# Occupancy Grid based Distributed Model Predictive Control of Mobile Robots*

Mohamed W. Mehrez[1], Tobias Sprodowski[2], Karl Worthmann[3], George K.I. Mann[1],
Raymond G. Gosine[1], Juliana K. Sagawa[2,4] and Jürgen Pannek[2]

*Abstract*— **In this paper, we introduce a novel approach of reducing the communication load in discrete time Distributed Model Predictive Control (DMPC) schemes of mobile robots. In a given group of robots, our approach is based on projecting the prediction states of each group member into a grid. This results in an occupancy grid prediction which is later communicated as tuples of integer values instead of floating point values. The proposed approach has the practical advantages of performing optimization in the continuous setting while conducting communication in the discrete setting. The discretization reduces the communication load significantly. The introduced approach is explored and evaluated numerically when applied to holonomic as well as non-holonomic robotic groups.**

## I. Introduction

Formation control of mobile robots has attracted significant interest in the last decade due to its potential application areas, e.g. patrolling missions [1], search and rescue operations [2], and situational awareness [3]. The control objective is to coordinate a group of robots to first form, and then maintain a prescribed formation. This can be done either in a cooperative or non-cooperative manner, see, e.g. [4] or [5]. In the latter case, each robot has its own target with respect to dynamic (e.g. collision avoidance) or static constraints (e.g. restrictions of the operating region).

Most of formation control approaches presented in the literature can be categorized under virtual structure, behavior-based, and leader follower [6]. In the virtual structure framework, a robotic team is considered as a rigid body and the individual robots are treated as points of this body. Thus, the overall motion of the formation defines the motion of the individual robots [7], [8]. In behavior-based control, we distinguish several kinds of behaviors which are assigned for the whole formation, e.g. formation keeping (stabilization), goal seeking, and obstacle avoidance [9]. Solution methods used in behavior-based control include motor scheme control [10], null-space-based behavior control [11], and potential fields functions [12]. The basic approach of the leader follower structure is that a follower is assigned to track another vehicle with an offset. Here, proposed solution methods include feedback linearization [13], backstepping control [14], and sliding mode control [15].

In this paper, we utilize a Model Predictive Control (MPC) scheme to stabilize a group of autonomous non-cooperative vehicles (formation stabilization) while avoiding collisions. In MPC, we first obtain a measurement of the state of the system, which serves as a basis for solving a finite horizon optimal control problem (OCP). This results in a sequence of future control values. Then, the first element of the computed sequence is applied before the process is repeated at the next sampling instant. Within this setting, the cost function of the OCP allows us to encode a control objective and to measure the performance of the closed loop. Moreover, static and dynamic constraints can be directly taken into account [16]. MPC has been used in several studies considering formation control, see, e.g. [17]–[23]. Here, we consider a distributed implementation of MPC, in which the system is split into subproblems regarding a single robot each. The robots solve their own (local) OCP and communicate with the other robots to avoid collisions. This information exchange is necessary since the strong coupling among the robots may render the control task infeasible in decentralized control (no communication), see, e.g. [24]. Here, the individual optimization tasks are executed in a fixed order similar to the method proposed by Richards and How [25], [26].

To formulate the coupling constraints, previous studies were based on the communication of predicted trajectories, cf. [27]. In contrast, we first partition the operating region into a grid. Then, the predicted trajectories are projected onto the grid, which serves as quantization of the communication data. Based on this information exchange, each robot formulates suitable coupling constraints. Utilizing the occupancy grid contributes to reducing bandwidth limits and congestion issues since a more compact data representation (integers instead of floating point values) is employed. In summary, the optimization is performed in a continuous set to make use of sensitivity information while the communication is conducted in a discrete set to reduce the transmission load. To construct the constraints, on the first hand we use the infinity norm and evaluate the closed-loop-performance with derivative-free optimization methods. On the other hand we

*M.W. Mehrez, K. Worthmann, and J. Pannek are supported by the Deutsche Forschungsgemeinschaft, Grant WO 2056/1. M.W. Mehrez, G.K.I. Mann, and R.G. Gosine are supported by Natural Sciences and Engineering Research Council of Canada (NSERC), the Research and Development Corporation (RDC), C-CORE J.I. Clark Chair, and Memorial University of Newfoundland.

[1] M.W. Mehrez, G.K.I. Mann, and R.G. Gosine are with Intelligent Systems Lab, Memorial University of Newfoundland, St. John's, NL, Canada. [m.mehrez.said,gmann,rgosine]@mun.ca

[2] T. Sprodowski, J.K. Sagawa and J. Pannek are with Department of Production Engineering, University of Bremen and BIBA Bremer Institut für Produktion und Logistik GmbH, Bremen, Germany. [spr,sag,pan]@biba.uni-bremen.de

[3] K. Worthmann is with Institute for Mathematics, Technische Universität Ilmenau, Ilmenau, Germany. karl.worthmann@tu-ilmenau.de

[4] J.K. Sagawa is with Department of Production Engineering, Federal University of São Carlos, Brazil. juliana@dep.ufscar.br

consider squircles as an approximation for a grid cell to be able to use gradient-based algorithms for optimization. Furthermore, we evaluate the minimal horizon length for both settings in holonomic and non-holonomic robots simulations such that a desired closed-loop performance is met.

We also present a criterion to define a minimum width for the grid cells, as well as an analysis of the influence of this parameter on the communication effort and total cost.

The paper is organized as follows: we first introduce the grid generation and present the distributed MPC scheme in Section II. Then, we derive a suitable representation of the collision avoidance constraints. Afterwards, we investigate the proposed method by means of numerical simulations for holonomic as well as non-holonomic robots in Section IV, before conclusions are drawn.

**Notation**: $\mathbb{R}$ and $\mathbb{N}$ denote the real and natural numbers, respectively. $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ represents the non-negative integers and $\mathbb{R}_{\geq 0}$ the non-negative real numbers. We abbreviate a set $\{0, 1, \cdots N\}$ with $[0 : N]$. Moreover, for a vector $x \in \mathbb{R}^n$, $n \in \mathbb{N}$, the infinity norm $\|\cdot\|_\infty$ is defined as $\|x\|_\infty := \max\{|x_i| : i \in [1 : n]\}$.

## II. Problem Setting

Firstly, we present the dynamics of the considered robots and a respective grid representation of the operating region. Secondly, we introduce the optimal control problem (OCP) of each subsystem. Finally, we integrate these components in a Distributed Model Predictive Control (DMPC) scheme.

### A. System Definition and Grid Generation

The control objective is to stabilize a group of $P$ subsystems, $P \in \mathbb{N}$, to target equilibria while avoiding collisions. Each subsystem is considered as part of a multi-agent system with state $z_p \in Z \subset \mathbb{R}^3$, $p \in [1 : P]$. For the $p$th subsystem, the discrete time dynamics are given by

$$z_p(n+1) = f(z_p(n), u_p(n)) \tag{1}$$

with vector field $f : \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^3$ and $n$ denotes the current time step. The state $z_p$ contains positional and orientation coordinates, i.e. $z_p = (x_p, y_p, \theta_p)^\top$. The control $u_p \in U \subset \mathbb{R}^2$ is given by $u_p = (v_p, \omega_p)^\top$. $v_p$ is the linear speed of the robot and $\omega_p$ is the angular orientation for holonomic and the angular speed for non-holonomic robots. Each robot has an initial state $z_p(0)$ and a reference $z_p^*$. State and control constraints are incorporated in

$$Z := [-\bar{x}, \bar{x}] \times [-\bar{y}, \bar{y}] \times \mathbb{R}, \qquad U := [-\bar{v}, \bar{v}] \times [-\bar{\omega}, \bar{\omega}]$$

for $\bar{x}, \bar{y}, \bar{v}, \bar{\omega} > 0$.

We partition the set $[-\bar{x}, \bar{x}] \times [-\bar{y}, \bar{y}]$ into a grid of squared cells, see Fig. 1. Each cell has a cell width $c$, where $c$ satisfies $2\bar{x} = a_{\max} c$ and $2\bar{y} = b_{\max} c$ with $a_{\max}$, $b_{\max} \in \mathbb{N}$. Therefore, each cell in the grid has a unique index $(a, b) \in \mathcal{G}$ where $\mathcal{G} := [0 : a_{\max} - 1] \times [0 : b_{\max} - 1] \subset \mathbb{N}_0^2$. Then, a location $(x, y)$ can be mapped to the discrete set $\mathcal{G}$ using the quantization $q : Z \to \mathcal{G}$ defined as

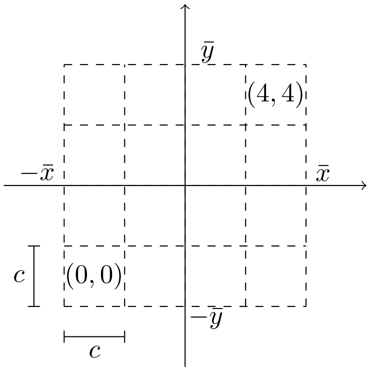

Fig. 1. Example of a discretized set $\mathcal{G}$ with $a_{\max} = b_{\max} = 5$

$$q(z) = (a, b) := \left( \left\lfloor \frac{x + \bar{x}}{c} \right\rfloor, \left\lfloor \frac{y + \bar{y}}{c} \right\rfloor \right). \tag{2}$$

Note that a location $(x, y)$ can be also expressed by a unique single index instead of $(a, b)$ by utilizing a full enumeration of the squared cells.

### B. Distributed Model Predictive Control

We first present the notation requirements to set up the DMPC algorithm. Thereafter, we define our quantized trajectory, communication scheme and cost objective to compose the optimal control problem (OCP).

For a finite control sequence

$$u_p = (u_p(0), u_p(1), \ldots, u_p(N-1))$$

and an initial value $z_p^0$, the predicted state trajectory of robot $p$ over prediction horizon $N \in \mathbb{N}$ is given by

$$z_p^u\left(\cdot; z_p^0\right) := \left(z_p^u\left(0; z_p^0\right), z_p^u\left(1; z_p^0\right), \ldots, z_p^u\left(N; z_p^0\right)\right).$$

Here, we removed the subscript $p$ from $u_p$ in $z_p^u\left(\cdot; z_p^0\right)$ to simplify our notation. The trajectory together with the introduced quantization $q(\cdot)$ allows us to define the occupancy grid $\mathcal{I}_p(n) \in (\mathbb{N}_0 \times \mathcal{G})^{N+1}$ at time $n$ as

$$\begin{aligned} \mathcal{I}_p(n) &:= \left(n + k, q\left(z_p^u\left(k; z_p^0\right)\right)\right)_{k \in [0:N]} \\ &=: \left(n + k, a_p^u(k; z_p^0), b_p^u(k; z_p^0)\right)_{k \in [0:N]}. \end{aligned}$$

where $a_p^u(k; z_p^0)$ and $b_p^u(k; z_p^0)$ denote the quantized state. While each robot $p$ sends only one such package $\mathcal{I}_p(n)$, it collects all received occupancy grid predictions in the information $i_p$, i.e.

$$i_p(n) := \left(\mathcal{I}_1(n), \ldots, \mathcal{I}_{p-1}(n), \mathcal{I}_{p+1}(n), \ldots, \mathcal{I}_P(n)\right).$$

Later, we use this information to construct the coupling constraints induced by robots $q \in Q := [1 : P] \setminus \{p\}$. The main motivation of communicating the occupancy grid prediction is to reduce the communication load.

Here, we utilize a distributed model predictive control scheme, which implements a local controller for each robot $p$. To this end, for a given reference state $z_p^*$ we choose the (local) control objective using stage costs $\ell_p : Z \times U \to \mathbb{R}_{\geq 0}$.

$\ell_p$ should be positive definite with respect to $z_p^*$. Then, for an initial condition $z_p^0 = z_p(n)$ and given information $i_p(n)$, the controller solves a finite horizon optimal control problem with prediction horizon $N$. At every timestep $n$, each robot $p$ has to minimize the costs by solving the following OCP:

$$\underset{u_p}{\operatorname{argmin}}\, J_p^N\left(u_p; z_p^0, i_p\left(n\right)\right) := \sum_{k=0}^{N-1} \ell\left(z_p^u\left(k; z_p^0\right), u_p\left(k\right)\right) \tag{3}$$

subject to the constraints

$$\begin{aligned} z_p^u(k+1; z_p^0) = f(z_p(k; z_p^0), u_p(k)), &\quad k \in [0 : N-1], \\ u_p(k) \in U, &\quad k \in [0 : N-1], \\ -G\left(z_p^u\left(k; z_p^0\right), i_p\left(n\right)\right) \leq 0, &\quad k \in [0 : N], \\ z_p^u\left(k; z_p^0\right) \in Z, &\quad k \in [0 : N]. \end{aligned}$$

As a result, an optimal control sequence

$$u_p^* = \left(u_p^*\left(0\right), \ldots, u_p^*\left(N-1\right)\right) \tag{4}$$

is obtained which minimizes the cost function (3). The corresponding value function $V_p^N : Z \times (\mathbb{N}_0 \times \mathcal{G})^{(P-1)(N+1)} \to \mathbb{R}_{\geq 0}$ is defined as

$$V_p^N\left(z_p^0, i_p\left(n\right)\right) := J_p^N(u_p^*; z_p^0, i_p). \tag{5}$$

The detailed construction of the collision avoidance constraints $G$ will be explicated in the ensuing Section III. Using this formulation of the OCP, the DMPC algorithm is defined as follows:

**Algorithm 1** DMPC-algorithm for the overall system

```
1: Obtain and communicate admissible control sequences $u_p$ for initial values $z_p\left(0\right)$ for all $p \in [1 : P]$
2: for $n = 0, 1, \ldots$ do
3:     for $p$ from 1 to $P$ do
4:         Receive $i_p\left(n\right)$
5:         Solve OCP (3) and Apply $u_p^*\left(0\right)$
6:         Broadcast $\mathcal{I}_p\left(n\right)$
7:     end for
8: end for
```

In the generic case, the calculation of an admissible set of controls for all subsystems in the initialization of Algorithm 1 is particularly demanding. Known proofs for recursive feasibility of the scheme with and without terminal conditions [25]–[27] depend on the success of the latter, which is then propagated by the order of subsystems. In our setting, however, the trivial choice $u_p \equiv 0$ is always admissible which allows to conclude recursive feasibility and stability under well known conditions.

## III. Formulation of the Constraints

In this section, we derive an appropriate cell width $c$ based on the dynamics of the system and specify a safety distance between two robots. Thereafter, we expound two methods to formulate the collision avoidance constraints $G$ in the OCP (3): the former based on the infinity norm and the latter relies on a squircular approximation.

To formulate the coupling constraints, we first define the backward mapping $f_c : \mathbb{N}_0 \times \mathcal{G} \to \mathbb{R}^2$ given by

$$f_c\left((n, a, b)\right) = \underbrace{\left(c\left(a + 0.5\right) - \bar{x}, c\left(b + 0.5\right) - \bar{y}\right)^\top}_{=:(x^c, y^c)_{(a,b)}} \tag{6}$$

which transforms a cell index $(a, b) \in \mathcal{G}$ to the corresponding cell center location $(x^c, y^c)_{(a,b)} \in [-\bar{x}, \bar{x}] \times [-\bar{y}, \bar{y}]$.

### A. Grid Generation and Safety Margins

The cell width $c$ has to be chosen large enough such that a cell can not be skipped during one sampling step $T$. We assume that the dynamics (1) is Lipschitz continuous with respect to its first argument $z_p$, i.e. there exists $L > 0$ satisfying

$$\|f(z_p, u_p) - f(\hat{z}_p, u_p))\| \leq L\|z_p - \hat{z}_p\| \quad \forall\, u_p \in U$$

for all $z_p$, $\hat{z}_p \in Z$. Then, we obtain the minimum cell width

$$\underline{c} = \frac{2\bar{x}}{\left\lceil \frac{2\min\{\bar{x}, \bar{y}\}}{e^{LT}} \right\rceil},$$

where the denominator represents the upper bound on the number of equidistant quantization intervals ensuing that with a larger sampling time the minimum cell size increases. Furthermore the lipschitz continuity holds for $e^{LT} \ll \infty$. To guarantee a minimum distance $d_{\min}$ between the subsystems and to avoid overlapping of predicted trajectories, see Fig. 2 (left) for an illustration, we additionally inflate each (occupied) cell. The inflation margin is given by $\max\{d_{\min}, \bar{v} \cdot T\}$. Furthermore, we have to ensure that constraint violations due to sampling cannot occur. Here, the worst case is an intermediate step with an angle $\frac{\pi}{4}$ illustrated in Fig. 2 (right). Therefore, cells are further inflated to compensate for constraint violation due to sampling by $\bar{v}T\cos(\pi/4)/2$. Thus, the safety margins of an occupied cell are specified by a square of side $\Psi$ given by

$$\Psi := c + 2\max\{d_{\min}, \bar{v}T\} + \bar{v}T\cos(\pi/4) + \epsilon$$

where $\epsilon \ll 1$ is a numerical safety margin.

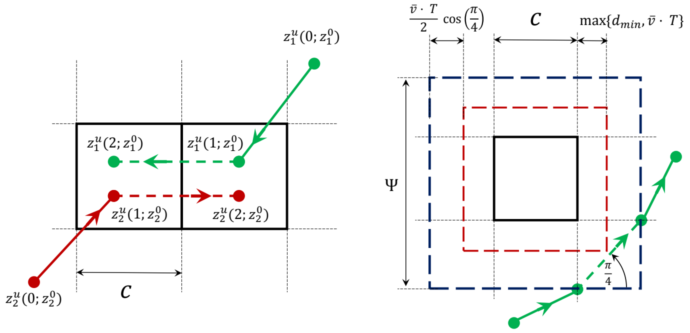

Fig. 2. Left: overlapping of prediction trajectories. Right: cell inflation to avoid violation of a state constraint due to discretization.

### B. Constraint Construction

As we obtain the information from the other robots $i_p$, we utilize the infinity norm $\|\cdot\|_\infty$ to define the constraint function $g_{p,q}(k) = g\left(z_p^u\left(k; z_p^0\right), \mathcal{I}_q(n)(k)\right)$ for robot $p$ induced by robot $q$ via

$$-\left\|\begin{pmatrix} x_p(k) \\ y_p(k) \end{pmatrix} - f_c\left(\mathcal{I}_q(n)(k)\right)\right\|_\infty + \frac{\Psi}{2}$$

with $k \in [0:N]$, $q \in [1:P-1]$, and $\mathcal{I}_q(n)$ representing the information of the $q$th robot in $i_p(n)$.

As the infinity-norm is not differentiable, we have to utilize a derivative-free optimization method upon implementation. A derivative-free optimizer is, in general, more time consuming than a derivative-based one, see, e.g. [28]. Therefore, we present a differentiable approximation of the cell constraints using squircles [29]. A squircle, which is a special case of a superellipsoid, is a geometric shape between a square and a circle. The polar equations of a squircle are given by

$$x(\phi) = 2^{-3/4}h \cdot |\cos(\phi)|^{\frac{1}{2}} \cdot \mathrm{sgn}(\cos(\phi)),$$
$$y(\phi) = 2^{-3/4}h \cdot |\sin(\phi)|^{\frac{1}{2}} \cdot \mathrm{sgn}(\sin(\phi))$$

with $\phi \in [0, 2\pi)$. $h$ is the width of the square, which a squircle is approximating from the outside. Now, the coupling constraints $g_{p,q}(k)$ of the $p$th robot can be formulated by using the Euclidean norm as[1]

$$-\left\|\begin{pmatrix} x_p(k) \\ y_p(k) \end{pmatrix} - f_c\left(\mathcal{I}_q(n)(k)\right)\right\| + \frac{\Psi\sqrt{|\cos(\beta)|+|\sin(\beta)|}}{2^{3/4}}$$

with $k \in [0:N]$ and $q \in [1:P-1]$. The angle $\beta$ is calculated via

$$\beta = \arctan\left(\frac{y_p(k) - y_i^c(k)}{x_p(k) - x_i^c(k)}\right),$$

where $(x_i^c(k), y_i^c(k))$ are obtained via the mapping (6) given the predictions $\mathcal{I}_q(n)(k)$ where $q$ and $k$ are as denoted above.

Finally, we construct the constraint $G(z_p^u\left(k; z_p^0\right), i_p(n))$ of robot $p$ by combining $g_{p,q}(k)$, $q \in [1:P-1]$ and $k \in [0:N]$ as

$$\Big(\left(g_{p,1}(k)\right)_{k\in[0,N]}, \ldots, \left(g_{p,p-1}(k)\right)_{k\in[0,N]}, \ldots,$$
$$\left(g_{p,p+1}(k)\right)_{k\in[0,N]}, \ldots, \left(g_{p,P}(k)\right)_{k\in[0,N]}\Big).$$

Hence, the OCP (3) is subject to the coupling constraints $G$ characterizing the collision avoidance objective.

## IV. Numerical Experiments

In this section, we explore the performance of Algorithm 1 through numerical simulations of holonomic as well as non-holonomic mobile robots. To this end, we consider a group of 4 mobile robots ($P = 4$) and set the state constraints $Z := [-6(\mathrm{m}), 6(\mathrm{m})]^2 \times \mathbb{R}$. The minimum distance to be ensured between the robots is chosen as $d_{\min} = 0.5$ (m). The linear speed constraints $\bar{v}$ is set to $\bar{v} = 1$ (m/s) while the sampling time is set to $T = 0.5$ (seconds). This leads to a minimum cell width $\underline{c} = 0.5$ (m), see Subsection III-A. We perform simulations with different cell widths $c \in \{0.5, 1, 1.5, 2\}$. Furthermore, we investigate the performance of Algorithm 1 in terms of communication effort as well as accumulated costs:

- The communication effort is given by

$$K := \sum_{n=0}^{n_\#} \sum_{p=1}^{P} \#\mathcal{I}_p^c|_n, \tag{7}$$

i.e. the number of broadcasted tuples among all robots over the closed loop simulation.

- The accumulated costs are given by

$$V^N\left(z_r^0, i_r(0)\right)_{r \in P} := \sum_{n=0}^{n_\#} \sum_{p=1}^{P} V_p^N\left(z_p^0, i_p(n)\right). \tag{8}$$

In both cases, $n_\#$ denotes the number of iterations of the whole simulation. The robots initial conditions as well as references are chosen as shown in Table I. Closed loop simulations are executed until all robots meet the condition

TABLE I

Initial Conditions and References of the Robots

| System ($i$) | Initial Conditions ($z_i^0$) | References ($z_i^*$) |
|---|---|---|
| 1 | $(4.5, 4.5, -3\pi/4)^\top$ | $(-4.5, -4.5, \pi)^\top$ |
| 2 | $(-4.5, 4.5, -\pi/4)^\top$ | $(4.5, -4.5, 0)^\top$ |
| 3 | $(4.5, -4.5, 3\pi/4)^\top$ | $(-4.5, 4.5, \pi)^\top$ |
| 4 | $(-4.5, -4.5, \pi/4)^\top$ | $(4.5, 4.5, 0)^\top$ |

$$\|z_p(n) - z_p^*\| \leq 0.01. \tag{9}$$

Holonomic robots simulations were conducted utilizing the infinity norm in constructing the coupling constraints as shown in Section III-B. Therefore, the derivative-free optimization method COBYLA (Constrained Optimization by Linear Approximations) [30] was employed in the NLopt package [31]. Then, all simulations were implemented in C++. On the other hand, the non-holonomic robots simulations were performed utilizing squircles to compose the coupling constraints. Thus, the derivative-based optimization (interior-point) method provided by IPOPT [32] was used via CasADi toolbox [33] running in MATLAB.

### A. Holonomic Systems Simulations

For the holonomic systems the dynamics is given by

$$\begin{pmatrix} x_p \\ y_p \\ \theta_p \end{pmatrix}^+ = \begin{pmatrix} x_p \\ y_p \\ 0 \end{pmatrix} + \begin{pmatrix} T v_p \cos(\theta_p) \\ T v_p \sin(\theta_p) \\ \omega_p \end{pmatrix}. \tag{10}$$

The control constraint $\bar{\omega}$ is unbounded and the running costs $\ell$ are chosen as

$$\ell_p\left(z_p, u_p\right) := \left\|\begin{pmatrix} \left(x_p - x_p^*\right)^2 \\ 5\left(y_p - y_p^*\right) \end{pmatrix}\right\|^2 + 0.2\left\|u_p\right\|^2.$$

[1] The absolute function $|\cdot|$ is approximated by the differentiable expression $|e| \approx \sqrt{e^2 + \epsilon}$, for $e \in \mathbb{R}$ and $\epsilon \ll 1$.

Note that we need only the robot position in the cost function $\ell$ in contrast to the non-holonomic case, where the orientation is additionally considered. We chose according to the non-holonomic case the same coefficients and exponent in $\ell$ to allow better comparability between both settings. The initial control guess for each subsystem $p$ is set to $u_p := \frac{1}{N}\left(z_p^0(0) - z_p^*\right)$ where we ensure feasibility of the initial guess accordingly done by Richards and How [25]. This accelerates the first execution of the optimization to obtain a first prediction. If the later prediction steps of the obtained control values $u_p$ in the initialization are not feasible, they are set to $0$.

Fig. 3 (left) illustrates the required communication effort $K$ computed via (7). It can be noticed that the full communication value increases with larger cell sizes as the simulation time also increases. Therefore, the load of communication increases due to longer execution times. The accumulated costs (8) are depicted in Fig. 3 on the right. We conclude that the accumulated costs rise with larger cell sizes, as the subsystems have to circumvent larger grid cells.

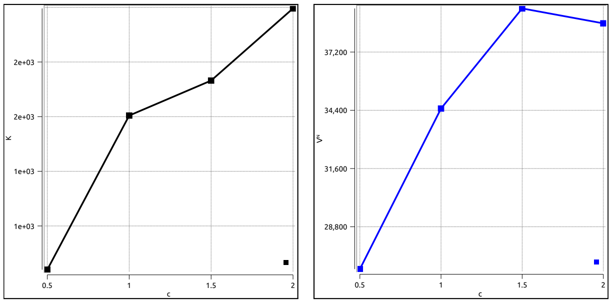

Fig. 3. 4 holonomic systems ($N = 12$): cell width $c$ vs. communication $K$ (left) and $V^N$ (right)

The necessary prediction horizon grows with increasing cell sizes as the subsystems need a longer prediction horizon to circumvent occupied cells. Therefore, the minimum horizon length for a chosen cell size $c$ is shown in Table II. In Figure 4 the trajectories the closed loop simulation trajectories are depicted.

TABLE II
PREDICTION HORIZON $N$ REQUIRED FOR EACH CELL WIDTH $c$ AND THE NUMBER OF CLOSED LOOP ITERATIONS

| cell size $c$ | $N$ | Number of Iterations $n$ |
|---|---|---|
| 0.5 | 10 | 28 |
| 1 | 10 | 36 |
| 1.5 | 12 | 39 |
| 2 | 12 | 38 |

We can see from the results, that a larger cell size leads to a larger prediction horizon $N$. The time evolution of the sum $V_P^N = \sum_{p=1}^{P} V_p^N(\cdot,\cdot)$ for each considered cell width as well as the time evolution of $V_p^N(\cdot,\cdot)$ for each robot with $c = 1$ are depicted in Fig. 5.

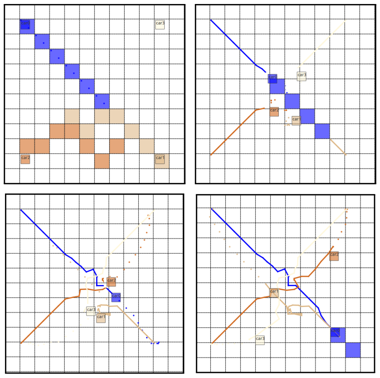

Fig. 4. 4 holonomic systems: Closed-loop simulations snap shots taken for $c = 1$ and prediction horizon $N = 12$. From up left to bottom right the snap shots are taken at $n = 0$ (left up), $n = 9$ (right up), $n = 19$ (left down) and $n = 24$ (right down).

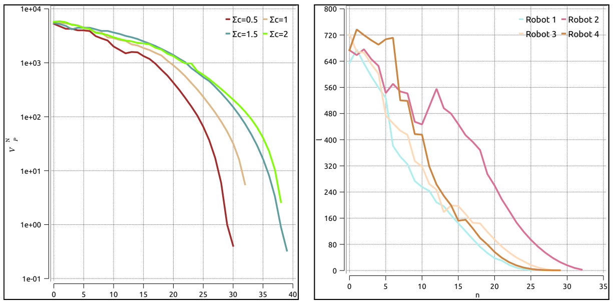

Fig. 5. 4 holonomic systems: evolution of $V_P^N = \sum_{p=1}^{P} V_p^N$ with $c = \{0.5, 1.0, 1.5, 2.0\}$ and $N = 9$ (left). Evolution of $V_p^N$ for each robot with $c = 1$ and $N = 12$ (right).

### B. Nonholonomic Systems Simulations

For the non-holonomic systems the dynamics is given by

$$\begin{pmatrix} x_p \\ y_p \\ \theta_p \end{pmatrix}^+ = \begin{pmatrix} x_p \\ y_p \\ \theta_p \end{pmatrix} + T \begin{pmatrix} v_p \cos(\theta_p) \\ v_p \sin(\theta_p) \\ \omega_p \end{pmatrix}. \tag{11}$$

The control constraint $\bar{\omega}$ is given by $\bar{\omega} = 1$. Following [34], [35] the running costs $\ell$ are chosen as

$$\ell_p\left(z_p, u_p\right) := \left\| \begin{pmatrix} (x_p - x_p^*)^2 \\ 5(y_p - y_p^*) \\ (\theta_p - \theta_p^*)^2 \end{pmatrix} \right\|_Q^2 + 0.2 \left\| u_p^2 \right\|^2$$

to ensure stabilization of non-holonomic robots without terminal constraints or costs.

The conducted simulations here are similar to the ones performed in Section IV-A, but for non-holonomic robots. Figures 6 and 7 show the results that correspond to the ones shown in Figures 3 and 5, respectively. Likewise as in the holonomic case, the communication effort and total cost expands with the increase of the cell width. The evolution of the total cost is also similar for both cases.

Figure 8 shows 4 snap shots of the closed loop trajectories for $c = 2$ and $N = 9$. Moreover, Figure 9 shows that the inter-robot distances did not violate the safety distance, i.e. $d_{\min}$.

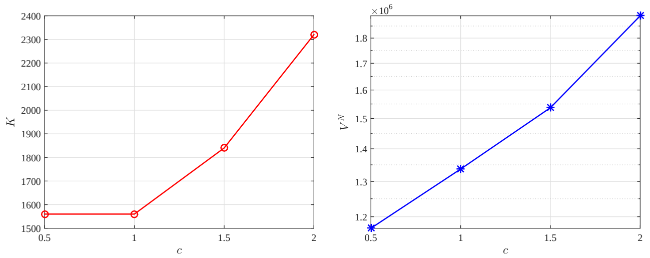

Fig. 6. 4 non-holonomic systems ($N = 9$): cell width $c$ vs. communication $K$ (left) and $V^N$ (right).

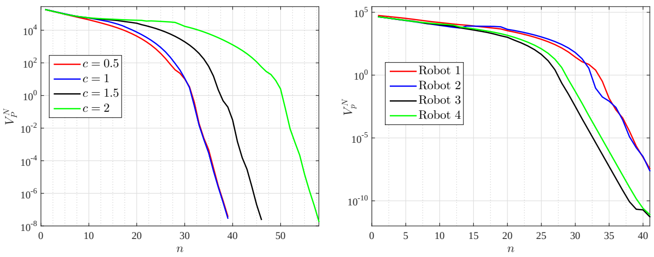

Fig. 7. 4 non-holonomic systems: evolution of $V_P^N = \sum_{p=1}^{P} V_p^N$ with $c = \{0.5, 1.0, 1.5, 2.0\}$ and $N = 9$ (left). Evolution of $V_p^N$ for each robot with $c = 1$ and $N = 9$ (right).

### C. Conclusion of Experiments

The low computational costs can be highligted as a relevant advantage of the proposed approach in relation to previous models based on the communication of predicted trajectories [27]. This is due to the fact that the information is transmitted as integers instead of floating point values. Moreover, it can be observed from Figures 3 and 6 that the required number of the communicated tuples as well as the accumulated sum of the value functions increase in general with increasing the cell size. This is primarily because of the longer simulation time needed. Moreover, in both settings, holonomic and non-holonomic, Figures 5 and 7 (left) illustrate the decreasing costs for the robots. With increasing cell sizes, the accumulated value function decreases more slowly as the robots have to circumvent larger cells.

From the individual costs functions (see Figure 5, 7 right) we can conclude that the decrease of the cost function depends on two aspects: optimization order and converging of the robots. The first one is relevant because the individual optimization tasks are executed by each robot in a fixed order. Therefore, the solution space is smaller for the last robot when all previous robots have reserved their occupancy grid predictions. This explains the curves of cost for the holonomic case shown in Fig. 5 (right), where robot 1 decreases its costs faster than the following robots. This seems to be contradictory in the non-holonomic case (see Fig. 7 right), but it is a result of the conflict with the second aspect, i.e. the converging of the robots: as the prediction horizons of the robots are converging, the first optimizing robot (1) realizes the opposite robot (3) as an obstacle at first and draw aside this robot which leads to a slight increase of the cost function of robot 1. The same situation occurs for the robot 2. As both robots (1 and 2) are drawing aside, the other robots (3 and 4) have to wait as the previous optimizing robots are blocking the alternative route.

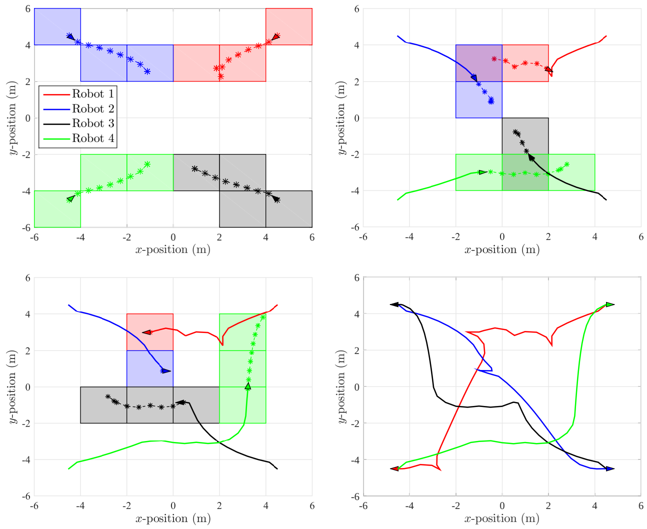

Fig. 8. Closed loop simulation snap shots for cell size $c = 2$, and prediction horizon $N = 9$. The snap shots are taken at the time steps $n = 1$ (top left), $n = 9$ (top right), $n = 22$ (bottom left), and $n = 59$ (bottom right).

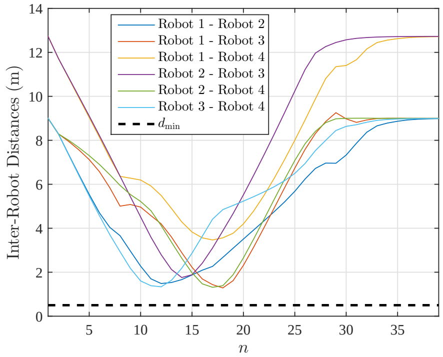

Fig. 9. Time evolution of inter-robots distances for 4 non-holonomic systems with $N = 9$ and $c = 0.5$.

## V. Conclusion

In this paper, we proposed a distributed MPC scheme based on communication of occupancy grid predictions for mobile robots. Instead of communicating the predicted trajectories in the continuous state set, we use a discretization of the state set to generate occupancy grid predictions. Each occupancy grid prediction is composed of a sequence of tuples describing the corresponding occupied cell indices. Changing the data representation of the communicated information in a discrete space leads to less communication capacity required for the proposed controller. Moreover, in the proposed method, optimization is performed in a continuous setting; thus, sensitivity information is not lost.

In this work, we provided two possible formulations of the coupling constraints utilized by the proposed controller. The first method is based on infinity norm while the latter utilizes a squirclar shape as an approximation of a square. The second method has the advantage of being less conservative than a circular approximation as well as being suitable for derivative-based optimization methods. The proposed controller with each of the formulations was applied, respectively, to a set of holonomic robots as well as a set of non-holonomic robots, yielding satisfactory results. Experiments carried out with different widths for the grid cells also demonstrated that the total cost and communication effort increase with the increase of the cell width.

Future considerations of this study lies in further reducing the communication load of the developed controller. This is currently being investigated by adopting differential communication schemes, in which only altered cells from previous communication steps are broadcasted. Moreover, investigation of the proposed controller stability as well as various priority rules for optimization are promising aspects of future work.

## References

[1] J. Acevedo, B. Arrue, I. Maza, and A. Ollero, "Distributed approach for coverage and patrolling missions with a team of heterogeneous aerial robots under communication constraints," *International Journal of Advanced Robotic Systems*, vol. 10, no. 28, pp. 1–13, 2013.

[2] M. Bernard, K. Kondak, I. Maza, and A. Ollero, "Autonomous transportation and deployment with aerial robots for search and rescue missions," *Journal of Field Robotics*, vol. 28, no. 6, pp. 914–931, 2011.

[3] M. A. Hsieh, A. Cowley, J. F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C. J. Taylor, Y. Endo, R. C. Arkin, B. Jung, D. F. Wolf, G. S. Sukhatme, and D. C. MacKenzie, "Adaptive teams of autonomous aerial and ground robots for situational awareness," *Journal of Field Robotics*, vol. 24, no. 11-12, pp. 991–1014, 2007.

[4] B. T. Stewart, S. J. Wright, and J. B. Rawlings, "Cooperative distributed model predictive control for nonlinear systems," *Journal of Process Control*, vol. 21, no. 5, pp. 698–704, 2011.

[5] M. Farina and R. Scattolini, "An output feedback distributed predictive control algorithm," *IEEE Conference on Decision and Control and European Control Conference*, no. December, pp. 8139–8144, 2011.

[6] M. Saffarian and F. Fahimi, "Non-iterative nonlinear model predictive approach applied to the control of helicopters group formation," *Robotics and Autonomous Systems*, vol. 57, no. 6–7, pp. 749 – 757, 2009.

[7] M. A. Lewis and K.-H. Tan, "High precision formation control of mobile robots using virtual structures," *Autonomous Robots*, vol. 4, no. 4, pp. 387–403, 1997.

[8] W. Ren and R. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 73–82, 2004.

[9] M. Farina, A. Perizzato, and R. Scattolini, "Application of distributed predictive control to motion and coordination problems for unicycle autonomous robots," *Robotics and Autonomous Systems*, vol. 72, pp. 248–260, 2015.

[10] T. Balch and R. Arkin, "Behavior-based formation control for multi-robot teams," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 926–939, 1998.

[11] G. Antonelli, F. Arrichiello, and S. Chiaverini, "Flocking for multi-robot systems via the null-space-based behavioral control," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 1409–1414, 2008.

[12] T. Balch and M. Hybinette, "Social potentials for scalable multi-robot formations," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1, pp. 73–80 vol.1, 2000.

[13] G. Mariottini, F. Morbidi, D. Prattichizzo, G. Pappas, and K. Daniilidis, "Leader-follower formations: Uncalibrated vision-based localization and control," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 2403–2408, 2007.

[14] X. Li, J. Xiao, and Z. Cai, "Backstepping based multiple mobile robots formation control," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 887–892, 2005.

[15] J. Sanchez and R. Fierro, "Sliding mode control for robot formations," in *Intelligent Control. 2003 IEEE International Symposium on*, pp. 438–443, 2003.

[16] R. Scattolini, "Architectures for distributed and hierarchical model predictive control – a review," *Journal of Process Control*, vol. 19, no. 5, pp. 723 – 731, 2009.

[17] R. L. Raffard, C. J. Tomlin, and S. P. Boyd, "Distributed optimization for cooperative agents: application to formation flight," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 3, pp. 2453–2459 Vol.3, Dec 2004.

[18] E. Franco, T. Parisini, and M. M. Polycarpou, "Cooperative control of distributed agents with nonlinear dynamics and delayed information exchange: a stabilizing receding-horizon approach," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pp. 2206–2211, Dec 2005.

[19] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549 – 558, 2006.

[20] M. W. Mehrez, G. K. Mann, and R. G. Gosine, "Formation stabilization of nonholonomic robots using nonlinear model predictive control," in *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on*, pp. 1–6, May 2014.

[21] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Distributed aperiodic model predictive control for multi-agent systems," *IET Control Theory Applications*, vol. 9, no. 1, pp. 10–20, 2015.

[22] T. Sprodowski and J. Pannek, "Stability of distributed mpc in an intersection scenario," *Journal of Physics: Conference Series*, vol. 659, no. 1, p. 012049, 2015.

[23] M. W. Mehrez, G. K. I. Mann, and R. G. Gosine, "An optimization based approach for relative localization and relative tracking control in multi-robot systems," *Journal of Intelligent & Robotic Systems*, pp. 1–24, 2016.

[24] A. Venkat, J. Rawlings, and S. Wright, "Stability and optimality of distributed model predictive control," *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 6680–6685, 2005.

[25] A. Richards and J. How, "A decentralized algorithm for robust constrained model predictive control," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 5, pp. 4261–4266 vol.5, June 2004.

[26] A. Richards and J. P. How, "Robust distributed model predictive control," *International Journal of Control*, vol. 80, no. 9, pp. 1517–1531, 2007.

[27] L. Grüne and K. Worthmann, *Distributed Decision Making and Control*, ch. A distributed NMPC scheme without stabilizing terminal constraints. No. 417 in Lecture Notes in Control and Information Sciences, Springer Verlag, 2012.

[28] S. Körkel, H. Qu, G. Rücker, and S. Sager, *Current Trends in High Performance Computing and Its Applications: Proceedings of the International Conference on High Performance Computing and Applications, August 8–10, 2004, Shanghai, P.R. China*, ch. Derivative Based vs. Derivative Free Optimization Methods for Nonlinear Optimum Experimental Design, pp. 339–344. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[29] M. Fernández-Guasti, "Analytic geometry of some rectilinear figures," *Int. Jour. of Math. Ed. in Sci. & Tech.*, vol. 23, no. 6, pp. 895–901, 1992.

[30] M. J. D. Powell, "Direct search algorithms for optimization calculations," *Acta Numerica*, vol. 7, pp. 287–336, 1998.

[31] S. Johnson, "The NLopt nonlinear-optimization package," 2004.

[32] A. Wächter and T. L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[33] J. Andersson, *A General-Purpose Software Framework for Dynamic Optimization*. PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, October 2013.

[34] K. Worthmann, M. W. Mehrez, M. Zanon, G. K. I. Mann, R. G. Gosine, and M. Diehl, "Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1394–1406, 2016.

[35] K. Worthmann, M. W. Mehrez, M. Zanon, G. K. Mann, R. G. Gosine, and M. Diehl, "Regulation of differential drive robots using continuous time mpc without stabilizing constraints or costs," in *Proceedings of the 5th IFAC Conference on Nonlinear Model Predictive Control (NPMC'15), Sevilla, Spain*, pp. 129–135, 2015.