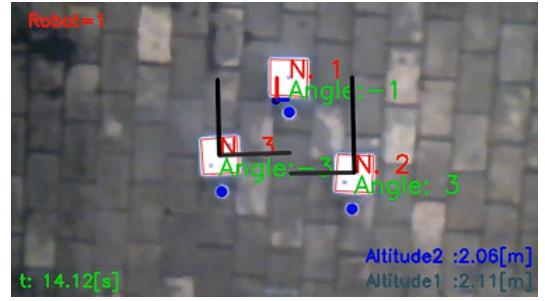


Fig. 8. Screenshot of the result obtained after the image processing

design, namely the Frequency Response Toolbox (FRTool) De Keyser and Ionescu (2006). The following controllers were designed:

$$\begin{aligned} C_{pitch}(s) &= 0.2317(2.05s + 1) \\ C_{roll}(s) &= 0.315(1.495s + 1) \\ C_{throttle}(s) &= 1.63(0.33s + 1) \\ C_{yaw}(s) &= 0.01 \end{aligned} \quad (11)$$

Due to the fact that the system (4) already includes an integrator, P and PD controllers were chosen to drive the quadrotor. Moreover the main objective is not to precisely follow the setpoint but rather to see all the ground vehicles in one image.

5. EXPERIMENTAL RESULTS

In this section the results obtained with the proposed formation control configuration are presented. The experiment consists in keeping a triangular formation of three robots, one leader and two followers. As mentioned on section 2, the main advantage of using an UAV as remote vision sensor is the ability to give a larger setpoint for the lateral distance of the followers compared to use the on-board camera available on the UGV.

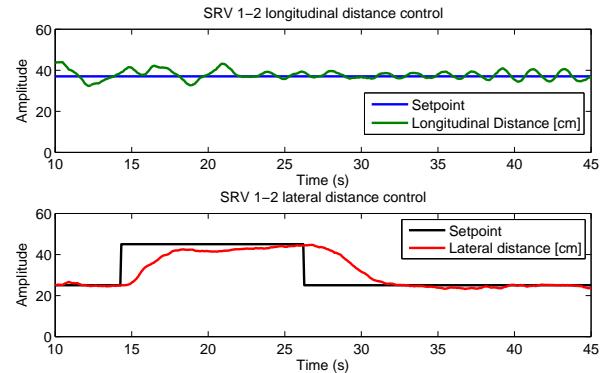


Fig. 9. Results of formation control for robot couple 1-2

The performance obtained for the triangular formation is depicted in Fig. 9 and Fig. 10. It is possible to observe that the longitudinal and lateral distances are controlled after a big change in the setpoint (± 20 cm at time 14s and 26s for couple 1-2 and at time 22s and 34s for couple 1-3) without steady-state error. The robustness of the controller designed is also tested during the experiment, as the same controller (i.e. longitudinal and lateral) is used for all the robots, despite the robots present quite different dynamics which moreover change on time due to errors in the encoder, bad contacts or wear on the gear-box.

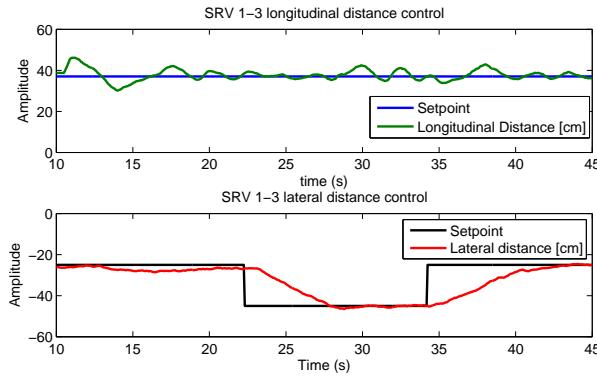


Fig. 10. Results of formation control for robot couple 1-3

The performance of the position control for the quadrotor is depicted in Fig. 11. The quadrotor follows the UGV-leader precise and smoothly. The altitude of the quadrotor is 2 m Fig. 11(a) and because the formation remained on a straight formation the orientation was always zero Fig. 11(b). Finally, the setpoint for the X- and Y-movements of the quadrotor is zero, because it is considered as the error between the UAV and the UGV-leader.

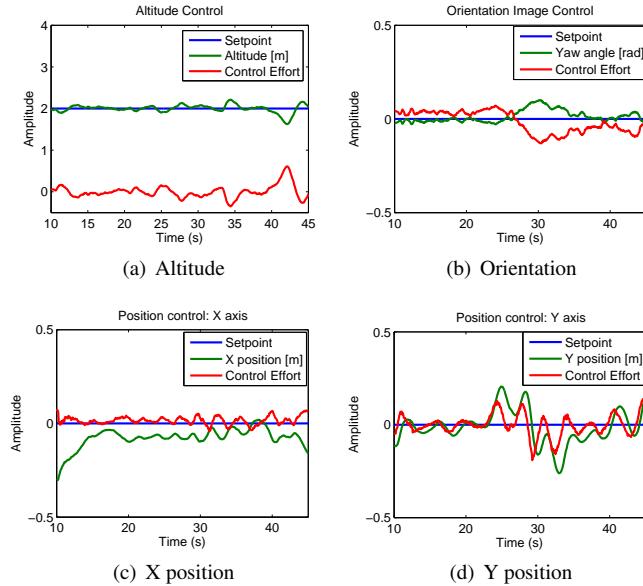


Fig. 11. Position control of the quadrotor during formation control

6. CONCLUSION

We have presented in this work a leader-follower formation control based on UGVs, from which the sensor was an UAV acting as remote vision sensor. A complete description of the control scheme used for the UGVs to follow the leader and for the UAV to provide accurate information based on image processing was presented. Finally experimental results demonstrate the feasibility of the proposed approach, which given its flexibility open new possibilities for formation control schemes.

Future work include introducing more UGV in the formation and to try different formation architectures (e.g. linear formation); which are possible extensions once the main framework presented in this paper was built. Other interesting extension

is in the field of cooperative and distributed control once bi-directional communication between the ground robots is implemented.

REFERENCES

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York.

Bristeau, P., Caliou, F., Vissiere, D., and Petit, N. (2011). The navigation and control technology inside the ar.drone micro uav. In *18th IFAC World Congress*, 1477–1484. Milano.

Chevalier, A., Copot, C., Cristescu, S., Ionescu, C., and De Keyser, R. (2013). Emulation of a highway bottleneck using leader-follower formation control. In *IEEE 8th International Symposium on Applied Computational Intelligence and Informatics*, 131–136.

Chiem, S. and Cervera, E. (2004). Vision-based robot formations with b閦ier trajectories. In *Intelligent Autonomous Systems 8*, 191–198. IOS Press.

Cristescu, S., Ionescu, C., Wyns, B., and De Keyser, R. (2012). Leader-follower string formation using cascade control for mobile robots. In *20th Mediterranean Conference on Control and Automation*, 1092–1098.

Das, A., Fierro, R., Kumar, V., Ostrowski, J., Spletzer, J., and Taylor, C. (2002). A vision-based formation control framework. *Trans. on Robotics and Automation*, 18, 813–825.

De Keyser, R. and Ionescu, C. (2006). A frequency response tool for cacsd in matlab. In *IEEE International Symposium on Computer Aided Control Systems Design*, 2275–2280. Munich.

Goerzen, C., Z.Kong, and B.Mettler (2010). A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent & Robotic Systems*, 57 (1-4), 65–100.

Ljung, L. (2007). *System identification: theory for the user*. Prentice-Hall.

Michael, N., Fink, J., and Kumar, V. (2007). Controlling a team of ground robots via an aerial robot. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 965–970. San Diego.

Murray, R. (2007). Recent research in cooperative-control of multivehicle systems. *Journal of Dynamics, Systems, Measurement and Control*, 129, 571–583.

Neamtu, D., Fabregas, E., Wyns, B., De Keyser, R., Dormido, S., and Ionescu, C. (2011). A remote laboratory for mobile robot applications. In *18th IFAC World Congress (IFAC)*, 7280–7285.

Tabuada, P., G.J.Pappas, and Lima, P. (2005). Motion feasibility of multi-agent formations. *IEEE Trans. on Robotics*, 21, 387–392.

Vidal, R., O.Shakernia, and Sastry, S. (2003). Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation. In *Proc. of the Int. Conf. on Robotics and Automation*, 584–589.

Vlas, T. (2013). *Identification, Control and Vision-based Autonomous Navigation of an AR.Drone quadrotor*. Bachelor thesis, Ghent University.

Xu, C., Rahmani, A., and Egerstedt, M. (2010). Multi-uav convoy protection: an optimal approach to path planning and coordination. *IEEE Transactions on Robotics*, 26(2), 256–268.