23rd International Conference
ENGINEERING MECHANICS 2017
Svratka, Czech Republic, 15 – 18 May 2017

# FORMATION CONTROL OF MOBILE ROBOTS UNDER ROS

K. Besseghieur*, W. Kaczmarek**, J. Panasiuk***, P. Prusaczyk****

**Abstract:** *In this paper, a new framework is proposed for implementing the formation control laws on nonholonomic mobile robots based on ROS (Robot Operating System). To achieve the desired formation, mobile robots need to localize themselves within the environment, to communicate their positions to each other and to measure their corresponding velocities. ROS provides some convenient packages that make the formation problem easier to solve. We describe each of these packages and how they can be used to solve the formation control problem under ROS.*

**Keywords: Mobile robots, Formation control, Robot Operating System (ROS).**

## 1. Introduction

Multi-robot systems present a more robust and cheaper solution to certain tasks that are better performed using several low-cost robots rather than single, complex ones. A multi-robot system may be required to travel over large distances in order to reach a site related to a mission or task. While traversing the distances, it may be desirable for the robots to move in a rigid formation with fixed inter-robot distances. This gives rise to the formation control problem. Moreover, the latter has several potential applications for mine sweeping, boarder patrolling and for cooperative mapping to name a few.

The formation problem has been regarded as an important problem in multi-robot systems where the objective is to make a team of vehicles move toward and maintain a desired geometric pattern, while maintaining a featured motion. According to the survey presented in (Guanghua et al., 2013), and the references therein, formation structure can be divided into three strategies: the leader–follower strategy, the behavioral and the virtual structure approaches. Several approaches have been proposed in the literature to solve this problem. However, most of the existing literature tackle the theoretical side of the problem mainly the controller design is considered where several control strategies are adopted to make the formation errors converge to zero. Nevertheless, some of them have carried on real experiments to prove the effectiveness of their proposed controller. Furthermore, multi robots systems implementation on ROS has rarely been considered except in few works like (Muddu et al., 2015) about multi robots coverage and (Hennes et al., 2012) about multi robot collision avoidance.

In this paper, we propose a new framework for all the mobile robots based on ROS so that real experiments for the formation control problem can be conducted effectively. This allows all the researches in this area to assess the performance and effectiveness of their controllers throughout the experiments. Due to its simplicity and scalability, the leader-follower approach is considered in this paper. However, the proposed framework can be extended so that other formation strategies can be implemented. The remainder of the paper is organized as follows: the next section is dedicated for a small introduction about ROS concepts and their use in the formation problem. In section 3, formation control framework is sketched where each part of the latter is detailed and the needed ROS packages are presented. In the last section, we conclude our work and draw some future work directions.

* Khadir Besseghieur, MS.: Faculty of Mechatronics & Aerospace, Millitary University of Technology, Kaliskiego 2, 00-908, Warsaw, PL, besseghieurkh@hotmail.fr

** Wojciech Kaczmarek, PhD.: Faculty of Mechatronics & Aerospace, Millitary University of Technology, Kaliskiego 2, 00-908, Warsaw, PL, wojciech.kaczmarek@wat.edu.pl

*** Jarosław Panasiuk, PhD.: Faculty of Mechatronics & Aerospace, Millitary University of Technology, Kaliskiego 2, 00-908, Warsaw, PL, jaroslaw.panasiuk@wat.edu.pl

*** Piotr Prusaczyk, MS.: Faculty of Mechatronics & Aerospace, Millitary University of Technology, Kaliskiego 2, 00-908, Warsaw, PL, piotr.prusaczyk @wat.edu.pl

## 2. ROS background

ROS is a Linux-based, an open source software package that provides a software framework to aid in the development of complex robotic applications (Quigley et al., 2015). It is based on the concepts of nodes, topics, messages and services. A node is an executable program that performs computation. Nodes need to communicate with each other to complete the whole task. The communicated data are called messages. ROS provides an easy way for passing messages and establishing communication links between nodes, which are running independently. They pass these messages to each other over a Topic, which is a simple string. However, topics are asynchronous, synchronous communication is provided by services. Services act in a call-response manner where one node requests that another node execute a one-time computation and provide a response. For more details about ROS, the reader can refer to (Quigley et al., 2015).

We can see how these concepts help to solve the formation control problem in ROS. Nodes for instance are responsible for launching several algorithms to control, localize the robot and to transmit the data. The topic "/tf" is the holder of the robot's postures. Therefore, when a node needs a robot position, it has to subscribe to this topic. In our framework, services are essentially used in communication between robots where the control node solicits the communication node to transmit postures to another robot.

The presented formation implementation relies on several operational assumptions to narrow the implementation goal to a specific scope which is about testing the formation control laws with real experiments. First, it is assumed an occupancy grid representation of the static map is available to all robots. This removes the requirement for multi-robot SLAM and map merging, which are outside the scope of this work. Second, each robot knows its initial position and posesses the required sensors in order to maintain an accurate estimation of its pose within the two dimensional map using the navigation stack. A wireless communication network is assumed to be available over the entire coverage region.

Our test bench includes "Turtlebot" robots equipped with the Kinect camera sensor and with embedded laptops. The robots communicate over an 802.11n WIFI network (Fig. 1).

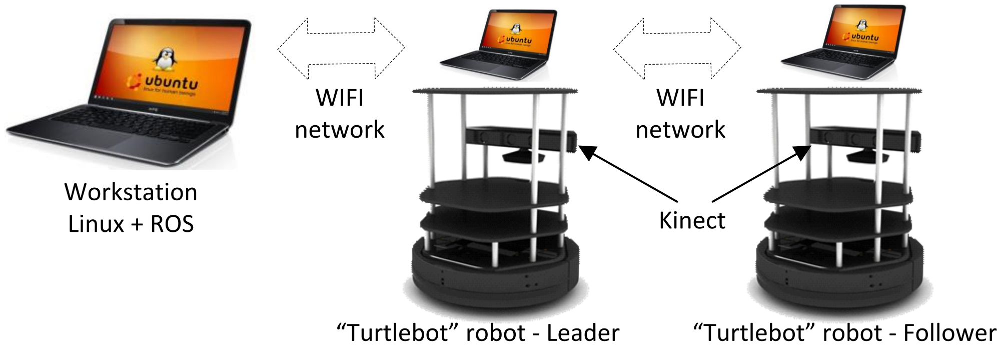


*Fig. 1: A general view of the system.*

## 3. Formation control framework on ROS

To build a robust formation control framework on ROS, we need before to know how each robot contributes to realize the whole formation. We decided to set up our framework according to what is needed in the leader-follower approach. Note that this framework can be extended to implement different formation control strategies.

In the leader-follower approach, each robot takes another neighboring robot as a leader to determine its motion. The leader robot moves along predefined trajectory while the follower robots keep track of the leader robot and maintain desired distance and bearing angle.

Most of the control laws proposed in the literature show that the leader's relative coordinates and velocities are needed in the control laws implemented on the followers. Therefore, the complete multi-robot formation control strategy can be divided into three phases that each robot must be able to carry out on its own. Each robot must be able to:

- Localize itself within the map.
- Communicate data with other robots. The communication direction is from the leader to its followers.

- Execute the implemented formation control laws. Only for the followers. Note that the leader can be directly controlled from the workstation keyboard using the "Teleoperation" ROS package.

In addition to the basic nodes needed for running the robot, three additional nodes are indispensable for controlling the robots to achieve formation. AMCL node for localization, Ad Hoc Communication node for data transmission and the control node that executes the control algorithm. The nodes are running simultaneously and thus they have to communicate to each other through ROS topics or ROS services as depicted in fig.2. Note that on each robot, all the nodes are executed under a specific namespace for the robot. The ROS parameter 'tf_prefix' is exclusive for each robot as well.

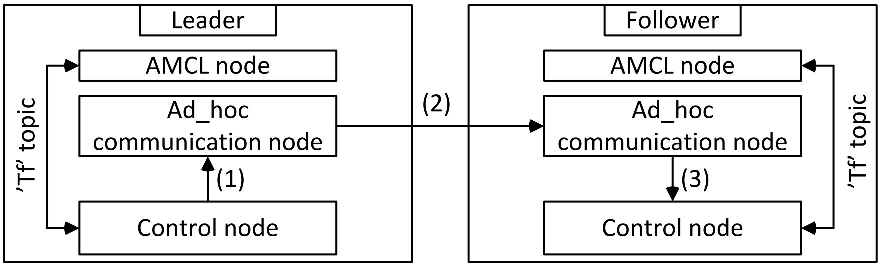


*Fig. 2: Formation control framework.*

### 3.1. Localization

The ROS navigation stack is used to provide the localization of the robot. The Localization method uses the Adaptive Monte-Carlo Localization (AMCL) approach presented in (Hennes et al., 2012), it is based on a weighted particle system in which each particle represents an estimated pose of the robot and consists of two phases of calculation: the prediction and update phases. AMCL combines the onboard encoders' measurements and the data provided by the Kinect sensor to provide an accurate estimation of the robot position. The AMCL node publishes the robot's postures into the 'tf' topic.

### 3.2. Data communication

Several ROS packages have been proposed to transmit data between several machines based on ROS. The multimaster_fkie package (Tiderko et al., 2016) allows the discovery and synchronization of robots as well as unicast and multicast transmissions based on UDP where two additional nodes need to be run. Master_discovery node connects to its local master and broadcasts the time stamp of the last change to the network and master_sync node connects to all known master_discovery nodes then it registers or unregisters the remotely available topics and services with the local ROS master. SocRob Multicast package is proposed in (Reis et al., 2013) based on Reconfigurable and Adaptive TDMA (RA-TDMA) communication protocol. The Ad Hoc Communication package considered in (Ander et al., 2014) is preferred in our framework where an Ad hoc On-Demand Distance Vector (AODV) for unicast and Multicast transmission is implemented. It uses automatic repeat request (ARQ) on data link and transport layers for unicast and multicast allowing reliable transmissions. Comparing with the multimaster_fkie package, the ad_hoc communication package provides more reliable transmission of data and the communication is only established on demand.

The communication between robots is mainly used for transmitting the leader's postures and velocities to its followers. These data are essential for controlling the follower to keep the desired separation and bearing with the leader. However, the two control nodes do not communicate directly, they utilize the intermediate Ad Hoc Communication node as depicted in Fig. 2. (1) represents a service call which includes the hostnames of the sender and the recipient robots, data to be transmitted and the topic where the data will be published at the destination. The Ad Hoc Communication package wraps the data into an extended MAC frame and transmits the frame using a raw socket (2). When the data has successfully been received at the destination, it will be published in the predefined topic. Note that, a custom ROS msg file that includes the variables definitions for the data to be transmitted must be defined as well as a new ROS srv file, which holds that custom msg file, is necessary in the service call for data transmission.

### 3.3. The control node

The control nodes are considered as the main part in the robots control. However, the algorithms differ when it comes to a leader or a follower robot control. As sketched in Fig. 3, after the initialization phase, the program goes into the control loop. The nodes then listen to their corresponding '/tf' topics and subscribe to the topic where the robot's velocities are being published as well as subscribing to the topics published by the ad hoc communication node to get the leader's postures and the workstation order. A test is held where only a received character from the workstation interrupts an endless wait loop. This order is transmitted using the same package Ad Hoc Communication. After receiving the order, each follower executes the implemented control laws, which are based on nonlinear system control theory. The obtained velocities are then published to the corresponding velocity command topics. Whereas for the leader, it is controlled from the workstation using the 'Teleoperation' ROS package.

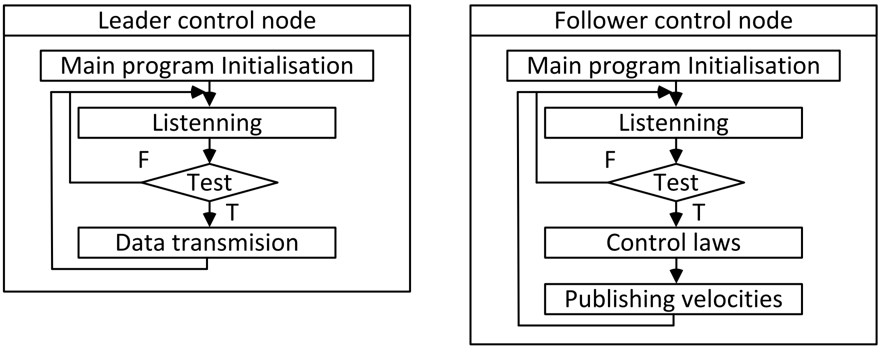


*Fig. 3: Control nodes algorithms.*

## 4. Conclusion

In this work, a ROS framework for implementing formation control algorithms on mobile robots based on ROS is designed that can be exploited by researches in this area to assess their control algorithms performances. The proposed framework is divided into three phases where the localization and communication phases are functionalities provided by existing ROS packages whereas the control phase indicates the custom ROS node. The framework has been tested on our test-bench, the communication is reliable but the localization can be more accurate, this will be tackled in our future work by improving the localization algorithm or employing better sensors than the Kinect camera where the Hokuyo Lidar is considered to be a better alternative. Future works include extending this framework for other formation control strategies than the leader-follower approach.

### Acknowledgement

My acknowledgements are directed to the co-authors of this paper for providing me with the necessary equipment to carry on this research and to the anonymous reviewers of this paper.

### References

Andre, T., Neuhold, D. and Bettstetter, C. (2014) Coordinated multi-robot exploration: Out of the box packages for ROS. In Globecom Workshops (GC Wkshps), pp. 1457-1462, IEEE.

Guanghua, W., Deyi, L., Wenyan, G. and Peng, J. (2013) Study on formation control of multi-robot systems. In Intelligent System Design and Engineering Applications, 2013 3 rd Int. Conf on pp. 1335-1339, IEEE.

Hennes, D., Claes, D., Meeussen, W. and Tuyls, K. (2012) Multi-robot collision avoidance with localization uncertainty. In Proc. 11th Int. Conf. on Autonomous Agents and Multiagent Systems-Volume 1, pp. 147-154.

Muddu, R.S.D., Wu, D. and Wu, L. (2015) A frontier based multi-robot approach for coverage of unknown environments. In Robotics and Biomimetics (ROBIO), 2015 IEEE Int. Conf on pp. 72-77, IEEE.

Quigley M., Gerky B. and Smart W.D. (2015) Programming Robots with ROS, a Practical Introduction to the Robot Operating System, First. O'Reilly Media, Inc.

Reis, J.C., Lima, P.U. and Garcia, J. (2013) Efficient distributed communications for multi-robot systems. In Robot Soccer World Cup, pp. 280-291, Springer Berlin Heidelberg.

Tiderko, A., Hoeller, F. and Röhling, T. (2016) The ROS Multimaster Extension for Simplified Deployment of Multi-Robot Systems, chapter in Robot Operating System (ROS) Volume 625 of the series Studies in Computational Intelligence pp. 629-650.