



# Time-optimal trajectory planning for a rigid formation of nonholonomic mobile platforms

Simon Schmidt · Hubert Gatringer · Andreas Mueller

Received: 1 June 2023 / Accepted: 4 August 2023 / Published online: 25 September 2023  
 © The Author(s) 2023

**Abstract** This paper presents a method for planning time-optimal trajectories for a formation of multiple nonholonomic (heavy duty) platforms (HDPs) to cooperatively transport an object to a specified pose. The first part addresses the mobile platforms themselves while the second part provides a trajectory planning approach derived from the well-known *virtual leader approach*. In order to ensure proper transport of the shared payload, the vehicles are modeled individually, resulting in a formation control problem. The goal of the optimization process is to minimize a cost function that balances time optimality, smooth control signals, and formation rigidity. The optimal control problem (OCP) takes into account the kinematics of the vehicles as well as their physical limitations. It is solved by using a multiple shooting method, which yields the desired trajectories for all vehicles while ensuring smooth control signals. The paper includes optimization results for several scenarios involving two and three HDPs together with various target poses, demonstrating the effectiveness of the proposed method.

**Keywords** Cooperative transport · Formation control · Nonholonomic vehicle · Optimal trajectory planning

## Zeitoptimale Trajektorienplanung für eine starre Formation nichtholonomer mobiler Plattformen

**Zusammenfassung** Dieser Artikel präsentiert eine Methode zur Planung zeitoptimaler Trajektorien für eine Formation mehrerer nichtholonomer

(Schwerlasttransport-) Plattformen (HDPs). Ziel ist dabei der kooperative Transport eines Objektes zu einer gewünschten Endposition. Der erste Abschnitt behandelt die mobilen Plattformen, während der zweite Abschnitt ein Konzept zur Trajektorienplanung liefert, das aus dem bekannten „Virtual Leader-Ansatz“ abgeleitet ist. Um den korrekten Transport der gemeinsamen Nutzlast zu gewährleisten, werden die Fahrzeuge individuell modelliert, was zu einem Formationsregelungsproblem führt. Das Ziel des Optimierungsprozesses besteht darin, eine Kostenfunktion zu minimieren, die Zeitoptimalität, glatte Steuersignale und die Formationssteifigkeit ausbalanciert. Das Optimalsteuerungsproblem (OCP) berücksichtigt die Kinematik der Fahrzeuge sowie ihre physikalischen Einschränkungen. Das Lösen erfolgt mit einem Mehrfach-Schieß-Verfahren, das die gewünschten Trajektorien inklusive der glatten Steuersignale für alle Fahrzeuge liefert. Der Artikel enthält zudem Optimierungsergebnisse für Szenarien mit zwei und drei HDPs und unterschiedlichen Zielpositionen, die die Wirksamkeit der vorgeschlagenen Methode demonstrieren.

**Schlüsselwörter** Kooperativer Transport · Formationsregelung · Nichtholonomes Fahrzeug · Optimale Trajektorienplanung

## 1 Introduction

Autonomous ground vehicles are crucial for automating logistics systems. Cooperative transport involving multiple agents can reduce the payload of individual mobile platforms while enhancing the maneuverability of the objects being transported. Formation control, initially explored in UAV applications, is now gaining attention in AGV applications. A comprehensive survey on the fundamental problem and

S. Schmidt (✉) · H. Gatringer · A. Mueller  
 Institute of Robotics, Johannes Kepler University Linz,  
 Altenberger Str. 69, 4040 Linz, Austria  
[simon.schmidt@jku.at](mailto:simon.schmidt@jku.at)

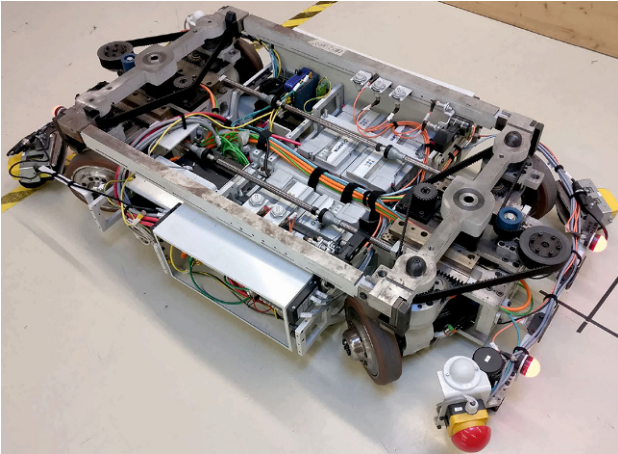


Fig. 1 Photo of the heavy duty platform

solution approaches can be found in [1]. The primary challenge lies in achieving virtual rigidity among the fleet of mobile robots, which is a key aspect of any formation control scheme. Two types of formations must be distinguished: fixed topology with switching geometry [2] and switching topology formations [3, 4]. Additionally, the rigidity property, particularly for 6D-agents like UAVs, needs to be investigated [5]. Formation control schemes assume that agents can execute any desired motion, but non-holonomic mobile robots have kinematic constraints that limit their movement compared to omnidirectional platforms. This paper addresses the underlying path planning problem, specifically using special autonomous ground vehicles as mobile agents (see Fig. 1). The considered nonholonomic vehicle possesses unique kinematics, enabling car-like steered longitudinal and lateral motion, pure rotation, and wheel alignment even while stationary.

There are various approaches to cooperative transport, one of which is the virtual leader method. In [6], this method is applied to develop a distributed model predictive controller (DMPC) for a formation of differential-driven vehicles. Additionally, an adaptation for a car-like vehicle is proposed as the basis for the optimal control problem (OCP). A similar DMPC for omnidirectional vehicles is discussed in [7], which also employs the virtual leader approach and compares it with an algebraic graph theory method. Both approaches enable tracking of predefined setpoints or trajectories for the formation but do not offer a trajectory planning method considering the formation's kinematics. On the other hand, [8] utilizes the virtual leader approach to design a trajectory tracking controller with collision avoidance for a formation of robots equipped with differential drives. The advantage of treating the vehicles individually and formulating the issue as a formation control problem is the flexibility to accommodate any number of vehicles, regardless of their kinematics.

This paper is organized as follows. At first, a brief overview of the HDP's hardware setup and kinematics along with a control strategy is given in Sect. 2. In Sect. 3, a method for planning time optimal trajectories for a single vehicle is proposed, which is then extended to a formation of multiple mobile platforms cooperatively transporting an object. Additionally, a special case for two vehicles is considered. Sect. 4 presents optimization results for the cases of two and three vehicles. The paper closes with a summary and an outlook for the realization of the planned trajectories on the HDPs in Sect. 5. Note that this paper is an extension of [9].

**Notation** Since this paper focuses solely on planar problems, positional vectors are denoted as  ${}_{\mathcal{K}}\mathbf{r} = ({}_{\mathcal{K}}x, {}_{\mathcal{K}}y)^T$ , where  $\mathcal{K}$  refers to the corresponding coordinate frame. For positional vectors expressed in the initial frame of reference,  $\mathcal{K} = I$ , the frame indication is neglected,  $\mathbf{r} = {}_I\mathbf{r} = (x_r, y_r)^T$ . Regarding orientations, the chosen frame of reference is irrelevant and is therefore also neglected.

## 2 A pseudo-omnidirectional heavy duty platform

### 2.1 Hardware

The HDP (Fig. 1) is a vehicle equipped with four wheels positioned at the corners of the chassis, each being driven by its own electric motor. It is designed to handle payloads weighing up to 1.5 tons. The HDP's chassis has dimensions of  $L = 1.18$  m in length,  $B = 0.55$  m in width and the wheel radii measure  $r = 0.125$  m. Moreover, the off-center distance of each wheel equals  $a = 0.11$  m. To differentiate between the front and rear parts of the vehicle, as well as the left and right sides (Fig. 2), the angular velocities of the wheels are denoted as  $\beta_{i,j}$ , where  $i$  can take the values of either  $l$  or  $r$  to represent the left or right side, and  $j$  can be either  $F$  or  $R$  to represent the front or rear. The motor-wheel units are mounted in such a way that they can rotate around an off-center point, with the corresponding steering angles being denoted as  $\alpha_{i,j}$ . Due to mechanical coupling, the front and rear wheels on each side  $i$  of the vehicle can only pivot in opposite directions,

$$\alpha_i := \alpha_{i,F} = -\alpha_{i,R} \quad i \in \{l, r\}. \quad (1)$$

Therefore, the vehicle is steered passively by appropriately driving the wheels. To enable proper control of the robot's motion, both the angular velocity and the steering angle of each wheel are measured. The HDP is also equipped with two additional motors located at the front and back, allowing it to lift the payload up to 0.3 m. However, in the considered two-dimensional scenario, this lifting functionality does not contribute to finding feasible trajectories for the vehicle and is thus not considered further.

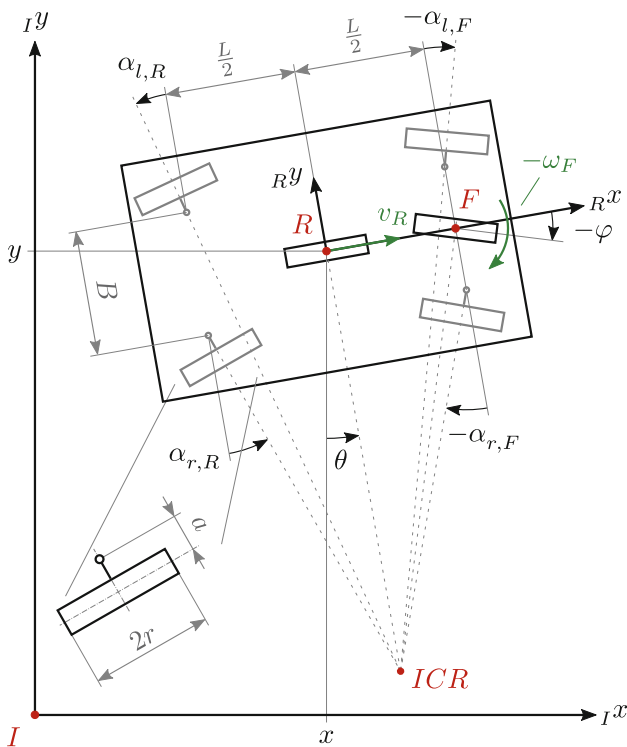


Fig. 2 Kinematics of the HDP

For the fundamental control of the system, hardware provided by *B&R Industrial Automation* is utilized. This includes a *X20 PLC* (Programmable Logic Controller) along with corresponding input and output modules. Two *ACOPOS* servo amplifiers are employed to drive three motors each. For comprehensive  $360^\circ$  monitoring of the vehicle's surroundings, two *Li-DAR* laser scanners are positioned at opposite corners of the chassis. The entire setup communicates through a *Powerlink* bus system.

## 2.2 Kinematics

To enable a wheeled vehicle to move without wheel slippage, it is necessary for the lateral directions of all wheels to intersect at a single point known as the *Instantaneous Center of Rotation* (ICR). The motion of the robot can then be interpreted as a rotation around this particular point. The mechanical coupling of the wheels restricts the ICR to the body fixed  ${}_R y$  axis as well as to  $\pm\infty$  on the  ${}_R x$  axis. Therefore, it is possible to distinguish between four different driving modes, illustrated in Fig. 3. In the *Steering* mode, the lateral directions of the wheels on the left and right sides do not intersect at a single point. Consequently, it is not possible to move the vehicle in this mode without wheel slippage, but it is still essential for transitioning between the other modes. In Fig. 3 this is illustrated by two ICRs for the left and right wheels. In contrast to the first mode, the remaining three modes enable the vehicle to drive in different ways. The

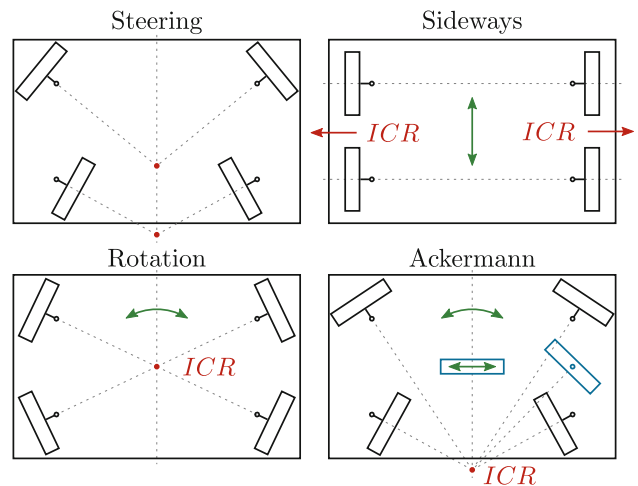


Fig. 3 The ICR for the four driving modes of the HDP

second mode, referred to as *Sideways*, permits lateral movement of the vehicle, resulting in a trivial linear motion along a straight line. When driving sideways, the ICR is positioned at the vehicle's longitudinal axis at infinity. In the third mode, the HDP can rotate around the geometric center of the wheels' mounting points, also leading to trivial motion. The final mode allows the vehicle to behave like an all-wheel-steered car, enabling nonlinear motion and facilitating tracking of more complex trajectories. This wheel alignment configuration is commonly known as *Ackermann Kinematics*. For this mode, the ICR can move along the robot's body fixed  ${}_R y$  axis and is only restricted by the physical steering angle limits. Due to these four driving modes, the HDP can be categorized as quasi-omnidirectional, as it needs to rotate before it can move in any desired direction. This paper focuses solely on the Ackermann mode, as the primary emphasis is not on planning trajectories with automated mode switching.

The kinematics of the HDP can be fully described using a set of minimal coordinates,  $\mathbf{z}(t) = (\mathbf{r}^T(t) \theta(t) \boldsymbol{\alpha}^T(t) \boldsymbol{\beta}^T(t))^T$ , which encompass the position  $\mathbf{r} = (x \ y)^T$  and orientation  $\theta$  of the chassis, as well as the steering angles  $\boldsymbol{\alpha} = (\alpha_l \ \alpha_r)^T$  and wheel angles  $\boldsymbol{\beta} = (\beta_{l,R} \ \beta_{l,F} \ \beta_{r,R} \ \beta_{r,F})^T$ . For brevity, time dependencies of the states, inputs and all variables depending on those are not further denoted. Each mode  $m$  is associated with a distinct set of minimal velocities  $\dot{\mathbf{s}}_m$ , which can be mapped to the time derivative of the minimal coordinates,  $\dot{\mathbf{z}} = \mathbf{H}_m(\mathbf{q})\dot{\mathbf{s}}_m$ . The number of minimal velocities for each mode is determined by the number of *nonholonomic* constraints, which impose restrictions solely on the vehicle's velocity, not its position.

The kinematics of the Ackermann mode can be represented by the bicycle model, featuring a fixed wheel at the center of the chassis and a steering wheel located between the HDP's two front wheels, as depicted in Fig. 2. Hence, this simplified model can be em-

ployed in the design of an appropriate path tracking and feed-forward controller for the HDP, as well as in the planning of suitable trajectories. The chosen model, representing the bicycle's kinematics,

$$\dot{\mathbf{q}} = \underbrace{\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{pmatrix}}_{\mathbf{H}(\mathbf{q})} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ \frac{1}{l} \tan\varphi & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_R \\ \omega_F \end{pmatrix} =: \mathbf{g}_1(\mathbf{q}) v_R + \mathbf{g}_2(\mathbf{q}) \omega_F, \quad (2)$$

involves four minimal coordinates,  $\mathbf{q} = (x \ y \ \theta \ \varphi)^T$ , and two minimal velocities,  $\dot{\mathbf{s}} = (v_R \ \omega_F)^T$ . For simplicity, the parameter  $l = L/2$  is introduced. Note that the interpretation of the kinematic model as a projection of the minimal velocities into the directions of the input vector fields  $\mathbf{g}_1(\mathbf{q})$ ,  $\mathbf{g}_2(\mathbf{q})$  in the state space  $\mathbb{R}^4$  will be relevant for trajectory planning in Sect. 3.

To control the HDP, given values for the bicycle model's steering angle  $\varphi$  and minimal velocities need to be converted to the necessary steering angles and angular wheel velocities for the HDP. The conversion of the steering angles is derived using the ICR condition,

$$\alpha_{llr}(\varphi) = \arctan\left(\frac{L \tan \varphi}{L \mp B \tan \varphi}\right). \quad (3)$$

The angular wheel velocities, corresponding to the input for the HDP's kinematic model, can be expressed by

$$\dot{\boldsymbol{\beta}} = \mathbf{U}(\varphi) \dot{\mathbf{s}} \quad \text{with} \quad \mathbf{U}(\varphi) = \begin{pmatrix} c_{v,l}(\varphi) & -c_{\omega,l}(\varphi) \\ c_{v,l}(\varphi) & c_{\omega,l}(\varphi) \\ c_{v,r}(\varphi) & c_{\omega,r}(\varphi) \\ c_{v,r}(\varphi) & -c_{\omega,r}(\varphi) \end{pmatrix} \quad (4)$$

using the abbreviations

$$c_{v,llr}(\varphi) = \frac{1}{Lr} \left( (L \mp B \tan \varphi) \sqrt{\frac{\tilde{c}_{v,llr}(\varphi)}{(L \mp B \tan \varphi)^2 \mp 2 \tan \varphi}} \right),$$

$$\tilde{c}_{v,llr}(\varphi) = (B^2 + L^2) \tan^2 \varphi \mp 2BL \tan \varphi + L^2 \quad \text{and}$$

$$c_{\omega,llr}(\varphi) = \frac{aL^2}{r(B^2 \cos^2 \varphi \pm 2BL \sin \varphi \cos \varphi - B^2 - L^2)}.$$

Moreover, to ensure proper steering of the vehicle, the correlation between the angular wheel and angular steering velocities,

$$\dot{\boldsymbol{\beta}} = \mathbf{V} \dot{\boldsymbol{\alpha}} \quad \text{with} \quad \mathbf{V} = \begin{pmatrix} -\frac{a}{r} & 0 \\ \frac{a}{r} & 0 \\ 0 & \frac{a}{r} \\ 0 & -\frac{a}{r} \end{pmatrix}, \quad (5)$$

is required. This expression can be derived considering the HDP's steering configuration ( $\dot{\mathbf{s}} = \dot{\boldsymbol{\alpha}}$ ,  $v_R = 0$ ,  $\omega_F = 0$ ) but is also evident, since (1) implies  $\dot{\alpha}_i = \dot{\alpha}_{i,F} = -\dot{\alpha}_{i,R}$  for  $i \in \{l, r\}$ .

### 2.3 Control of a single platform

In order to move the HDP along planned trajectories  $\mathbf{q}_d(t)$ , a suitable control strategy consisting of three components is implemented. A path tracking controller ensures that the vehicle follows the given path by providing desired minimal velocities  $\dot{\mathbf{s}}_c$ , which are then used in the low-level control to compute the required feedback torque  $\mathbf{M}_{FB}$ . Furthermore, a feed-forward controller incorporates the HDP's dynamics yielding a feed-forward torque  $\mathbf{M}_{FF}$ .

#### 2.3.1 Path tracking control

For path tracking control, there are several options available. In [10] a flatness based control concept is proposed, originally designed for the kinematic model of a car but applicable to the bicycle model and thus the HDP. This controller requires knowledge about the desired trajectory up to its second derivative. Another option is the *Stanley* controller [11], which only requires the desired trajectory up to the first derivative. It's important to note that the Stanley controller is primarily designed for forward driving and would need slight modifications to handle reverse driving, as the generated paths may require it. Furthermore, a (D)MPC could be implemented for each individual vehicle, following the approach outlined in [6]. This controller only requires the desired trajectory without any derivatives. However, implementing DMPC would necessitate communication between the robots, allowing them to consider other agents and respond effectively to potential disturbances. It should be emphasized that implementing such communication is a non-trivial task. Therefore, this control concept can be considered for future development, given the similarity between the optimization problem it is based on and the one used for trajectory planning in Sect. 3.

The chosen path tracking controller introduced in [12] computes the desired minimal velocities based on the vehicle's positional and orientational error with the former being expressed in the trajectory's current coordinate frame,

$${}_{s} \mathbf{e}_r = \mathbf{R}_{sI}(\underbrace{\mathbf{r}_d - \mathbf{r}}_{\mathbf{r}_e}), \quad (6)$$

using the rotation matrix

$$\mathbf{R}_{sI} = \begin{pmatrix} \cos \theta_d & \sin \theta_d \\ -\sin \theta_d & \cos \theta_d \end{pmatrix}. \quad (7)$$



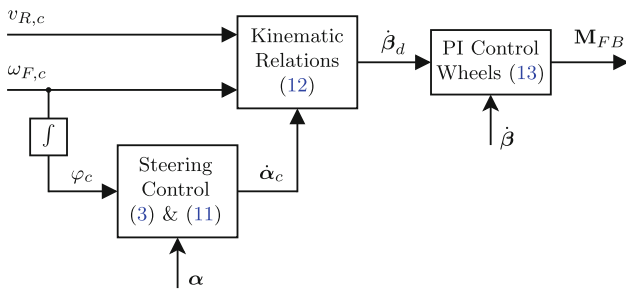


Fig. 4 Low-level control for the HDP

The error in the orientation, on the other hand, is independent of the coordinate frame, and reads  $e_\theta = \theta_d - \theta$ . Using these error definitions, the path tracking control law

$$\dot{\mathbf{s}}_c = \begin{pmatrix} v_{R,d} \cos e_\theta + k_x s e_x \\ \omega_{F,d} + v_{R,d} (k_y s e_y + k_\theta \sin e_\theta) \end{pmatrix} \quad (8)$$

results in an asymptotically stable closed loop behavior for controller parameters  $k_x, k_y, k_\theta > 0$ . This statement can be proven via *Lyapunov's indirect method* along with the respective *Lyapunov's function*

$$V = \frac{1}{2} (s e_x^2 + s e_y^2) + \frac{1 - \cos e_\theta}{k_y} \geq 0, \quad (9)$$

for  $|e_\theta| < \pi/2$ . Note that this concept also requires solely the desired trajectory, without any derivatives. The computed minimal velocities leading to the vehicle tracking its desired path are consequently handled in the HDP's low level control generating the required feedback motor torque.

### 2.3.2 Low-level control

The HDP's steering angles and angular wheel velocities are governed by respective cascaded controllers. Initially, the desired angular steering velocity  $\omega_{F,c}$ , provided by the path tracking controller, is integrated to obtain the desired steering angle  $\varphi_d$ . This steering angle is then transformed to the HDP's desired steering angles  $\alpha_{l/r,d}$  by making use of (3). Thus, the steering error  $\mathbf{e}_\alpha := \alpha_d - \alpha$  can be introduced. In order to achieve asymptotically stable steering control, it is essential that the related error dynamics take the form of

$$\dot{\mathbf{e}}_\alpha \stackrel{!}{=} -\mathbf{K}_{P,\alpha} \mathbf{e}_\alpha \Leftrightarrow \dot{\mathbf{e}}_\alpha + \mathbf{K}_{P,\alpha} \mathbf{e}_\alpha = 0, \quad (10)$$

with  $\mathbf{K}_{P,\alpha} > \mathbf{0}$ . Typically, while driving in the Ackermann mode, changes in the steering angle occur at a slow rate, allowing for the assumption  $\dot{\alpha}_d \approx \mathbf{0}$ . Therefore, the control law

$$\dot{\alpha}_c := \mathbf{K}_{P,\alpha} \mathbf{e}_\alpha, \quad (11)$$

is sufficient to ensure asymptotically stable steering error dynamics.

Next, the desired angular wheel velocities are computed by combining (4) and (5),

$$\dot{\boldsymbol{\beta}}_d = [\mathbf{U}(\varphi_c) \quad \mathbf{V}] \begin{pmatrix} \dot{\mathbf{s}}_c \\ \dot{\alpha}_c \end{pmatrix}. \quad (12)$$

To maintain these desired angular velocities, a Proportional-Integral (PI) controller is implemented, which yields the feedback torque,

$$\mathbf{M}_{FB} = \mathbf{K}_{P,\beta} (\dot{\boldsymbol{\beta}}_d - \dot{\boldsymbol{\beta}}) + \mathbf{K}_{I,\beta} \int (\dot{\boldsymbol{\beta}}_d - \dot{\boldsymbol{\beta}}) dt, \quad (13)$$

with  $\mathbf{K}_{P,\beta}, \mathbf{K}_{I,\beta} > \mathbf{0}$ .

The proposed low-level control scheme for the HDP is illustrated in Fig. 4, taking the desired minimal velocities from the path tracking controller, as well as the current steering angles and angular velocities, and providing in the feedback torque.

### 2.3.3 Feed-forward control

The feed-forward control considers the HDP's dynamics and generates a feed-forward torque based on the desired trajectory. In the case of an ideal and undisturbed vehicle, this torque would be sufficient to keep it on its desired path. Consequently, the feedback control just needs to compensate deviations due to the real vehicles imperfections as well as potential disturbances. Note that, especially when transporting heavy payloads, feed-forward control is crucial to ensure sufficient path tracking and in this particular case the formation's rigidity. The HDP's equations of motion are transformed to fit the bicycle model, resulting in

$$\mathbf{M}(\varphi) \ddot{\mathbf{s}} + \mathbf{g}(\varphi, \dot{\mathbf{s}}) = \mathbf{B}(\varphi) \mathbf{M}_{\text{Mot}}, \quad (14)$$

with the mass matrix  $\mathbf{M}(\varphi) \in \mathbb{R}^{2 \times 2}$ , the input matrix  $\mathbf{B}(\varphi) \in \mathbb{R}^{2 \times 4}$  and the vector  $\mathbf{g}(\varphi, \dot{\mathbf{s}}) \in \mathbb{R}^2$  summarizing the remaining terms related to friction as well as centripetal and coriolis effects. The feed-forward torque is then derived by minimizing the squared motor torques with the vehicle's dynamics being considered as an equality constraint by using a Lagrange parameter  $\lambda$ ,

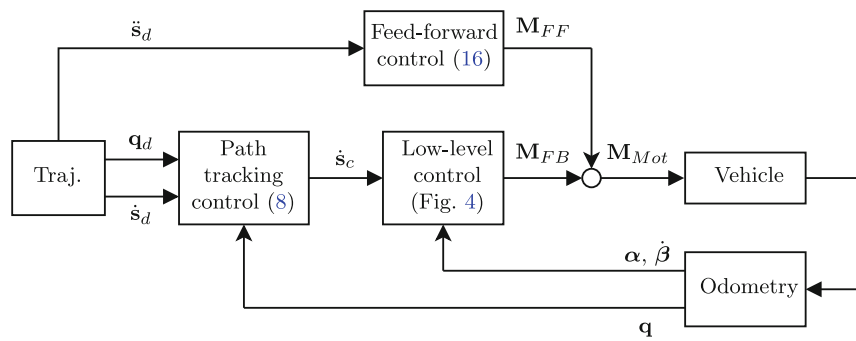
$$\underset{\lambda, \mathbf{M}_{\text{Mot}}}{\text{minimize}} \left( \mathbf{M}_{\text{Mot}}^T \mathbf{M}_{\text{Mot}} + \lambda^T (\mathbf{B}(\varphi) \mathbf{M}_{\text{Mot}} - \mathbf{M}(\varphi) \ddot{\mathbf{s}} + \mathbf{g}(\varphi, \dot{\mathbf{s}})) \right). \quad (15)$$

This minimization problem can then be solved using the right pseudo inverse  $\mathbf{B}^+(\varphi) = \mathbf{B}^T(\cdot) (\mathbf{B}(\cdot) \mathbf{B}^T(\cdot))^{-1}$ ,

$$\mathbf{M}_{FF} = \mathbf{B}^+(\varphi_d) (\mathbf{M}(\varphi_d) \ddot{\mathbf{s}}_d + \mathbf{g}(\varphi_d, \dot{\mathbf{s}}_d)). \quad (16)$$

Note that the feed-forward control requires the desired minimal velocities up to the first derivative,  $\dot{\mathbf{s}}_d$ . For further details regarding the vehicle's dynamics and the feed-forward control, see [13].

Fig. 5 Control scheme for the HDP



### 2.3.4 Overall control scheme

The comprehensive control scheme for the HDP consisting of the three discussed parts is depicted in Fig. 5. The odometry is responsible for estimating the vehicle's current state by utilizing sensor data, i.e. measurements of the steering angle and angular wheel velocities. While a detailed explanation of the odometry module is beyond the scope of this work, further information can be found in [13].

## 3 Time optimal formation trajectory planning

### 3.1 The bicycle model

Before addressing the problem of trajectory planning for a formation of multiple vehicles, a method for planning a path for a single vehicle, as depicted in Fig. 6, is proposed. The time optimal path, taking the mobile platform from an initial pose  $\mathbf{q}_I$  to a given final pose  $\mathbf{q}_F$ , will result from solving a time discrete OCP. Since the Feed-forward control requires the time derivative of the minimal velocities, the bicycle's kinematic model (2) is extended by introducing  $\mathbf{v} := \dot{\mathbf{s}}$  as the system's new input,

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{s}} \end{pmatrix} = \begin{bmatrix} \mathbf{H}(\mathbf{q}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{s}} \end{pmatrix} \quad \text{mit} \quad \mathbf{x} := \begin{pmatrix} \mathbf{q} \\ \dot{\mathbf{s}} \end{pmatrix}, \quad (17)$$

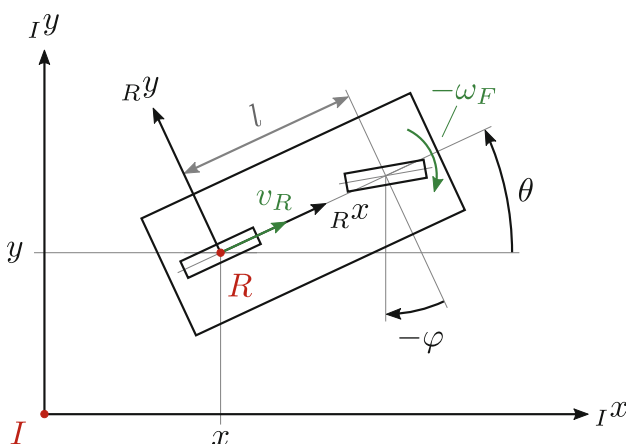


Fig. 6 Kinematics of the bicycle model

where  $\mathbf{I} \in \mathbb{R}^{2 \times 2}$  denotes the identity matrix. To incorporate this kinematic model into the discretized OCP, a fourth-order Runge-Kutta scheme (RK4) is employed to discretize the system,  $\mathbf{x}_{k+1} = \mathbf{f}_z(\mathbf{q}_k, \dot{\mathbf{s}}_k, \mathbf{v}_k)$ . Furthermore, the OCP takes into account the physical limitations of the HDP, i.e. limits for the steering angles  $\alpha_{\min} \leq \alpha \leq \alpha_{\max}$  and the angular wheel velocities  $\beta_{\min} \leq \beta \leq \beta_{\max}$ . The constraints on these variables are incorporated into the OCP using (3) and (4). Without loss of generality, constraints are also imposed on the remaining states of the model, allowing for additional restrictions on the position and orientation of the vehicle,  $\mathbf{q}_{\min} \leq \mathbf{q} \leq \mathbf{q}_{\max}$ .

Concatenating the relevant minimal coordinates  $\bar{\mathbf{q}} = (\mathbf{q}_1^T \dots \mathbf{q}_N^T)^T$  as well as new inputs  $\bar{\mathbf{v}} = (\mathbf{v}_0^T \dots \mathbf{v}_{N-1}^T)^T$  at all sample times  $t_k = 0, T/N, \dots, T$ , the OCP can be denoted as

$$\begin{aligned} \min \quad & J(T, \bar{\mathbf{q}}, \bar{\mathbf{v}}, \mathbf{q}_F) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}_z(\mathbf{q}_k, \dot{\mathbf{s}}_k, \mathbf{v}_k) \quad k = 0 \dots N-1 \\ & \mathbf{q}_0 = \mathbf{q}_I \\ & \mathbf{q}_N = \mathbf{q}_F \\ & \alpha_{\min} \leq \alpha_k \leq \alpha_{\max} \quad k = 1 \dots N-1 \\ & \beta_{\min} \leq \beta_k \leq \beta_{\max} \quad k = 0 \dots N-1 \\ & \mathbf{q}_{\min} \leq \mathbf{q}_k \leq \mathbf{q}_{\max} \quad k = 1 \dots N-1. \end{aligned} \quad (18)$$

The overall cost functional is composed of three terms  $J_i(\cdot)$  with respective weights  $w_i$ , reading

$$J(\cdot) = w_T J_T(T) + w_A J_A(\bar{\mathbf{q}}, \mathbf{q}_F) + w_S J_S(\bar{\mathbf{v}}). \quad (19)$$

The first term,  $J_T(T) = \sum_{k=0}^N \Delta t_k = T$  with  $\Delta t_k := t_{k+1} - t_k = T/N$ , minimizes the overall time and the third one,  $J_S(\bar{\mathbf{v}}) = \sum_{k=0}^{N-1} \mathbf{v}_k^T \mathbf{v}_k$ , ensures smooth input signals by minimizing the squared inputs  $\mathbf{v}$ .

The second term,  $J_A(\cdot) = \sum_{k=1}^N l_k(\mathbf{q}_k, \mathbf{q}_F)$  with

$$l_k(\cdot) = \sum_{i=1}^4 w_i (\mathbf{g}_i^T(\mathbf{q}_F) \Delta \mathbf{q}_k)^{k_i}, \quad (20)$$

determines how the setpoint  $\mathbf{q}_F$  is approached by projecting the deviations  $\Delta \mathbf{q}_k = \mathbf{q}_k - \mathbf{q}_F$  into the directions of the input vector fields  $\mathbf{g}_1(\mathbf{q})$ ,  $\mathbf{g}_2(\mathbf{q})$  and the subsequent Lie brackets

$$\mathbf{g}_3(\mathbf{q}) := [\mathbf{g}_1, \mathbf{g}_2](\mathbf{q}) = \frac{1}{l \cos^2 \varphi} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad (21)$$

$$\mathbf{g}_4(\mathbf{q}) := [\mathbf{g}_1, \mathbf{g}_3](\mathbf{q}) = \frac{1}{l(\sin^2 \varphi - 1)} \begin{pmatrix} \sin \theta \\ -\cos \theta \\ 0 \\ 0 \end{pmatrix}. \quad (22)$$

Note that the system's singularity at  $\varphi = \pm\pi/2$  is also present in these Lie-Brackets. This part of the cost functional plays a crucial role in the OCP for the MPC presented in [6]. In that case, it is essential for finding feasible trajectories without terminal deviations for a (more or less) arbitrary setpoint, due to the absence of terminal conditions. In contrast, when seeking feasible solutions of (18) this term is not mandatory, because of the here present terminal condition ( $\mathbf{q}_N = \mathbf{q}_F$ ). Nevertheless, the term is included in the overall cost functional, since it can be beneficial for finding more direct paths to the desired setpoint, although not necessarily optimal in terms of time. Furthermore, the OCP can be easily adapted to fit the requirements of an MPC approach.

From an illustrative point of view, the input vector fields correspond to the instantaneously controllable directions of the vehicle in the configuration space, i.e. driving straight and turning the front wheel. The results from the Lie brackets on the other hand correspond to the not instantaneously controllable directions, i.e. turning the whole vehicle and moving it sideways. Note that these are the two only linear independent Lie brackets of the input vector fields and subsequent Lie brackets, with  $\mathbf{g}_1(\mathbf{q}), \dots, \mathbf{g}_4(\mathbf{q})$  spanning the whole state space  $\mathbb{R}^4$ , since the vehicle is controllable. Due to the choice for the exponents  $k_1 = k_2 = 12$ ,  $k_3 = 6$ , and  $k_4 = 4$ , as proposed in [6], deviations in the not instantaneously controllable directions of the vehicle near the setpoint are penalized more than deviations in the remaining directions, with the ones in the lateral direction being penalized the most. In the case of a MPC, this specific choice for  $J_A(\cdot)$  and its exponents ultimately results in the vehicle approaching the desired setpoint along its  $Rx$  direction. For this use case, it enables a more direct setpoint approach, eliminating the need for extra distancing and reducing the overall space required. With the coefficients  $w_1, \dots, w_4$  the respective behavior can be further adjusted.

The overall cost functional  $J(\cdot)$  balances three factors: achieving a time-optimal trajectory, ensuring smooth input signals, and determining how the setpoint is approached. Thus, the coefficients  $w_T$ ,  $w_A$ , and  $w_S$  provide the flexibility to finely adjust the optimization result. Solving the proposed OCP yields a feasible, time optimal trajectory  $\mathbf{q}_k^*$  for the vehicle along with the required input signals  $\hat{\mathbf{p}}_k^*$  and  $\mathbf{v}_k^*$  for all sample times  $t_k^* = kT^*/N$ . It is important to note, that the resulting trajectory is time optimal only in

terms of the kinematic constraints, since the vehicle dynamics have not been considered in the OCP. Moreover, the time optimal part of the cost functional is balanced with the two other terms, which is another reason for why the trajectories are not purely time optimal.

To solve the OCP, the direct Multiple Shooting method [14] is employed with the  $N$  discretization intervals as shooting intervals. This method is implemented within the CASADI framework [15], and solved using the Interior Point method with the integrated IPOpt solver [16].

### 3.2 Formation trajectory planning for multiple vehicles

In this section, the proposed method for planning time optimal trajectories for a single vehicle will be adapted to a group of  $n \geq 2$  nonholonomic mobile platforms to cooperatively transport an object from an initial pose  $\hat{\mathbf{q}}_I$  to a desired final pose  $\hat{\mathbf{q}}_F$ . Extending the OCP (18) to multiple robots is straightforward. However, in order to ensure proper payload transport, the vehicles must also maintain their relative distances while following the resulting trajectories. This requirement needs to be mathematically formulated and incorporated into the OCP.

To simplify notation, the states  $\mathbf{q}_i$  of all vehicles are concatenated, resulting in vectors denoted as  $\tilde{\mathbf{q}} = (\mathbf{q}_1^T \dots \mathbf{q}_n^T)^T$  respectively. The individual discretized kinematic models of the robots are combined into one system of equations,  $\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{f}}_z(\tilde{\mathbf{q}}_k, \tilde{\mathbf{s}}_k, \tilde{\mathbf{v}}_k)$ . Furthermore, all relevant parameters that describe the payload's geometry, particularly its mounting points,  $\mathbf{r}_i$  for  $i = 1, \dots, n$ , are concatenated into the formation parameters  $\mathbf{p}$ .

The formation is described by its geometric center as well as the minimal coordinates of all mobile platforms. The formation center serves as a virtual leader and is represented by its pose  $\hat{\mathbf{q}} = (\hat{x} \ \hat{y} \ \hat{\theta})^T$  including its position and an assigned orientation. In Fig. 7, a formation consisting of three vehicles along with the virtual leader is depicted as an example. The leader's configuration can be estimated based solely on the other vehicles' positions,  $\hat{\mathbf{q}} = \hat{\mathbf{q}}(\tilde{\mathbf{q}})$ . The location of the virtual leader is given by

$$\hat{\mathbf{r}}(\tilde{\mathbf{q}}) = \frac{1}{n} \sum_{i=1}^n \begin{pmatrix} x_i \\ y_i \end{pmatrix}. \quad (23)$$

For estimating the leader's orientation, we consider the orientations of the virtual connection lines of the payload's mounting points. Without loss of generality, let  $\vartheta_{ij}$  be the angle relative to the  $jy$ -axis of the virtual line connecting the mounting points  $\mathbf{r}_i$  and  $\mathbf{r}_j$ ,

$$\vartheta_{ij} := \text{atan2} \left( \frac{x_i - x_j}{y_j - y_i} \right). \quad (24)$$

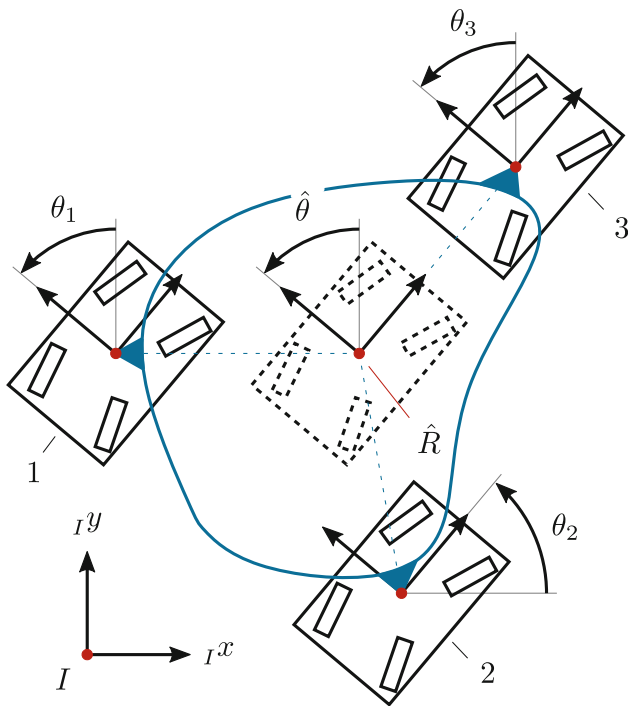


Fig. 7 Formation of  $n = 3$  vehicles and its representation by a virtual leader

By averaging the change of these angles compared to a reference configuration  $R$ , where  $\theta(\tilde{\mathbf{q}}_R) := 0$ , for all possible connecting lines, the estimation for the virtual leader's orientation is obtained,

$$\hat{\theta}(\tilde{\mathbf{q}}) = \frac{1}{|\mathcal{S}|} \sum_{(i,j) \in \mathcal{S}} (\vartheta_{ij} - \vartheta_{ij,R}). \quad (25)$$

Here,  $\mathcal{S} = \{(i, j) \in \{1, \dots, n\}^2 : j > i\}$  represents the set of all possible connection lines, with each line included exactly once.

Following that, the desired position of each robot can be expressed either with respect to the virtual leader,

$$\mathbf{r}_{i,d} = \mathbf{r}_{i,d}(\hat{\mathbf{q}}, \mathbf{p}), \quad (26)$$

or directly through the vehicles' minimal coordinates and the formation parameters using (23) and (25),

$$\mathbf{r}_{i,d} = \mathbf{r}_{i,d}(\hat{\mathbf{q}}(\tilde{\mathbf{q}}), \mathbf{p}) = \mathbf{r}_{i,d}(\tilde{\mathbf{q}}, \mathbf{p}). \quad (27)$$

In order to maintain the formation exactly, the positions of all robots have to coincide with their respective desired position,  $\mathbf{r}_i = \mathbf{r}_{i,d}(\tilde{\mathbf{q}}, \mathbf{p})$ . Including this rigidity condition in the extended OCP for all  $n$  mobile platforms as an additional equality constraint at every sample time  $t_k$  would ensure admissible relative trajectories of the robots. However, it would also lead to a problem that is hard to solve numerically.

To address this issue, considering that the robots will track the resulting trajectories using the proposed

tracking controller and therefore will have small tracking errors, the strict requirement of exact formation maintenance can be relaxed. Instead, the constraint can be modified to allow the vehicles' positions to be within a certain margin of error  $\epsilon_r > 0$  around their desired values,  $-\epsilon_r \leq \mathbf{r}_i - \mathbf{r}_{i,d}(\cdot) \leq \epsilon_r$ . This modification significantly improves the solvability of the OCP and reduces computation time, making it already sufficient for finding feasible trajectories for the given task. However, further improvement in solvability and computation time can be achieved by minimizing the margins between the desired and actual positions of the vehicles, in addition to just constraining them.

Finally, the general OCP for planning time optimal trajectories for multiple vehicles reads

$$\begin{aligned} \min_{\tilde{\mathbf{v}}} & J(T, \tilde{\mathbf{q}}, \tilde{\mathbf{v}}, \tilde{\mathbf{q}}_F, \mathbf{p}) \\ \text{s.t.} & \tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{f}}_z(\tilde{\mathbf{q}}_k, \tilde{\mathbf{s}}_k, \tilde{\mathbf{v}}_k) \quad k = 0 \dots N-1 \\ & \tilde{\mathbf{q}}_0 = \tilde{\mathbf{q}}_I \\ & \tilde{\mathbf{q}}_N = \tilde{\mathbf{q}}_F \\ & \tilde{\alpha}_{\min} \leq \tilde{\alpha}_k \leq \tilde{\alpha}_{\max} \quad k = 1 \dots N-1 \\ & \tilde{\beta}_{\min} \leq \tilde{\beta}_k \leq \tilde{\beta}_{\max} \quad k = 0 \dots N-1 \\ & \tilde{\mathbf{q}}_{\min} \leq \tilde{\mathbf{q}}_k \leq \tilde{\mathbf{q}}_{\max} \quad k = 1 \dots N-1 \\ & -\tilde{\epsilon}_r \leq \tilde{\mathbf{r}}_k - \tilde{\mathbf{r}}_{k,d} \leq \tilde{\epsilon}_r \quad k = 1 \dots N-1. \end{aligned} \quad (28)$$

In this formulation, various variables, including vehicle states, inputs, position vectors, and error limits, are concatenated using the tilde ( $\sim$ ) notation. The initial and final conditions for each robot are derived directly from the payload's initial  $\hat{\mathbf{q}}_I$  and final pose  $\hat{\mathbf{q}}_F$  using (26),

$$\mathbf{q}_{i,\mathcal{D}} = \begin{pmatrix} \mathbf{r}_{i,d}(\hat{\mathbf{q}}_{\mathcal{D}}, \mathbf{p}) \\ \hat{\theta}_{\mathcal{D}} \end{pmatrix}, \quad (29)$$

with  $\mathcal{D} \in \{I, F\}$ . Note that all robots are required to have the same orientation as the leader at the start and end of the trajectory to ensure that the formation can continue moving seamlessly after completing the computed trajectory. The OCP incorporates state and input restrictions for all agents, extending the analogous constraints from (18) to encompass all vehicles, while also incorporating the new formation constraints.

The overall cost functional,

$$J(\cdot) = w_T J_T(T) + w_A J_A(\tilde{\mathbf{q}}, \tilde{\mathbf{q}}_F) + w_S J_S(\tilde{\mathbf{v}}) + w_F J_F(\tilde{\mathbf{q}}, \mathbf{p}), \quad (30)$$

is now composed of four terms that account for different aspects of the optimization problem, in contrast to the three aspects from the single vehicle from Sect. 3.1. The first term,  $J_T(T) = T$ , leads to time optimality and remains the same for both single and multiple vehicle OCPs. The second term,

$$J_A(\cdot) = \sum_{i=1}^n \sum_{k=1}^N l_k(\mathbf{q}_{i,k}, \mathbf{q}_{i,F}), \quad (31)$$



is responsible for how the vehicles approach their setpoints,  $\mathbf{q}_{i,F}$  and is computed by applying the respective cost functional for a single vehicle,  $l_k(\cdot)$ , to all agents and summing up the results. The third term ensures smooth input signals and is obtained in the same way by simply extending the cost functional for one robot to all agents,

$$J_S(\tilde{\mathbf{v}}) = \sum_{i=1}^n \sum_{k=0}^{N-1} \mathbf{v}_{i,k}^T \mathbf{v}_{i,k}. \quad (32)$$

The fourth term in the cost functional minimizes the formation error  $\mathbf{e}_{r,k}(\tilde{\mathbf{q}}_k, \mathbf{p})$  and can be expressed as

$$J_F(\cdot) = \sum_{i=1}^n \sum_{k=1}^N \mathbf{e}_{r,k}^T(\cdot) \mathbf{e}_{r,k}(\cdot), \quad (33)$$

with  $\mathbf{e}_{r,k}(\tilde{\mathbf{q}}_k, \mathbf{p}) = \mathbf{r}_{i,k} - \mathbf{r}_{i,d}(\tilde{\mathbf{q}}_k, \mathbf{p})$  using (27). Again, the overall cost functional  $J(\cdot)$  is a trade off between now four aspects, being time optimality, smooth input signals, the way the setpoint is approached and additionally the magnitude of formation errors.

### 3.3 Trajectory planning for 2 vehicles with equal orientation

In addition to the general case of a formation with  $n \geq 2$  robots, we want to consider a special case with  $n = 2$  mobile platforms maintaining the same orientation as the payload throughout the entire transport,  $\theta_1 = \theta_2 = \hat{\theta}$  (see Fig. 8). Note that for the case  $n > 2$ , the vehicles must also be aligned laterally in order to enable the formation to move along a trajectory. Otherwise the vehicles' movements would be restricted significantly, allowing only for linear, longitudinal motion. This setup simplifies the mounting of the object

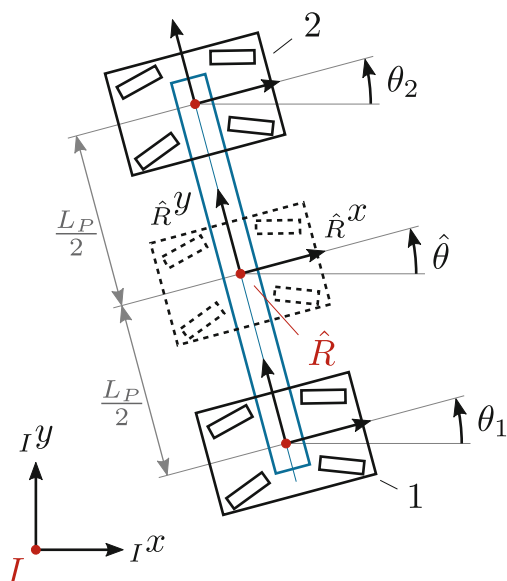


Fig. 8 Special case with  $n = 2$  vehicles

on the vehicles, as no rotatable connections are necessary.

The additional requirement is included in the OCP in two ways. First, all vehicle orientations are constrained, similar to their restricted positions, by enforcing  $-\epsilon_\theta \leq \theta_i - \hat{\theta}(\tilde{\mathbf{q}}) \leq \epsilon_\theta$ , with the error limit  $\epsilon_\theta > 0$ . Second, the formation part of the cost functional is extended by adding a second term that minimizes the robots' orientation errors at each time step,  $e_{\theta,k}(\cdot)$ , resulting in

$$J_F(\tilde{\mathbf{q}}, \mathbf{p}) = \sum_{i=1}^n \sum_{k=1}^N \mathbf{e}_{r,k}^T(\cdot) \mathbf{e}_{r,k}(\cdot) + e_{\theta,k}^2(\cdot), \quad (34)$$

with  $e_{\theta,k}(\tilde{\mathbf{q}}_k) = \theta_{i,k} - \hat{\theta}(\tilde{\mathbf{q}}_k)$ . The leader's orientation is computed using (25). Hence, the resulting OCP for the case of two vehicles transporting an object while maintaining the same orientation can be written as

$$\begin{aligned} \min_{\tilde{\mathbf{v}}} \quad & J(T, \tilde{\mathbf{q}}, \tilde{\mathbf{v}}, \tilde{\mathbf{q}}_F, \mathbf{p}) \\ \text{s.t.} \quad & \tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{f}}_z(\tilde{\mathbf{q}}_k, \tilde{\mathbf{s}}_k, \tilde{\mathbf{v}}_k) \quad k = 0 \dots N-1 \\ & \tilde{\mathbf{q}}_0 = \tilde{\mathbf{q}}_I \\ & \tilde{\mathbf{q}}_N = \tilde{\mathbf{q}}_F \\ & \tilde{\alpha}_{\min} \leq \tilde{\alpha}_k \leq \tilde{\alpha}_{\max} \quad k = 1 \dots N-1 \\ & \tilde{\beta}_{\min} \leq \tilde{\beta}_k \leq \tilde{\beta}_{\max} \quad k = 0 \dots N-1 \\ & \tilde{\mathbf{q}}_{\min} \leq \tilde{\mathbf{q}}_k \leq \tilde{\mathbf{q}}_{\max} \quad k = 1 \dots N-1 \\ & -\tilde{\epsilon}_r \leq \tilde{\mathbf{r}}_k - \tilde{\mathbf{r}}_{k,d} \leq \tilde{\epsilon}_r \quad k = 1 \dots N-1 \\ & -\tilde{\epsilon}_\theta \leq \tilde{\theta}_k - \tilde{\theta}_{k,d} \leq \tilde{\epsilon}_\theta \quad k = 1 \dots N-1. \end{aligned} \quad (35)$$

## 4 Optimization results

In this section the optimization results for a general case ( $n = 3$ ) as well as the special case ( $n = 2$ , Sect. 3.3) will be presented. The distance between all of the payload's mounting points is set as  $L_P = 1$  m, in both scenarios. For the case of  $n = 3$  the object resembles an equilateral triangle. The HDP's steering angles are limited to  $|\alpha_{l/r}| \lesssim 0.6 \Leftrightarrow |\varphi| \leq \pi/4$  and the limits for the angular wheel velocities are  $|\dot{\beta}_{l/r,R/F}| \leq 2$  1/s. The desired bounds for formation errors are chosen as  $\epsilon_x = \epsilon_y = 10^{-3}$  m for both cases and for the special case, the additional orientation bound is set to  $\epsilon_\theta = 10^{-3}$  rad.

As initial pose of the payload,  $\tilde{\mathbf{q}}_I = \mathbf{0}$  is chosen. For the setpoint approach part of the cost functional,  $J_A(\cdot)$ , the chosen weights are  $w_1 = 1$ ,  $w_2 = 0.1$ ,  $w_3 = 5$  and  $w_4 = 50$ . Furthermore, the weights for the overall cost functional read  $w_T = 10^4$ ,  $w_A = 1/N$ ,  $w_F = 2 \cdot 10^9/N$  and  $w_S = 10^9/N$ , where  $N = 500$  is the number of control intervals chosen for optimization.

The desired final pose for the general case ( $n = 3$ ) is  $\tilde{\mathbf{q}}_{F,1} = (L_P \ L_P \ \pi/2)^T$ . The set of all possible mounting point connection lines reads  $\mathcal{S} = \{(1,2), (1,3), (2,3)\}$ . Figure 9 shows the trajectories for all three vehicles as well as the payload's geometric center together with the longitudinal velocity  $v_R$  and the steering angle  $\varphi$  of the representative bicycle models. Note that the color for all vehicles and the payload is consistent

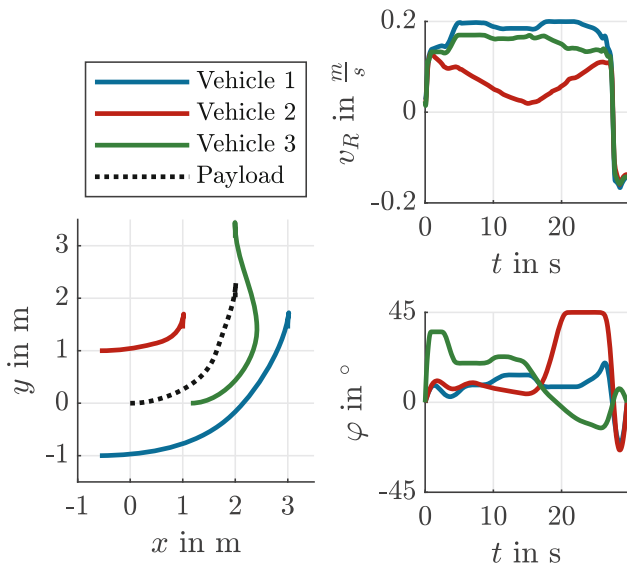


Fig. 9 Trajectories of the first scenario

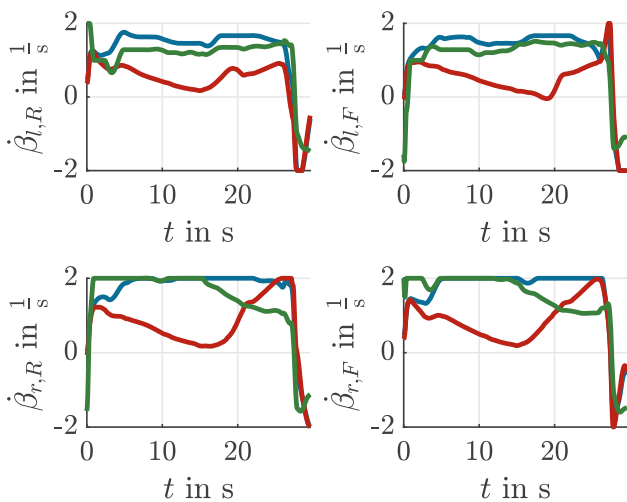


Fig. 10 Angular velocities of the wheels for  $\hat{\mathbf{q}}_{F,1}$

across all the plots. The optimal time is  $T_1^* = 29.61$  s and the optimization took about a minute on average. Figure 10 depicts the robots' inputs being the angular wheel velocities. Due to time optimality, at least one of these overall 12 input values is at its limit throughout most of the trajectory. Only towards the end, when approaching the setpoint, none of the input values reach their limits because of the setpoint approach part in the overall cost functional. For  $w_A = 0$ , one of the angular wheel velocities would be at its limit at all times, but the formation would require more space during the maneuver. Moreover, Fig. 11 illustrates the formation errors  $\mathbf{e}_r(\cdot) = (e_x \ e_y)^T$ , which always remain within their designated boundaries. Obviously, the chosen error limits could be reduced to a certain degree, which would result in more exact trajectories for the vehicles but for the cost of longer optimization times.

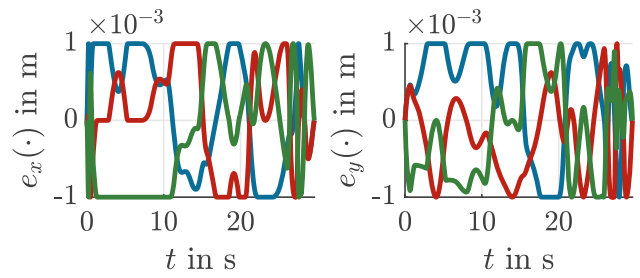


Fig. 11 Formation related errors of the first scenario

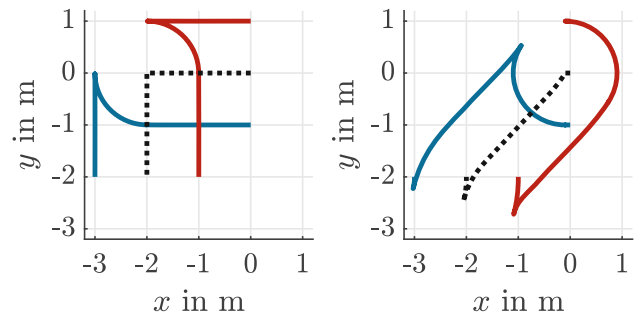


Fig. 12 Impact of  $J_A(\cdot)$  ( $n=2$ , left:  $w_A=0$ , right:  $w_A \neq 0$ )

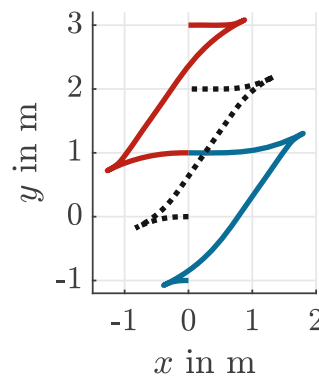


Fig. 13 Parallel parking scenario ( $n=2$ )

For the special case ( $n=2$ , Sect. 3.3), only the resulting trajectories are shown as the inputs and errors exhibit similar behavior to the general case. In this scenario, there is only one connecting line for the two robots' mounting points,  $\mathcal{S} = \{(1,2)\}$ . Figure 12 shows trajectories for the desired endpoint  $\hat{\mathbf{q}}_{F,2} = (-L_P \ -L_P \ -\pi/2)^T$ , comparing the results with and without the approach part in the cost functional. Consequently, the optimal time for the scenario with the approach part ( $w_A \neq 0$ ,  $T_{2,1}^* = 30.21$  s) is smaller than the one without it ( $w_A = 0$ ,  $T_{2,2}^* = 39.79$  s). As already discussed, the trajectory with the approach part poses a more direct approach. Figure 13 presents the resulting trajectory for a parallel park scenario for the case of  $n=2$ . The chosen endpoint reads  $\hat{\mathbf{q}}_{F,3} = (0 \ L_P \ 0)^T$  with an optimal time of  $T_3^* = 38.67$  s.

## 5 Summary and outlook

This paper proposes a method for generating time optimal trajectories for multiple HDPs in order to transport a shared payload from a initial to a given final pose. The next step involves implementing and testing these trajectories on physical robots. However, validating the absolute formation errors solely through the vehicle's odometry is insufficient due to its inaccuracies and limited ability, estimating only the position of a single robot relative to its initial configuration. Hence, an external measuring system, e.g. a laser tracker or motion capture system, has to be employed. Furthermore, a proper way of mounting the payload on the vehicles needs to be developed, allowing for minor displacement of the object relative to the robots mitigating non-rigidity of the formation. Future research will also focus on utilizing the remaining, unused drive modes of the HDP in addition to the Ackermann steering. Further, considerations such as collision avoidance and the payload's weight distribution can be incorporated into the optimization process.

**Acknowledgment** This work has been supported by the “LCM –K2 Center for Symbiotic Mechatronics“ within the framework of the Austrian COMET-K2 program.

**Funding** Open access funding provided by Johannes Kepler University Linz.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Oh K-K, Park M-C, Ahn H-S (2015) A survey of multi-agent formation control. *Automatica* 53:424–440
- Lin Z, Ding W, Yan G, Yu C, Giua A (2013) Leader–follower formation via complex laplacian. *Automatica* 49(6):1900–1906
- Rahimi R, Abdollahi F, Naqshi K (2014) Time-varying formation control of a collaborative heterogeneous multi agent system. *Rob Auton Syst* 62(12):1799–1805
- Han T, Guan Z-H, Chi M, Hu B, Li T, Zhang X-H (2017) Multi-formation control of nonlinear leader-following multi-agent systems. *ISA Trans* 69:140–147
- Michieletto G, Cenedese A, Franchi A (2016) Bearing rigidity theory in se (3). 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE, pp 5950–5955
- Rosenfelder M, Ebel H, Eberhard P (2021) Cooperative distributed model predictive formation control of non-holonomic robotic agents, pp 11–19 <https://doi.org/10.1109/MRS50823.2021.9620683>
- Ebel H, Eberhard P (2021) A comparative look at two formation control approaches based on optimization and algebraic graph theory. *Rob Auton Syst* 136:103686. <https://doi.org/10.1016/j.robot.2020.103686>
- Mastellone S, Stipanović DM, Graunke CR, Intlekofer KA, Spong MW (2008) Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. *Int J Rob Res* 27(1):107–126. <https://doi.org/10.1177/0278364907084441>
- Schmidt S, Gattringer H, Müller A (2023) Time optimal trajectory planning for a rigid formation of nonholonomic heavy duty platforms. *Proceedings of the Austrian Robotics Workshop (ARW), 2023*, pp 61–77
- Rudolph J (2021) Flatness-based Control: an Introduction, 1st edn. Shaker Verlag, Düren <https://doi.org/10.22028/D291-34336>
- Thrun S et al (2006) Stanley: The robot that won the darpa grand challenge. *J Field Robotics* 23(9):661–692. <https://doi.org/10.1002/rob.20147>
- Kanayama Y, Kimura Y, Miyazaki F, Noguchi T (1991) A stable tracking control method for a non-holonomic mobile robot. *IROS*, pp 1236–1241
- Pucher F, Gattringer H, Stöger C, Müller A, Single U (2019) Modeling and analysis of a novel passively steered 4wd mobile platform concept. In: Aspragathos NA, Koustoumpardis PN, Moulanianis VC (eds) *Advances in Service and Industrial Robotics*. Springer, Cham, pp 264–271
- Diehl M, Bock HG, Diedam H, Wieber P-B (2006) In: Diehl M, Mombaur K (eds) *Fast Direct Multiple Shooting Algorithms for Optimal Robot Control*. Springer, Berlin, Heidelberg, pp 65–93 [https://doi.org/10.1007/978-3-540-36119-0\\_4](https://doi.org/10.1007/978-3-540-36119-0_4)
- Andersson JAE, Gillis J, Horn G, Rawlings JB, Diehl M (2019) CasADi – A software framework for nonlinear optimization and optimal control. *Math Program Comput* 11(1):1–36. <https://doi.org/10.1007/s12532-018-0139-4>
- Wächter A, Biegler LT (2006) On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Math Program* 106:25–57

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Simon Schmidt**, received a BSc. degree in Mechatronics from Johannes Kepler University Linz (JKU) in 2021 and is currently finishing his master's degree in Mechatronics at the JKU. His research is focused on model-based and optimal control of mechatronical systems and (co-operative) mobile robotics.



**Hubert Gatringer**, received a Dipl.-Ing. degree in Mechatronics and the Dr. tech. (Ph.D.) degree in Technical Sciences from Johannes Kepler University Linz (JKU) in 2001 and 2006, respectively. In 2011 he became Associate Professor with the Institute of Robotics at JKU. His research is focused on modeling and control of flexible link/joint robotics systems and walking machines, optimal path planning, kinematically redundant industrial robots.



**Andreas Mueller**, obtained diploma degrees in mathematics (1997), electrical engineering (1998), and mechanical engineering (2000), and a PhD in mechanics (2014). He received his habilitation in mechanics (2011) and is currently professor in robotics at the Johannes Kepler University, Linz, Austria. His current research interests include holistic modeling, model-based and optimal control of mechatronic systems, redundant robotic systems, parallel kinematic machines, biomechanics, and computational dynamics.