Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 96 (2021) 248–253

www.elsevier.com/locate/procedia

CIRPe 2020 – 8th CIRP Global Web Conference – Flexible Mass Customisation

A Comparison of Different Approaches for Formation Control of Nonholonomic Mobile Robots regarding Object Transport

Tobias Recker[a,*], Malte Heinrich, Annika Raatz[a]

[a]*Leibniz University Hannover, Institute of Assembly Technonolgy, An der Universitt 2, 30823 Garbsen, Germany*

Abstract

Controlling the formation of several mobile robots allows for the connection of these robots to a large virtual unit. This enables a group of mobile robots to carry out tasks that a single robot could not perform. For this purpose, the use of nonholonomic mobile robots is especially useful, as they often have a higher payload and are suitable for a wider range of terrains. However, most research in the area of formation control is focused on holonomic robots, since their superior mobility allows for better control and allows for the research on more sophisticated control techniques. The remaining articles explicitly dealing with nonholonomic robots often do cover common controllers, but do not include realistic simulations or comparison of different controls on the same trajectory. Therefore, in this paper, we present a comparative analysis of two frequently used control approaches. We compare the behavior of a l-ψ-controller and a Cartesian reference-based controller with different types of reference value generation and pose determination. The evaluation of all resulting control schemes is based on the task of collaborative object transport. To do so, we selected performance criteria geared towards applicability in real processes. In addition, we used an error model, which takes into account the noise and accuracy of all sensors (IMU and encoder) as well as the drift in odometry caused by the slip of the robot's wheels. The comparison includes a series of simulations using two trajectories with a changing number of robots and different formation geometries. In the simulations we got slightly better results for the Cartesian control law.

Keywords: Formation control; Object transport; Nonholonomic mobile robots

1. Introduction

Robot formations are often used to extend the capabilities of a single robot or to break down complex overall tasks into simpler subtasks [1]. In this way, robot formations increase the flexibility of the overall robotic system and improve it's ability to adapt in the context of flexible mass customization. To function as a unit, it is usually necessary for the group of robots to maintain their formation even under the influence of disturbances. Consequently, several approaches to formation control have been presented in the past. In this paper, two of these approaches will be compared using the example of cooperative object transport. As a first step, we simulate the movement of a formation of several mobile robots along a given trajectory. We will pay special attention to the deviation in the distance between the robots in our formation, since the workpiece may fall off one of the robots if the deviation becomes too large. For this publication, we will limit our investigations to decentralized control approaches, as these require less communication effort and can be scaled more easily.

1.1. Related work

In formation control, a distinction is usually made between holonomic and nonholonomic robots. Since the actual formation control is much simpler for holonomic robots, the research in this field focuses on other aspects like obstacle avoidance [2] or constraint optimization [3]. Due to the kinematic constraints, the problem of formation control for non-holonomic robots is much more complex, so that many approaches have been tested in the past. In addition to behavior-based approaches, virtual structure, and leader-follower approaches, several other approaches are mentioned and evaluated in literature [4]. Despite of these publications, that deal with the control of formation during movement are relatively scarce and often focus on the response to static or dynamic obstacles [5]. Also, they rarely

* Corresponding author. Tel.: +49-511-762-18246
E-mail address: Recker@match.uni-hannover.de (Annika Raatz).

2212-8271
10.1016/j.procir.2021.01.082

contain a detailed quantitative analysis of the deviations that occur [6],[7]. This is because most of the work is focused either on reaching the desired formation [8][9] or changing between two formations [2].

However, for the design of multi-robot processes and the transfer to industrial applications, it is crucial to take into account the occurring errors and error types. This especially applies to cooperative object transports, where a large component is to be transported by the interaction of several smaller robots. If the formation is not maintained well enough, the workpiece will slide back and forth on the robots and, in the worst case, fall off one of the robots. If the object is fixed to the robots, forces are introduced into the component instead, which may damage the component. In the literature, some studies have been presented which investigate the transport of objects with a group of mobile robots [3],[10].In [10], the resulting errors are described in great detail, although there is no comparison and classification of the measured variables and no statement is made as to how well the control scheme is suitable for object transport. As the results from other studies can not easily be compared for the same setup, we have built a framework to test different control laws under identical conditions (see Sec. 2.1) and then determine which controller is fitting our application the best.[1]

1.2. Aim of application and basic concept

In the future, we plan on using a group of MiR 200 mobile robots to carry large and heavy objects. The robots will be equiped with a compensation unit which compensates for small lateral movements of the workpiece and a rotary bearing to decouple the angle of rotation of the cargo platform from that of the robot (see Fig. 1).

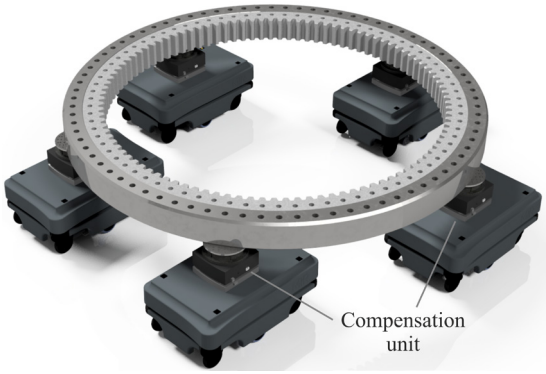


Fig. 1: 5 Mobile Industrial Robots (MIR) 200 carrying a crane bearing

In this paper, we want to find a suitable controller structure for this transport and simulate which deviations are to be expected during transport and then design the compensation units accordingly. To do so, we will first describe the subtasks of formation control and formulate the control problem. In section 2, we then present the two controllers and our implementation. Section 4 contains all of our simulation results. Finally, concluding remarks and future work are given in Section 5.

[1] All of our code including controller setting can be found under https://github.com/matchRos/MiR200_Formation

2. Formation Control

The main goal of this paper is the evaluation and comparison of leader-follower based formation control approaches for mobile robots within transport processes. Therefore, a control structure for solving the formation control problem is presented. Since there is a high amount of literature on different control laws, no new control law is developed, but two existing laws are embedded into an applicable structure. The first one is the l-ψ-control (l.c.) presented by *Desay et al.* in [6] and the second is the Cartesian control (c.c.) presented by *Kanayama et al.* in [11]. The control laws should be interchangeable (see Sec. 1.1) , which can be achieved by splitting the control problem into four smaller sub-tasks:

- Abstraction of interfaces for using both control laws independently
- Determination of control velocities by the control law
- Generation of target states for each formation member
- Estimation of current states for each formation member

We will start with the abstraction of an interface for both control laws and the determination of control velocities in this section an present the other sub-tasks in Sec. 3.

2.1. Abstraction of an interface

For our controller, we are using the structure shown in Fig. 2. We have integrated all sub-tasks into individual blocks, to enable independent testing. All blocks were designed to use standardized interfaces to ensure interchangeability. Using the *control law* from figure 2 as an example, this means that the control law calculates the new speed target $\boldsymbol{v}_{cf}$ for every follower robot, based on its measured speed $\boldsymbol{v}_f$ and position $\boldsymbol{x}_f$ as well as the position reference $\tilde{\boldsymbol{x}}_f$ of the respective follower robot, the feedforward velocity $\tilde{\boldsymbol{v}}_f$ and the speed of the leader $\tilde{\boldsymbol{v}}_l$. All these variables must be in the correct state-space representation for the corresponding control law:

$$\boldsymbol{v}_{cf} = \boldsymbol{f}(\boldsymbol{e}, \boldsymbol{x}_f, \boldsymbol{v}_l, \boldsymbol{v}_f, \tilde{\boldsymbol{v}}_f) \begin{cases} \boldsymbol{x}_f = {}_{(l\psi)}\boldsymbol{x}_f \text{ for l.c.} \\ \boldsymbol{e} = {}_{(l\psi)}\boldsymbol{e} \text{ for l.c.} \\ \boldsymbol{f} = \boldsymbol{f}_1 \text{ for l.c.} \\ \tilde{\boldsymbol{v}}_f = - \text{ for l.c.} \\ \boldsymbol{x}_f = - \text{ for c.c.} \\ \boldsymbol{e} = {}_{(xy\phi)}\boldsymbol{e} \text{ for c.c} \\ \boldsymbol{f} = \boldsymbol{f}_2 \text{ for c.c.} \\ \tilde{\boldsymbol{v}}_f = {}_{(xy\phi)}\tilde{\boldsymbol{v}}_f \text{ for l.c.} \end{cases} \quad (1)$$

All other interfaces ($\tilde{\boldsymbol{v}}_f$,$\tilde{\boldsymbol{x}}_f$,$\boldsymbol{v}_f$,$\boldsymbol{x}_f$,$\boldsymbol{v}_l$,$\boldsymbol{x}_l$) were defined and implemented according to the same principle. The only exception being that the current position of the leader $\boldsymbol{x}_l$ can only be described in a ${}_{(xy\phi)}\boldsymbol{x}_l$ form, as the l-ψ-coordinates only describe a position relative to the leader.

2.2. Determination of control velocities

The movements of each individual robot are controlled by setting control velocities. For robots with a differential drive

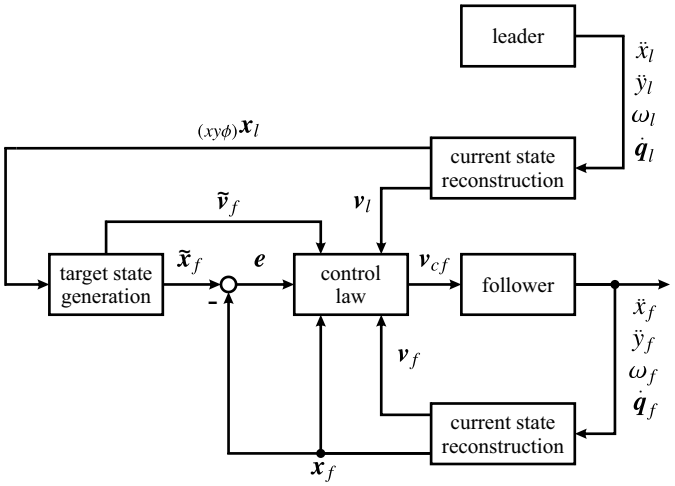


Fig. 2: Control structure gained by combining the mentioned tasks.

such as the MiR200, these are the translational speed v and the rotational speed ω. Both speeds are provided by the respective control law (see Sec. 2.2.1 and 2.2.2).

2.2.1. l-ψ-control

The first implemented law is the l-ψ-control law. Within this control law, the state of the mobile robot is constructed as a combination of the distance l_{lf} between a leading and the following robot and the angle ψ_{lf} between them. Based on this state representation of the mobile robots, the linear v_{cf} and angular control velocities ω_{cf} of a specific following robot are gained by the method of exact linearization [6]:

$$\omega_{cf} = \frac{\cos\gamma}{d}\alpha_2 l_{lf}\left(\tilde{\psi}_{lf} - \psi_{lf}\right) - v_l \sin\psi_{lf} + l_{lf}\omega_l + \xi_{lf}\sin\gamma \quad (2)$$

$$v_{cf} = \xi_{lf} - d\omega_f \tan\gamma, \quad (3)$$

while substituting

$$\xi_{lf} = \frac{\alpha_1\left(\tilde{l}_{lf} - l_{lf}\right) + v_l\cos\psi_{lf}}{\cos\gamma}. \quad (4)$$

and introducing the minimal distance d between both robots and $\gamma = \psi_l + \psi_{lf} - \psi_f$. For the sake of compactness, we introduce the following quantities, where the prescript $(l\psi)$ identifies the l-ψ state representation:

State name	State description	State representation (t)
$_{(l\psi)}\boldsymbol{x}_f$	current state	$_{(l\psi)}\boldsymbol{x}_f = (l_{lf}, \psi_{lf}, \phi_f)^T$
$_{(l\psi)}\tilde{\boldsymbol{x}}_f$	target state	$_{(l\psi)}\tilde{\boldsymbol{x}}_f = (\tilde{l}_{lf}, \tilde{\psi}_{lf}, \tilde{\phi}_f)^T$
$_{(l\psi)}\boldsymbol{e}$	control difference	$_{(l\psi)}\boldsymbol{e} = {}_{(l\psi)}\tilde{\boldsymbol{x}}_f - {}_{(l\psi)}\boldsymbol{x}_f$
$\boldsymbol{v}_{cf}$	control velocity	$\boldsymbol{v}_{cf} = (v_{cf}, \omega_{cf})^T$
$\boldsymbol{v}_f$	current velocity	$\boldsymbol{v}_f = (v_f, \omega_f)^T$
$\boldsymbol{v}_l$	leader velocity	$\boldsymbol{v}_l = (v_l, \omega_l)^T$

With these definitions the general transmission behavior $\boldsymbol{f}_1$ of the angle distance control law is:

$$\boldsymbol{v}_{cf} = \boldsymbol{f}_1\left({}_{(l\psi)}\boldsymbol{e}, {}_{(l\psi)}\boldsymbol{x}_f, \boldsymbol{v}_l, \boldsymbol{v}_f\right). \quad (5)$$

2.2.2. Cartesian control

The second control law analyzed within this paper is for a Cartesian controller (6), (7). This law is based on the Cartesian state representation x, y, ϕ of formation members. The control velocity $\boldsymbol{v}_{cf}$ of a specific following robot is gained by Lyapunov stabilizing the error dynamics of the mobile robot [11]:

$$\omega_{cf} = \tilde{\omega}_f + \tilde{v}_f\left(\alpha_3\left(\tilde{y}_f - y_f\right) + \alpha_4\sin\left(\tilde{\phi}_f - \phi_f\right)\right) \quad (6)$$

$$v_{cf} = \tilde{v}_f\cos\left(\tilde{\phi}_f - \phi_f\right) + \alpha_3\left(\tilde{x}_f - x_f\right) \quad (7)$$

Analog to the l-ψ-control we define additional quantities:

State Name	State description	State representation (t)
$_{(xy\phi)}\boldsymbol{x}_f$	current state	$_{(xy\phi)}\boldsymbol{x}_f = (x_f, y_f, \phi_f)^T$
$_{(xy\phi)}\tilde{\boldsymbol{x}}_f$	target state	$_{(xy\phi)}\tilde{\boldsymbol{x}}_f = (\tilde{x}_f, \tilde{y}_f, \tilde{\phi}_f)^T$
$_{(xy\phi)}\boldsymbol{e}$	control difference	$_{(xy\phi)}\boldsymbol{e} = (\tilde{x}_f - x_f, \tilde{y}_f - y_f, \tilde{\phi}_f - \phi_f)^T$
$\tilde{\boldsymbol{v}}_f$	feed-forward velocity	$\tilde{\boldsymbol{v}}_f = (\tilde{v}_f, \tilde{\omega}_f)^T$

With the feed-forward velocity provided by the target state generation (see Fig. 2 and Sec. 3.2) the transfer characteristics $\boldsymbol{f}_2$ of the Cartesian control law result in:

$$\boldsymbol{v}_{cf} = \boldsymbol{f}_2\left({}_{(xy\phi)}\boldsymbol{e}, \tilde{\boldsymbol{v}}_f\right) \quad (8)$$

3. Subtasks and problem formulation

Another major challenge for later application is the determination of target values and especially the current pose. As will be shown in Sec. 4, errors in the pose estimation and the generation of target states have a significant impact on the performance of the control structure. Thus, both sub-tasks will be described in more detail in this section.

3.1. Generation of target states

Since our control laws are based on a leader-follower approach, each member of the formation determines its target states independently, based on the state given by a leading robot (also see [11],[6]). This generation of target states can primarily be achieved in two ways. Firstly, it is possible to calculate a trajectory for every follower based on the target path of the leader. This calculation can be performed a priori, but also necessitates that the trajectory of the leader is known in advance. The second possibility is the computation from the current position of the leading robot. In both cases, the rigid body transformation from the leading robot to the following robots is used.

The rigid body transformation ensures that every follower robot within this approach behaves as if it was fixed to a rigid point relative to their leading robot within the formation. For a free system without any constraints, a simple rigid transformation from leader to follower would be sufficient. However, this paper investigates the behavior of nonholonomic differential drive mobile robots. Therefore, the original rigid transformation for a following robot f based on the current pose of its leading robot l and its target relative pose from the leader $\tilde{\boldsymbol{r}}_{lf}$ is modified:

$$ {}_{(xy\phi)}\tilde{\boldsymbol{x}}_f = \overbrace{\boldsymbol{S}_f}^{\text{locking matrix}} \overbrace{\left({}_{(xy\phi)}\boldsymbol{x}_l + \boldsymbol{R}(\phi_l)\tilde{\boldsymbol{r}}_{lf}\right)}^{\text{rigid transformation}} + \overbrace{\boldsymbol{s}_f}^{\text{vector of constrains}} \tag{9} $$

The modifications in equation (9) are the multiplication of a locking matrix $\boldsymbol{S}_f$, and the addition of a vector $\boldsymbol{s}_f$, which contains the constraints of the robot. For the differential drive robot in this paper, these are:

$$ \boldsymbol{S}_f = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{s}_f = \begin{pmatrix} 0 \\ 0 \\ \arctan\left(\frac{\dot{y}_f}{\dot{x}_f}\right) \end{pmatrix} \tag{10} $$

For the Cartesian control, this modified rigid transformation can be used as-is. The transfer characteristics $\boldsymbol{g}_1$ of the target state generation in the Cartesian case can be expressed as:

$$ {}_{(xy\phi)}\tilde{\boldsymbol{x}}_f = \boldsymbol{g}_1\left({}_{(xy\phi)}\boldsymbol{x}_l, \tilde{r}_{lf}\right), \tag{11} $$

For the l-ψ case, a state transformation has to be determined additionally. For transforming a given state vector (x, y, ϕ) into a state representation $(l_{lf}, \psi_{lf}, \psi_f)$ equation (12) is used.

$$ \begin{aligned} {}_{(l\psi)}\tilde{\boldsymbol{x}}_f &= \begin{pmatrix} \tilde{l}_{lf} \\ \tilde{\psi}_{lf} \\ \tilde{\phi}_f \end{pmatrix} = \begin{pmatrix} \sqrt{(x_l - \tilde{x}_f)^2 + (y_l - \tilde{y}_f)^2} \\ \arctan\frac{y_l - \tilde{y}_f}{x_l - \tilde{x}_f} \\ \tilde{\phi}_f \end{pmatrix} \\ &= \hat{\boldsymbol{g}}_2\left({}_{(xy\phi)}\tilde{x}_f, {}_{(xy\phi)}\boldsymbol{x}_l\right) \end{aligned} \tag{12} $$

This state transformation $\hat{\boldsymbol{g}}_2$ applied to the general target state generation from equation (11) leads to the transfer characteristics $\boldsymbol{g}_2$ of the l-ψ-control:

$$ \begin{aligned} {}_{(l\psi)}\tilde{\boldsymbol{x}}_f &= \hat{\boldsymbol{g}}_2\left({}_{(xy\phi)}\tilde{x}_f, {}_{(xy\phi)}\boldsymbol{x}_l\right) \\ &= \hat{\boldsymbol{g}}_2\left(\boldsymbol{g}_1\left({}_{(xy\phi)}\boldsymbol{x}_l, \tilde{r}_{lf}\right), {}_{(xy\phi)}\boldsymbol{x}_l\right). \\ &= \boldsymbol{g}_2\left({}_{(xy\phi)}\boldsymbol{x}_l, \tilde{r}_{lf}\right) \end{aligned} \tag{13} $$

3.2. Feed forward

In contrast to the l-ψ-control, the Cartesian control requires the target velocity $\tilde{\boldsymbol{v}}_f$ of the following robot (see (7)). This feed forward velocity is determined from the time derivative of the modified rigid transformation (9):

$$ \begin{pmatrix} \dot{\tilde{x}}_f \\ \dot{\tilde{y}}_f \\ \dot{\tilde{\omega}}_f \end{pmatrix} = \boldsymbol{S}_f\left(\dot{\boldsymbol{x}}_l + \begin{pmatrix} 0 \\ 0 \\ \omega_l \end{pmatrix} \times \boldsymbol{R}(\phi_l)_{(l)}\tilde{\boldsymbol{r}}_{lf}\right) + \dot{\boldsymbol{s}}_f \tag{14} $$

and the correspondence:

$$ \tilde{\boldsymbol{v}}_f = \begin{pmatrix} \sqrt{\dot{\tilde{x}}_f + \dot{\tilde{y}}_f} \\ \frac{\dot{\tilde{y}}_f \dot{x}_f - \ddot{x}_f \dot{y}_f}{\tilde{v}_f^2} \end{pmatrix} \tag{15} $$

Finally, the target state generation is extended by the feed forward transfer characteristics:

$$ \tilde{\boldsymbol{v}}_f = \boldsymbol{h}_1\left({}_{(xy\phi)}\dot{\boldsymbol{x}}_l, \tilde{r}_{lf}\right) \tag{16} $$

which leads to the complete target state generation:

$$ \begin{aligned} \tilde{\boldsymbol{x}}_f &= \boldsymbol{g}\left({}_{(xy\phi)}\boldsymbol{x}_l, \tilde{r}_{lf}\right) \\ \tilde{\boldsymbol{v}}_f &= \boldsymbol{h}\left({}_{(xy\phi)}\boldsymbol{x}_l, \tilde{r}_{lf}\right) \end{aligned} \begin{cases} \tilde{\boldsymbol{x}}_f = {}_{(l\psi)}\tilde{\boldsymbol{x}}_f & \text{for l.c.} \\ \boldsymbol{g} = \boldsymbol{g}_2 & \text{for l.c.} \\ \boldsymbol{h} = \text{-} & \text{for l.c.} \\ \tilde{\boldsymbol{x}}_f = {}_{(xy\phi)}\tilde{\boldsymbol{x}}_f & \text{for c.c.} \\ \boldsymbol{g} = \boldsymbol{g}_1 & \text{for c.c.} \\ \boldsymbol{h} = \boldsymbol{h}_1 & \text{for c.c.} \end{cases} \tag{17} $$

Equations (14) and (15) suggest and the evaluation in Sec. 4 proves, that the noisy velocity data of the leading robot causes a negative effect for Cartesian control. This negative effect is remedied by filtering the velocity data of the following robots by a moving average filter, which improves the accuracy of numerical differentiation that is used to obtain the second derivatives in Eq. (15).

3.3. Estimation of current states for each formation member

In mobile robotics, a common tactic is to estimate the pose of a single mobile robot from the odometry. The odometry simply integrates the velocity ${}_{(xy\phi)}\boldsymbol{x}$ of the robot, from a given initial pose $\boldsymbol{x}_0$:

$$ {}_{(xy\phi)}\boldsymbol{x} = \int {}_{(xy\phi)}\dot{\boldsymbol{x}}dt + \boldsymbol{x}_0 \tag{18} $$

The velocity ${}_{(xy\phi)}\dot{\boldsymbol{x}}$ is calculated from the differential kinematics $\boldsymbol{J}$ of the platform using it's wheel velocities $\dot{\boldsymbol{q}}$

$$ {}_{(xy\phi)}\dot{\boldsymbol{x}} = \boldsymbol{j}(\dot{\boldsymbol{q}}) = \boldsymbol{J}\dot{\boldsymbol{q}}. \tag{19} $$

With equation (18) and (19) the transmission characteristics of pose estimation can be expressed as:

$$ {}_{(xy\phi)}\boldsymbol{x} = \boldsymbol{i}_1(\dot{\boldsymbol{q}}) = \int \boldsymbol{J}\dot{\boldsymbol{q}}dt + \boldsymbol{x}_0 \tag{20} $$

This pose estimation can be used for the cartesian control (c.c.), whereas the l-ψ-control (l.c.) requires a relative position and therefore uses the state transformation $\hat{\boldsymbol{i}}_2$:

$$ \begin{aligned} {}_{(l\psi)}\boldsymbol{x}_f &= \begin{pmatrix} l_{lf} \\ \psi_{lf} \\ \phi_f \end{pmatrix} = \begin{pmatrix} \sqrt{(x_l - x_f)^2 + (y_l - y_f)^2} \\ \arctan\frac{y_l - y_f}{x_l - x_f} \\ \phi_f \end{pmatrix} \\ &= \hat{\boldsymbol{i}}_2\left({}_{(xy\phi)}\boldsymbol{x}_f, {}_{(xy\phi)}\boldsymbol{x}_l\right) \end{aligned} \tag{21} $$

This leads to a transfer characteristics of the target state generation for l-ψ-control:

$$ \begin{aligned} {}_{(l\psi)}\boldsymbol{x}_f &= \hat{\boldsymbol{i}}_2\left({}_{(xy\phi)}\boldsymbol{x}_f, {}_{(xy\phi)}\boldsymbol{x}_l\right) \\ &= \hat{\boldsymbol{i}}_2\left(\boldsymbol{i}_1\left(\boldsymbol{q}_f\right), {}_{(xy\phi)}\boldsymbol{x}_l\right) \\ &= \boldsymbol{i}_2\left(\boldsymbol{q}_f, {}_{(xy\phi)}\boldsymbol{x}_l\right) \end{aligned} \tag{22} $$

The pose estimations (20) and (22) solely based on odometry are subject to a high drift, which prevents their applicability to a real transport process (see Sec. 4). Therefore, an improved pose estimation, based on a sensor data fusion by a Kalman filter is implemented. For this fusion, the pose estimation (20) is fused with measures $(\ddot{x}, \ddot{y}, \omega)$ from an inertial measurement unit (IMU). This is done by adding a Kalman filter $\boldsymbol{k}(\ddot{x}, \ddot{y}, \omega, \boldsymbol{i}_1(\boldsymbol{q}))$:

$$ {}_{(xy\phi)}\boldsymbol{x}_l = \boldsymbol{k}\left(\ddot{x}_l, \ddot{y}_l, \omega_l, \boldsymbol{i}_1\left(\boldsymbol{q}_l\right)\right) \tag{23} $$

and

$$_{(xy\phi)}\boldsymbol{x}_f = \boldsymbol{k}\left(\ddot{x}_f, \ddot{y}_f, \omega_f, \boldsymbol{i}_1\left(\boldsymbol{q}_f\right)\right) \tag{24}$$

More on this approach can be found in [12].

4. Evaluation

As described in the beginning, the purpose of this work is to select a controller structure for the subsequent object transport using real MIR200 Robots and to dimension the compensation units. Accordingly, we use performance parameters for the evaluation, which assess the suitability of a formation control specifically for the cooperative transport of objects. Here, the deviation in the distance between the robots is especially important, as the workpiece could fall off the formation if this value gets too high. The reference value for the distance $\tilde{l}_{ij}$ between robot i and j results from the desired formation and always remains constant with a given formation. Accordingly, the error Δl_{ij} can be calculated as the difference between the nominal and actual distance l_{ij} of two robots:

$$\Delta l_{ij}(t) = \tilde{l}_{ij}(t) - l_{ij}(t) \tag{25}$$

In this section, we will refer to the average of this error as MDE (mean distance error) and to the maximum as MAX.

$$\mathrm{MDE}_{ij} = \frac{1}{T}\int_0^T \Delta l_{ij}(t)^2 dt \qquad \mathrm{MAX}_{ij} = \max_{t>0}(|\Delta l_{ij}(t)|) \tag{26}$$

For this paper, we will use two different types of trajectories. First, these are Lissajous figures, since they include areas of high v and low ω as well as areas of low v and high ω (see Fig. 3) and therefore cover the entire system behavior. Second, we use a standard path planner to generate a more application-oriented leader trajectory (Fig. 4).

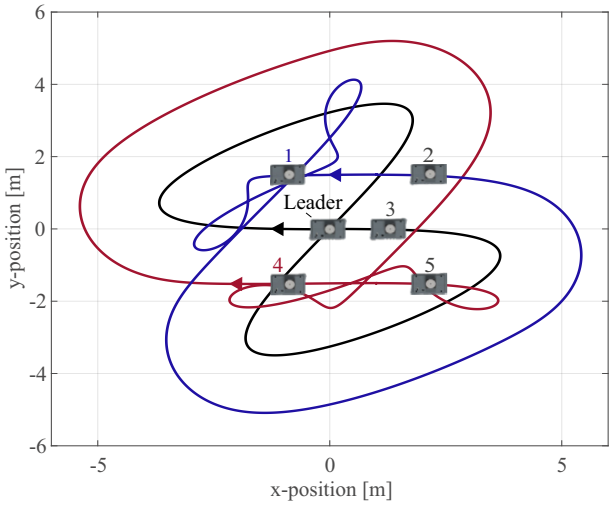


Fig. 3: A formation consisting of 6 robots moving along a Lissajous trajectory with frequency ratio 1:2

The following figures Fig. 5 and Fig. 6 show a comparison of the Cartesian- and l-ψ-control on the Lissajous trajectory from Fig. 3. Each row contains the MDE or MAX from each of the 6 robots to the 5 others. Since the distance from robot i to robot j is the same as from j to i, the matrices are symmetrical. As can be seen, especially in the comparison of the measurements from Fig. 6, the Cartesian control provides better results for the given trajectory. While the maximum deviation is similar (for the Lissajous trajectory), the average error for l-ψ-control is twice as large. This may be partly due to a sub-optimal parameter setting of the controllers, but also to the lack of feed forward velocities. In terms of transport, this would mean a higher continuous load on the workpiece as well as on the compensation units during transport.

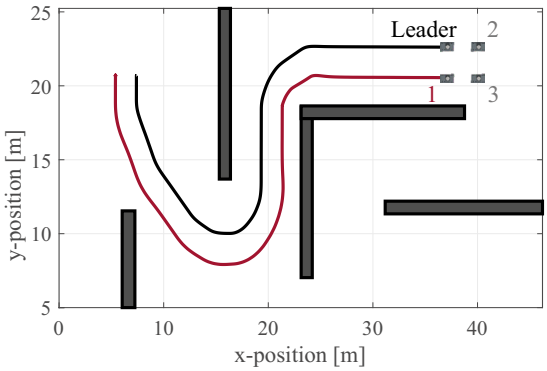
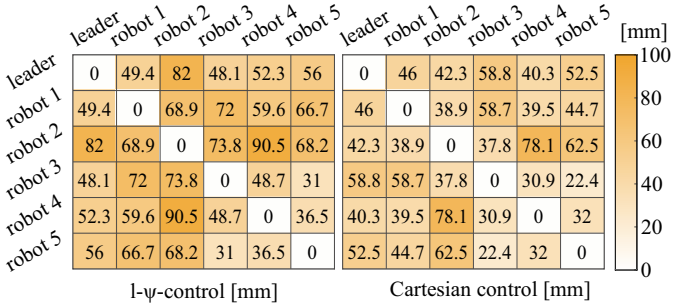


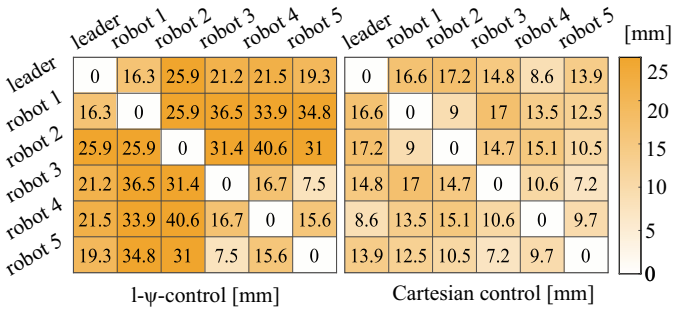
Fig. 4: A more application-oriented trajectory with 4 mobile robots



	leader	robot 1	robot 2	robot 3	robot 4	robot 5	leader	robot 1	robot 2	robot 3	robot 4	robot 5
leader	0	49.4	82	48.1	52.3	56	0	46	42.3	58.8	40.3	52.5
robot 1	49.4	0	68.9	72	59.6	66.7	46	0	38.9	58.7	39.5	44.7
robot 2	82	68.9	0	73.8	90.5	68.2	42.3	38.9	0	37.8	78.1	62.5
robot 3	48.1	72	73.8	0	48.7	31	58.8	58.7	37.8	0	30.9	22.4
robot 4	52.3	59.6	90.5	48.7	0	36.5	40.3	39.5	78.1	30.9	0	32
robot 5	56	66.7	68.2	31	36.5	0	52.5	44.7	62.5	22.4	32	0

Fig. 5: MAX of the l.c. compared to the c.c. for a Lissajous trajectory

In Fig 7 this difference can also be seen in a statistical analysis of the error. For this purpose, the deviations (Δl_{ij}) between all robots from three scenarios were cumulated and displayed as a boxplot. In the first case, we use the pose estimation of the leader to calculate the target states for all following robots. This means that errors affecting the master or the pose estimation also affect the followers. In the second scenario, the followers



	leader	robot 1	robot 2	robot 3	robot 4	robot 5	leader	robot 1	robot 2	robot 3	robot 4	robot 5
leader	0	16.3	25.9	21.2	21.5	19.3	0	16.6	17.2	14.8	8.6	13.9
robot 1	16.3	0	25.9	36.5	33.9	34.8	16.6	0	9	17	13.5	12.5
robot 2	25.9	25.9	0	31.4	40.6	31	17.2	9	0	14.7	15.1	10.5
robot 3	21.2	36.5	31.4	0	16.7	7.5	14.8	17	14.7	0	10.6	7.2
robot 4	21.5	33.9	40.6	16.7	0	15.6	8.6	13.5	15.1	10.6	0	9.7
robot 5	19.3	34.8	31	7.5	15.6	0	13.9	12.5	10.5	7.2	9.7	0

Fig. 6: MDE of the l.c. compared to the c.c. for a Lissajous trajectory

use the reference trajectory of the master to calculate their target states. This leads to a better control quality but requires the trajectory to be known in advance (see Sec. 3.1).

In the third scenario, we want to show the influence of noisy input data on the Cartesian control. As described in Sec. 3.2, we always use an input filter for the Cartesian control (scenarios I and II)[2]. In scenario III, this filter is disabled to show the influence of noisy input data. The simulations also already show that the position estimation solely based on odometry does not provide any usable results. As a consequence of the simulated wheel slip and the noise in the encoder data, individual robots deviated from their target position to such an extent they collide with each other after only three-quarters of the trajectory. Even using the fused pose estimation, there is a significant drift in the distance between individual robots. This becomes especially noticeable on the second trajectory (Fig. 4), which resembles a more realistic transport process.

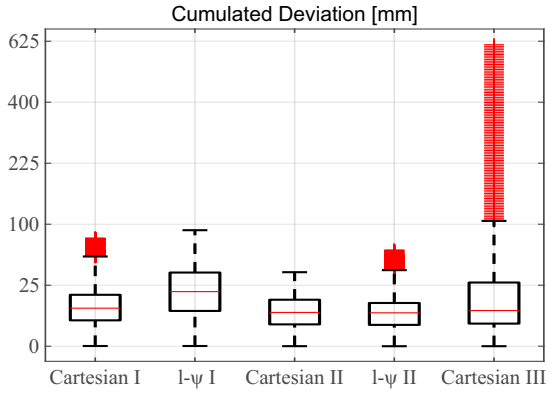


Fig. 7: Cumulated deviations of 6 robots on a Lissajous trajectory (using I: the actual leader-pose as a reference, II: the desired leader trajectory as a reference, III: no input filters)

After the 45.6 meters long trajectory, robots 1 and 2 drifted apart 78 *mm*. The other major source of error are changes in the angular velocity. Although in Fig. 8 it may seem like the deviations correspond to the angular velocity, they are really mainly influenced by changes in the velocity. This characteristic can primarily be observed on trajectories with high constant angular velocity (e.g., a circle)[3].

Excluding the drift, actual jumps in the error are quite small. A compensation path of 50 mm should, therefore, be completely sufficient, in the case of Cartesian control.

5. Conclusion and future work

In this paper, we compared the performance and applicability of two common control laws for formation control. Additionally, we developed a control structure that allows us to interchange the individual components of our controller by using standard interfaces. After formally describing these interfaces,

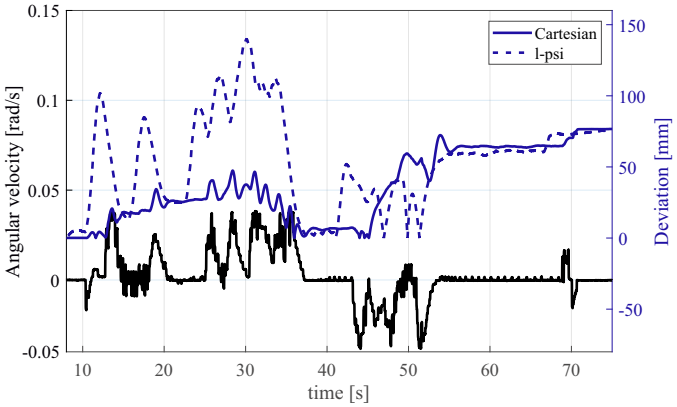


Fig. 8: Angular velocity of robot 1 including the deviation to the leader

we presented every component in detail. We also introduced two performance indicators to compare the control performance of both controllers. As has been shown, in particular with the application-oriented trajectory, the Cartesian control generally provides better results. However, this is at the expense of the susceptibility to noise. Another important source of error is the sensor drift. In the future, this error should be compensated either by even better compensation methods or in combination with an absolute measuring method. Here, methods that use the integrated laser scanners are particularly promising. Next, we plan to use our control structure to simulate more control laws and finally implement them on a formation of real robots.

References

[1] George Michalos, Niki Kousi, Sotiris Makris, George Chryssolouris, Performance Assessment of Production Systems with Mobile Robots, Procedia CIRP,Volume 41,2016,Pages 195-200,ISSN 2212-8271,
[2] J. P. Desai, J. P. Ostrowski and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," in IEEE Transactions on Robotics and Automation, vol. 17, no. 6, pp. 905-908, Dec. 2001.
[3] Alonso-Mora, J., Baker, S. and Rus, D. (2017) Multi-robot formation control and object transport in dynamic environments via constrained optimization, The International Journal of Robotics Research, 36(9).
[4] Yang Quan Chen and Zhongmin Wang, "Formation control: a review and a new consideration," 2005 IEEE/RSJ IROS, Edmonton, Alta, 2005.
[5] Mohammad Deghat, Iman Shames, Brian D.O. Anderson, Safe Formation Control with Obstacle Avoidance,IFAC Proceedings Volumes,Volume 44, Issue 1,2011,Pages 11252-11257.
[6] J. P. Desai, J. Ostrowski and V. Kumar, "Controlling formations of multiple mobile robots," Proceedings. 1998 IEEE ICRA, Leuven, Belgium.
[7] F. Xie and R. Fierro, "On Motion Coordination of Multiple Vehicles with Nonholonomic Constraints," 2007 ACC, New York, 2007.
[8] E. Montijano, E. Cristofalo, M. Schwager and C. Sagues, "Distributed formation control of non-holonomic robots without a global reference frame," 2016 IEEE ICRA, Stockholm, 2016, pp. 5248-5254.
[9] D. V. Dimarogonas and K. H. Johansson, "Further results on the stability of distance-based multi-robot formations," 2009 American Control Conference, St. Louis, MO, 2009, pp. 2972-2977.
[10] A. Yufka, O. Parlaktuna and M. Ozkan, "Formation-based cooperative transportation by a group of non-holonomic mobile robots," 2010 IEEE SMC, Istanbul, 2010, pp. 3300-3307.
[11] Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," Proceedings., IEEE ICRA, Cincinnati, OH, USA, 1990, pp. 384-389 vol.1.
[12] Moore T., Stouch D. (2016) A Generalized Extended Kalman Filter Implementation for the Robot Operating System. IAS 13. Advances in Intelligent Systems and Computing, vol 302. Springer, Cham

[2] the *l-ψ*-control is far less susceptible to noise so no input filer is required
[3] We were not able to include these measurements in this publication, but we will include all measurement data as separate files