Roads sweeping by unmanned multi-vehicle formations

Martin Saska and Vojtěch Vonásek and Libor Přeučil

***Abstract*— A system for autonomous roads sweeping by applying formations of mobile robots is presented in this paper. The proposed approach based on Receding Horizon Control solves the formation navigation, planning and stabilization in real-word environments with static and dynamic obstacles. The formations employed for sweeping are built up ad-hoc, taking into account length of robots' effectors (e.g. shovels, sweepers) and width of the working area. Presented method enables to smoothly merge smaller teams with the view of sweeping the larger roads (e.g. runways, highways). The formations can operate in two modes: sweeping and moving. In the sweeping mode, the formations are guided with an aim to effectively cover the cleaning roads, while in the moving mode, the planning system emphasizes the effort to reach a desired target. Furthermore, the moving mode enables to autonomously design complex formation maneuvers, as is reverse driving or turning on spot.**

I. INTRODUCTION

The autonomous sweeping of roads predestinates utilization of multi-robot systems for several reasons. In addition to the obvious aspect that a cluster of robots cleans up the designated area faster, most of the sweeping applications require to clean up the roads completely at once. As one can see for example on highways or airport runways, formations of human driven snowploughs or sweepers cover with their effectors the complete width of paths, since partly cleaned surface could be dangerous for traffic or emergency landing airplanes. The basic idea of our system is motivated by this existing concepts. We propose to built up formations of unmanned vehicles for cleaning the main roads completely (highways, runways, promenades for pedestrians etc.). Afterwards, these groups can be splitted to sub-formations for cleaning smaller roads (auxiliary roads, footways etc.) on demand of a task planning unit. Due to such ad-hoc formations, the system can effectively distributes the vehicles for particular subtasks depending on the width of sweeping roads.

In the classical literature, formation driving approaches are divided into the three main groups: virtual structure [4], behavioral techniques [10], and leader-follower methods [8]. We have chosen the leader-follower method as an appropriate approach for maintaining of ploughs with car-like kinematics. In this algorithm the followers maintain their position in the formation relative to the leader and therefore the state of the leading vehicle needs to be distributed within the team. In the literature, there is a broad offer of methods for formation stabilization in desired positions (see e.g. [6]) as well as for driving along predefined trajectories (see e.g. [3]), but approaches solving continues splitting and merging of independent groups are not investigated there. We designed such a framework solving the assemblage of ad-hoc formations in the dynamic environment of airports. The method is enough robust to deal with unforseen obstacles, changes in map structure as well as with failures of ploughs.

The tasks of sweeping involve physical constraints imposed by the vehicles (mobility constraints) or physical obstacles (environment constraints) and constraints enforced by inter vehicle relations (shape of the formation). Thus, it is important for control methodologies to incorporate system's constraints into the controllers design while preserving system stability which makes the Receding Horizon Control (RHC) especially appealing. Receding horizon control (also known as model predictive control) is an optimization based control approach often used for stabilizing linear and nonlinear dynamic systems (*e.g.*, see [1] and references reported therein). For a detailed survey of RHC methods we refer to [12] and references reported therein. The works applying RHC for formation driving control are presented in [5], [7]. We will apply the Receding Horizon Control for the formation navigation along a predefined path (the sweeping mode) and for the followers stabilization in the formation. Furthermore, we have extended the common RHC approach with an additional third time horizon to capture the overall structure of the environment for the virtual leaders trajectory planning to desired goal area (the moving mode). Such a method enables to solve general formation to target region problem as well as to autonomously design complex maneuvers of formations. For guiding the formation in arbitrary complicated maneuvers, we have developed a novel approach employing two virtual leaders, one for forward and one for backward movement.

The contribution of the presented work is in the following aspects. We propose a novel entire system for operating of the temporary built formations in the task of autonomous sweeping. The system enables to allocate robots to teams and the teams to sweeping subtasks in a way that the desired area will be cleaned up completely. We consider two different behaviors of robots. For the moving mode, where the formation should reach the beginning of the sweeping task in minimum time, we have developed the novel RHC method with additional time horizon responsible for complete trajectory planning. For the sweeping mode, we have adapted a standard RHC method to optimally cover the sweeping surface by effectors of the robots in formations. Furthermore, we have extended the trajectory planning algorithm with the

Authors are with Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague. {saska,vonasek,preucil}@labe.felk.cvut.cz

This work is supported by CTU under the grant SGS10/195/OHK3/2T/13 and by European Union under the grant Symbrion-Enlarged no. 258273.

approach employing two virtual leaders. This technique is be able to perform complicated maneuvers of the compact formations of car-like robots in dynamic environment, which was not possible with the existing methods. This ability is crucial in the sweeping applications and mainly in airport snow shoveling, since the cleaning formations have to turn on spot in case of blind roads or a road blockage.

II. Preliminaries

Let $\psi_j(t) = \{x_j(t), y_j(t), \theta_j(t)\} \in \mathcal{C}$, where $j \in \{1, \dots, n_r, L\}$, denote the configuration of each of the n_r followers and a virtual leader L of a formation at time t. $\mathcal{C} = \mathbb{R}^2 \times [0, 2\pi)$ is the configuration space. The Cartesian coordinates $x_j(t)$ and $y_j(t)$ define the position $\bar{p}_j(t)$ of a robot R_j and $\theta_j(t)$ denotes its heading. Let us assume that the environment of the robots contains a finite number n_0 of compact obstacles collected in a set of regions $\mathcal{O}_{obs}$. Finally, we need to define a circular detection boundary with radius r_s and a circular avoidance boundary with radius r_a, where $r_s > r_a$. Single robots should not respond to obstacles detected outside the region with radius r_s. On the contrary, distance between the robots and obstacles less than r_a is considered as inadmissable. These concepts are adopted from the concept of avoidance control [11] to be used in this paper for the collision avoidance guaranties. Moreover, we need to extend these zones for the virtual leaders depending on size of the formation to ensure that the result of the leader trajectory planning is feasible also for the followers. We will denote the extended radii as $r_{s,L}$ and $r_{a,L}$. The kinematics for any robots R_j, where $j \in \{1, \dots, n_r, L\}$, is described by the simple nonholonomic kinematic model: $\dot{x}_j(t) = v_j(t)\cos\theta_j(t)$, $\dot{y}_j(t) = v_j(t)\sin\theta_j(t)$ and $\dot{\theta}_j(t) = K_j(t)v_j(t)$. Velocity $v_j(t)$ and curvature $K_j(t)$ represent control inputs $\bar{u}_j(t) = \{v_j(t), K_j(t)\} \in \mathbb{R}^2$.

Let us define a time interval $[t_0, t_{N+M}]$ containing a finite sequence with $N + M + 1$ elements of nondecreasing times $\mathcal{T}(t_0, t_{N+M}) = \{t_0, t_1, \dots, t_{N-1}, t_N, \dots, t_{N+M-1}, t_{N+M}\}$. By integrating the kinematic model over a given interval $[t_0, t_{N+M}]$, and holding constant control inputs over each time interval $[t_k, t_{k+1})$ (from this point we may refer to t_k using its index k), we can derive the following model for the *transition points* at which control inputs change:

$$
\begin{aligned}
x_j(k+1) &= \begin{cases} x_j(k) + \frac{1}{K_j(k+1)}\left[\sin(\theta_j(k) + K_j(k+1)v_j(k+1)\Delta t) - \sin(\theta_j(k))\right], \text{if } K_j(k+1) \neq 0; \\ x_j(k) + v_j(k+1)\cos(\theta_j(k))\Delta t, \text{ if } K_j(k+1) = 0 \end{cases} \\
y_j(k+1) &= \begin{cases} y_j(k) - \frac{1}{K_j(k+1)}\left[\cos(\theta_j(k) + K_j(k+1)v_j(k+1)\Delta t) - \cos(\theta_j(k))\right], \text{if } K_j(k+1) \neq 0; \\ y_j(k) + v_j(k+1)\sin(\theta_j(k))\Delta t, \text{ if } K_j(k+1) = 0 \end{cases} \\
\theta_j(k+1) &= \theta_j(k) + K_j(k+1)v_j(k+1)\Delta t,
\end{aligned} \tag{1}
$$

where $\psi_j(k) = \{x_j(k), y_j(k), \theta_j(k)\}$ is the configuration at the transition point with index k. Control inputs $v_j(k+1)$ and $K_j(k+1)$ are extracted from $\bar{u}_j(k+1) := \bar{u}_j(t_k; t_{k+1} - t_k)$ at time index $k+1$.

In applications, the control inputs are limited by vehicle mechanical capabilities (i.e., chassis and engine). These constraints can be taken into account for each robot R_j limiting their control inputs by the following inequalities: $v_{min,j} \leq v_j(k) \leq v_{max,j}$ and $|K_j(k)| \leq K_{max,j}$, where $v_{max,j}$ is the maximal forward velocity of the j-th vehicle, $v_{min,j}$ is the limit on the backward velocity and $K_{max,j}$ is the maximal control curvature.

In the proposed method, we use the receding horizon control, that solves a finite horizon optimization control problem starting from current state $\psi(t_0)$ over the time interval $\langle t_0, t_0 + N\Delta t\rangle$. The duration $N\Delta t$ of the time interval $\langle t_0, t_0 + N\Delta t\rangle$ is known as the control horizon and N is number of transition points in the control horizon. After a solution from the optimization problem is obtained on a control horizon, a portion of the computed control actions is applied on the interval $\langle t_0, t_0 + n\Delta t\rangle$, known as the receding step. Parameter n is the number of transition points applied in one receding step. This process is then repeated on the interval $\langle t_0 + n\Delta t, t_0 + N\Delta t + n\Delta t\rangle$ as the finite horizon moves by *time steps* $n\Delta t$, yielding a state feedback control scheme strategy. See [9] for further details on RHC. Beyond this basic RHC concept, we have utilized a novel concept of RHC with additional time horizon $\langle t_N, t_{N+M}\rangle$ for global trajectory planning. Details on this method are published in section IV-B.

For the formation driving, we utilized a method in which the followers are maintained in relative distance to the virtual leader in curvilinear coordinates with two axes p and q, where p traces $\Psi_L(t)$ and q is perpendicular to p. The positive direction of p is defined from R_L back to the origin of the movement R_L and the positive direction of q is defined in the left half plane from the robots perspective. The shape of the formation is then uniquely determined by states $\psi_L(t_{p_i(t)}) = \{x_L(t_{p_i(t)}), y_L(t_{p_i(t)}), \theta_L(t_{p_i(t)})\}$, $i \in \{1, \dots, n_r\}$, in *travelled distance* $p_i(t)$ from R_L along the virtual leader's trajectory and by *offset distance* $q_i(t_{p_i(t)})$ between $\bar{p}_L(t_{p_i(t)})$ and $\bar{p}_i(t)$ in perpendicular direction from the virtual leader's trajectory. $t_{p_i(t)}$ is the time when the virtual leader was at the *travelled distance* $p_i(t)$ behind the actual position. The parameters $p_i(t)$ and $q_i(t)$ can vary for each follower during the mission. The following equations can be applied to convert the state of the followers in curvilinear coordinates to the state in rectangular coordinates:

$$
\begin{aligned}
x_i(t) &= x_L(t_{p_i(t)}) - q_i(t_{p_i(t)})\sin(\theta_L(t_{p_i(t)})) \\
y_i(t) &= y_L(t_{p_i(t)}) + q_i(t_{p_i(t)})\cos(\theta_L(t_{p_i(t)})) \\
\theta_i(t) &= \theta_L(t_{p_i(t)}).
\end{aligned} \tag{2}
$$

III. System overview

In this section, we introduce the structure of the proposed system for the formation sweeping. The highest level of the

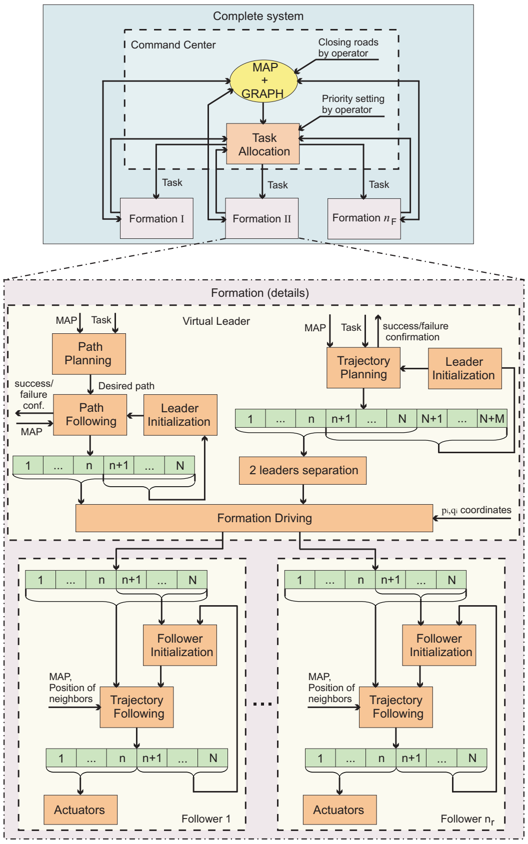


Fig. 1. Scheme of the complete sweeping system with detailed description of the formation control mechanism. The arrows denote communication links between the different modules.

proposed scheme (see the upper box called *Complete system* in Fig. 1) is divided into two types of units. The first one, *Command Center*, is responsible for the central tasks. The second one (blocks denoted as *Formation I - Formation* $n_{\mathcal{F}}$) represents the current constellation of vehicles where each unit corresponds to one formation. These units are independent from the *Command Center* most of the time.

The core of the *Command Center* is the *Task Allocation* module, which is using an agent-technology based method for autonomous design of ad-hoc formations and for planning their tasks. The obtained plan depends on priority setting for each road, on the airport traffic as well as on the snowing intensity. An execution step of the *Task Allocation* module is triggered by snowplows that have just accomplished (or failed) their task and they are waiting for new instructions. This paper does not aim at a description of the task allocation process, which is presented in our previous publication [13].

Here, we will focus in mechanisms of the formation control, stabilization, navigation and planning that are collected within the *Formation I - Formation* $n_{\mathcal{F}}$ units. These units can be logically divided to two blocks (see the bottom dash-dotted box called *Formation (details)* in Fig. 1). The fist block, called *Virtual Leader*, is responsible for navigation and control of the virtual leader (or leaders in the case of reverse driving) of the formation. This block can be physically placed onboard a sufficiently equipped follower within the formation. Such an approach increases the robustness and effectiveness of the system, because in cases of a disaffiliation with the command center the formation is able to accomplish its tasks independently. In the *Virtual Leader* control, we have to distinguish the two different behaviors: sweeping mode and moving mode. In the sweeping mode, the task allocated to the formation acts as an input of the *Path Planning* module. Here, a path that should be followed by the midpoint of the effective width of the formation is generated based on the map of the sweeping area. For example in the airport snow shoveling application, such a desired path is a sequence of axes of runways that has to be freed from snow. The virtual leader of the formation is navigated along this path using the RHC based control concept, which is described in Section IV-A. In the scheme in Fig. 1, this control mechanism is encapsulated in the *Path Following* module. In each step of the control loop, only first n samples of the solution are really implemented in the controller of the robots. Nevertheless, the rest of the solution can be used for the initialization of the planning task in the next step via the *Leader Initialization* module. This speeds up the planning process, because the plan will be significantly altered only in case of the changed topology of the environment.

In the moving mode, the desired task is represented with the target zone that should be reached by the formation. In this mode, the *Trajectory Planning* block should provide control inputs for the virtual leader (or virtual leaders in the case of reverse driving), but also the complete trajectory to the target zone which is collision free for the whole formation. The trajectory is described by a sequence of configurations of the virtual leader and by control inputs, that are applied in between the transition points. A part of the output of the *Trajectory Planning* is again re-used through the *Leader Initialization* module as a new initialization of the next control step. In the *2 leaders separation* module the virtual leader's trajectory is analyzed and if a change of the speed direction is detected, the trajectory is splitted in a way that the first virtual leader guides the formation during forward movement and the second virtual leader guides the formation during backward movement. In the *Formation Driving* module, the first N components of the leader plan are transformed to the desired configurations of the followers in both, sweeping as well as moving modes (see equations (2) for details on the formation driving approach).

The second main block, called *Follower* 1 - *Follower* n_r, is multiplied for each of the followers. The core of this block is the *Trajectory Following* module designing appropriate collision free control inputs for the vehicles. This part is responsible for the avoidance of impending collisions with obstacles or other members of the team and it should correct deviations from the desired trajectory provided by the virtual leader. Due to the RHC concept, again only the first n components of the optimal solution will be applied to the real system and the rest can be recycled in the *Follower Initialization* module. In [14], one can find implementation details and experiments describing the ability of dynamic obstacle avoidance, which cannot be presented here due to the space limitation.

IV. Leader trajectory planning and control

Details of the virtual leader trajectory planning and control methods from the *Path Following* resp. *Trajectory Planning* modules, which are employed during the sweeping resp. moving modes, are described in this section.

A. The sweeping mode

First, let us describe the k-th line segment of the desired path, designed in the *Path Planning* module, by equation $\varphi(k,s) = (P_k - P_{k-1})\,s + P_{k-1}$, where parameter s is within the interval $\langle 0,1\rangle$. The points P_k, where $k \in \{1,\ldots,\tilde{n}-1\}$, are connections of neighboring segments, P_0 is the beginning of the first segment and $P_{\tilde{n}}$ is the end of the last segment. The whole string of the segments is expressed as $\varphi(k,\cdot) = \{\varphi_x(k,\cdot), \varphi_y(k,\cdot)\}$, where $k \in \{1,\ldots,\tilde{n}\}$. Parameter $\tilde{n}$ is the number of segments provided by the *Task Allocation* module.

Having defined the desired path for the virtual leader, we can propose the leader's trajectory planning and control approach appropriate for the purpose of the sweeping by formations. The aim of the method is to find a control sequence which could navigate the virtual leader along the runways axes by minimizing a given cost function. By applying this concept, the group should be able to respond to changes in workspace that can be dynamic or newly detected static obstacles. To define the trajectory planning problem in a compact form we need to gather states $\psi_L(k)$, where $k \in \{1,\ldots,N\}$, into vector $\Psi_{L,N} \in \mathbb{R}^{3N}$ and the control inputs $\bar{u}_j(k)$, where $k \in \{1,\ldots,N\}$, into vector $\mathcal{U}_{L,N} \in \mathbb{R}^{2N}$. All variables describing the trajectory of the virtual leader can be collected in an optimization vector, $\Omega_{L,1} = [\Psi_{L,N}, \mathcal{U}_{L,N}] \in \mathbb{R}^{5N}$.

The trajectory planning can be then transformed to the minimization of cost function $J_{L,1}(\Omega_{L,1})$ subject to sets of constraints as

$$\begin{aligned} \min J_{L,1}(\Omega_{L,1}), \text{s.t. } & h_{T_N}(k) = 0, \forall k \in \{0,\ldots,N-1\} \\ & g_{T_N}(k) \leq 0, \forall k \in \{1,\ldots,N\} \\ & g_{r_{a,L}}(\Omega_{L,1}, \mathcal{O}_{obs}) \leq 0. \end{aligned}$$

The cost function $J_{L,1}(\Omega_{L,1})$ is presented as

$$\begin{aligned} J_{L,1}(\Omega_{L,1}) &= \sum_{k=1}^{N} d\left(\varphi(\cdot,\cdot), \bar{p}_L(k)\right)^2 \\ &+ \alpha \sum_{j=1}^{n_0} \left(\min\left\{0, \frac{dist_j(\Omega_{L,1}, \mathcal{O}_{obs}) - r_s}{dist_j(\Omega_{L,1}, \mathcal{O}_{obs}) - r_a}\right\}\right)^2 \\ &+ \beta \left(\int_{ind_s}^{1} \sqrt{(\varphi'_x(ind_k,s))^2 + (\varphi'_y(ind_k,s))^2}\,\mathrm{d}s \right. \\ &\quad \left. + \sum_{k=ind_k+1}^{\tilde{n}} \int_0^1 \sqrt{(\varphi'_x(k,s))^2 + (\varphi'_y(k,s))^2}\,\mathrm{d}s \right)^{-1}, \end{aligned}$$

where the first term penalizes solutions with states deviated from the desired path. The influence of the environment on the final solution is added to the cost function in the second term. Function $d\left(\varphi(\cdot,\cdot), \bar{p}_L(k)\right)$ provides the minimal distance between the desired path $\varphi(\cdot,\cdot)$ and the position $\bar{p}_L(k)$. Function $dist_j(\Omega_{L,1}, \mathcal{O}_{obs})$ provides Euclidean distance between obstacle j and the virtual leader's trajectory. The third term of the objective function is important for the convergence of the method. This part is inversely proportional to the length of the path $\varphi(\cdot,\cdot)$ between the closest point on $\varphi(\cdot,\cdot)$ to the last state $\psi_L(N)$ and the desired end of $\varphi(\cdot,\cdot)$. It "pulls" via the constraints $h_{T_N}(\cdot)$ all states $\psi_L(k)$, $k \in \{1,\ldots,N\}$, along $\varphi(\cdot,\cdot)$ to the end of $\varphi(\cdot,\cdot)$. The variables ind_k and ind_s are indexing the closest point on $\varphi(\cdot,\cdot)$. Finally, the constants α and β are utilized for the balancing of frequently antagonistic endeavors: i) closely follow the desired path, ii) avoid dynamic obstacles and iii) reach the desired goal as soon as possible.

The kinematic model (1) with initial conditions given by the actual state of the virtual leader is represented using equality constraints $h_{T_N}(k)$, $\forall k \in \{0,\ldots,N-1\}$. This satisfies that the obtained trajectory stays feasible with respect to kinematics of nonholonomic robots. Set of constraints $g_{T_N}(k)$, $\forall k \in \{1,\ldots,N\}$ characterizes bounds on the velocity and curvature of the virtual leaders defined in Section II. Finally, the avoidance inequality constraints $g_{r_{a,L}}(\Omega_{L,1}, \mathcal{O}_{obs})$, which characterize the safety avoidance regions, are defined as $g_{r_{a,L}}(\Omega_{L,1}, \mathcal{O}_{obs}) := r_{a,L}^2 - dist_j(\Omega_{L,1}, \mathcal{O}_{obs})^2, j \in \{1,\ldots,n_0\}$.

B. The moving mode

In this section, we describe the approach employing two virtual leaders, one for the forward movement and one for the backward movement. Their leading role is switched always when the sign of the leader's velocity is changed. The suspended virtual leader becomes temporarily a virtual follower. The virtual follower traces the virtual leader similarly as the other followers to be able to undertake its leading duties at time of the next switching. The virtual leaders will be positioned at the axis of the formation, one in front and one behind the formation. This concept is necessary during complex maneuvers of formations using the leader-follower approach, which cannot be applied in its simple version during forward and backward movements alternation [2].

In the presented approach, we propose to solve collision free trajectory planning and optimal control together for both virtual leaders in one optimization step. Beyond this, we extend the standard RHC method with one control horizon into an approach utilizing two finite time intervals T_N and T_M. The first time interval T_N should provide immediate control inputs for the formation regarding the local environment. The difference $\Delta t(k+1) = t_{k+1} - t_k$ is kept constant in this time interval. The second interval T_M takes into account information about the global characteristics of the environment to navigate the formation to the goal and to automatically compose the entire maneuver containing usually multiple switching between the virtual leaders. The transition points in this part can be distributed irregularly to effectively cover the environment. During the optimization process, more points should be automatically allocated in the regions where a complicated maneuver of the formation is needed. This is enabled due to the varying values of $\Delta t(k+1) = t_{k+1} - t_k$ that will be for the compact description collected into the vector $\mathcal{T}^{\Delta}_{L,M} = \{\Delta t(N+1), \ldots, \Delta t(N+M)\}$.

To define the trajectory planning problem in a compact form we need to gather states $\psi_L(k), k \in \{N+1, \ldots, N+M\}$ into vector $\Psi_{L,M} \in \mathbb{R}^{3M}$ and control inputs $\bar{u}_L(k), k \in \{N+1, \ldots, N+M\}$ into vector $\mathcal{U}_{L,M} \in \mathbb{R}^{2M}$. All variables describing the complete trajectory from the actual position of the first virtual leader until target region for both leaders can be then collected into the unique optimization vector $\Omega_L = [\Psi_{L,N}, \mathcal{U}_{L,N}, \Psi_{L,M}, \mathcal{U}_{L,M}, \mathcal{T}^{\Delta}_{L,M}] \in \mathbb{R}^{5N+6M}$.

The trajectory planning and the static as well as dynamic obstacle avoidance problem for both virtual leaders can be transformed to the minimization of single cost function $J_{L,2}(\Omega_{L,2})$ subject to sets of constraints as

$$\begin{aligned} \min J_{L,2}(\Omega_{L,2}), \text{s.t. } & h_{T_M}(k) = 0, \forall k \in \{0, \ldots, N+M-1\} \\ & g_{T_M}(k) \leq 0, \forall k \in \{1, \ldots, N+M\} \\ g_{S_F}(\psi_L(N+M)) \leq 0, & \, g_{r_{a,L}}(\Omega_{L,2}, \mathcal{O}_{obs}) \leq 0. \end{aligned}$$

The cost function $J_{L,2}(\Omega_{L,2})$ is given by

$$\begin{aligned} J_{L,2}(\Omega_{L,2}) = & \sum_{k=N+1}^{N+M} \Delta t(k) + \\ & + \alpha \sum_{j=1}^{n_0} \left(\min \left\{ 0, \frac{dist_j(\Omega_{L,2}, \mathcal{O}_{obs}) - r_{s,L}}{dist_j(\Omega_{L,2}, \mathcal{O}_{obs}) - r_{a,L}} \right\} \right)^2, \end{aligned}$$

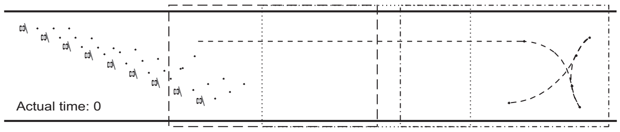
where the endeavor of the trajectory planning to reach a desired goal as soon as possible is expressed in the first part of $J_{L,2}(\cdot)$ and the influence of the environment on the final solution is added to the cost function in the second term. Again the influence of both parts of the cost function is adjusted by constant α.

The equality constraints $h_{T_M}(k), \forall k \in \{0, \ldots, N+M-1\}$ and the inequality constraints $g_{T_M}(k), \forall k \in \{1, \ldots, N+M\}$ are equivalent to $h_{T_M}(\cdot)$ and $g_{T_M}(\cdot)$ in the previous subsection. The stability constraint $g_{S_F}(\psi_L(N+M))$, guaranteeing that the obtained trajectory will enter the target region S_F, is defined as $g_{S_F}(\psi_L(N+M)) := r_{S_F} - \|\bar{p}_L(N+M) - C_{S_F}\|$. C_{S_F} is center of a target region S_F. As the target region we use a circle with radius r_{S_F} for simplification here.

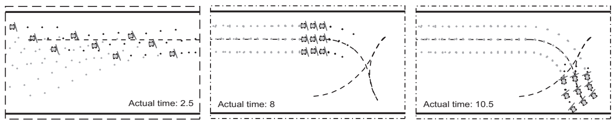
V. RESULTS

A. Computer simulations

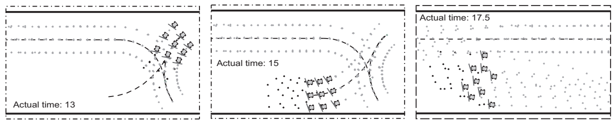
In the first simulation (Fig. 2), the approach presented in this paper is applied for turning the formation of snow ploughs at the end of cleaning runway. Such a maneuver cannot be pre-computed before the mission by operators, because the size of formations varies during the airport cleaning and also the places for turning cannot be predicted, as they are determined by the *Task Allocation* module as a response to changing environment. Once such a U-turn is required, the sweeping mode of the formation is switched to the moving mode. Using the approach presented in Section IV-B with settings $N = 4$, $M = 5$, $n = 2$, $\alpha = \beta = 1$ and $\Delta t = 0.25$s the complete turning maneuver can be found in one optimization process. Unfortunately, the cleaning formation is too widespread in the directions of p as well as q and the U-turn of the formation keeping the shoveling shape unchanged is impossible in the area bordered by the road-sides. An obvious solution is to relocate ploughs from the actual formation to a more compact shape for the turning. Once the U-turn is accomplished, the robots return to the previous positions suitable for the shoveling. The movement of the followers is planned autonomously, only the values of p and q coordinates in the turning formation are designed as an input of the turning maneuver.



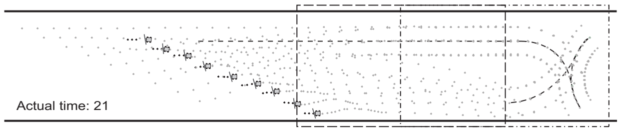
(a) The initial formation optimal for the snow sweeping.



(b) The transf. from sweeping to turning formations. (c) The compact shape appropriate for turning. (d) Time of taking command by the 2nd virtual leader.



(e) Time of switching the leadership back to the 1st leader. (f) The robots going back to the sweeping formation. (g) The transformation to the sweeping formation.



(h) The ploughs converge back to the previous formation.

Fig. 2. The U-turn of snowploughs at the end of runway.

B. Hardware experiment

The presented hardware experiments were performed on G2Bot robotic platform equipped with SICK laser range finder (not utilized in the experiment) and odometry. The G2Bot is a differential drive robot with a function emulating car-like robots kinematics. The proposed method was run on 1.2 GHz notebook with 1GB RAM. Speed commands was sent to robot's internal PC.

The first experiment is a a simplified version of the simulation in Fig. 2. In the experiment, two autonomous indoor robots are facing a blind corridor with an aim to turn at the end (see snapshots in Fig. 3). The map of the environment

(a) Two indoor robots in shoveling formations. (b) Robots have changed formation to a turning shape. (c) The second "plough" is overtaking the leadership.

(d) The leading duties are going back to the first robot. (e) The formation has finished the turning maneuver. (f) Robots are again ready for shoveling.

Fig. 3. Formation turning at the end of a blind corridor.

and positions of the vehicles are known before the mission. During the experiment, positions are updated using a dead reckoning because an external positioning system is not necessary in such a short term experiment. The actual position of robots and their plans are shared via wireless communication. The transitions between the different formations as well as the turning maneuver are computed automatically using the methods presented in Section IV-B. Only the positions of the vehicles within the formations (safety distances, required overlapping of shovels etc.) are given by experts.

The second experiment imitates a real airport snow shoveling in laboratory conditions. For the experiments the snow is made of small pieces of polystyrene and robots use straight fixed bars as shovels. The experimental scenario consists of two larger runways that has to be swept by two vehicles and four smaller roads for only one plough. Initially, the formation parameters are chosen to be $p1 = 0$m, $p2 = -1.25$m, $q1 = 0.2$m and $q2 = -0.2$m. The maximal speed of ploughs has been limited to 0.07m/s due to the utilized simple shovels and light polystyrene imitating the snow. The formation driving algorithm introduced in Section IV-A has been used with settings $N = 14$, $n = 2$, $\alpha = \beta = 1$ and $\Delta t = 2$s. Therefore the length of the control horizon is 1.96m for ploughs going with maximal speed. This enables to efficiently cover the surface of runways in sharp corners of the desired path. Time difference between two subsequent planning steps is 4s.[1] Fig. 4 shows snapshots from one of the experimental runs.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we described a system for roads sweeping by unmanned multi-vehicle formations. Beside the overall structure of the system, we presented details of novel navigation and stabilization methods for formations working in sweeping and moving modes. The functionality and robustness of the proposed algorithms were verified with a

[1]Maximal computational time needed for the leader or followers planning has been 3.23s using *fmincon* solver in MATLAB environment on the internal 1.2 GHz PC with 1GB RAM. Therefore the plans could be computed on-line during the movement of robots.

(a) Cooperative shoveling. (b) Splitting to independent units. (c) One cleaning cycle completed.

Fig. 4. Snapshots from snow shoveling hardware experiment.

simulation of iterative formations merging during airport snow shoveling, a simulation of snowploughs movement during U-turn, a simulation of respond to a robot's failure and finally with several hardware experiments in airport snow shoveling scenarios. The presented algorithms enable real time responses to dynamic environments due to the RHC concept and therefore a perfect knowledge of the environment is not necessary, which increases real-word applicability.

REFERENCES

[1] M. Alamir, *Stabilization of Nonlinear Systems Using Receding-Horizon Control Schemes*, ser. Lecture Notes in Control and Information Sciences. Berlin / Heidelberg, Germany: Springer, 2006, vol. 339.

[2] T. D. Barfoot and C. M. Clark, "Motion planning for formations of mobile robots," *Robotics and Autonomous Systems*, vol. 46, pp. 65–78, February 2004.

[3] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader-follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343–1349, 2008.

[4] K. D. Do, "Output-feedback formation tracking control of unicycle-type mobile robots with limited sensing ranges," *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 34–47, 2009.

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

[6] R. Fierro, A. Das, V. Kumar, and J. Ostrowski, "Hybrid control of formations of robots," in *Proceedings of IEEE Conference on Robotics and Automation*, vol. 1, May 2001.

[7] E. Franco, L. Magni, T. Parisini, M. Polycarpou, and D. Raimondo, "Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 324–338, February 2008.

[8] J. Fredslund and M. Mataric, "A general algorithm for robot formations using local sensing and minimal communication," *IEEE Transactions on Robotics and Automation, special issue on Advances in Multi-Robot Systems*, vol. 18, no. 5, p. 837846, October 2002.

[9] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. Balas, "Decentralized receding horizon control and coordination of autonomous vehicle formations," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 19–33, Jan. 2008.

[10] J. Lawton, R. Beard, and B. Young, "A decentralized approach to formation maneuvers." *IEEE Transactions on Robotics and Automation*, vol. 19, no. 6, p. 933941, December 2003.

[11] G. Leitmann, "Guaranteed avoidance strategies," *Journal of Optimization Theory and Applications*, vol. 32, pp. 569–576, 1980.

[12] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[13] M. Saska, M. Hess, and K. Schilling, "Route scheduling approach for airport snow shoveling using formations of autonomous ploughs," in *Proceedings of 10th International Conference on Control, Automation, Robotics and Vision (ICARCV 2008)*, Hanoi, Vietnam, December 2008.

[14] M. Saska, J. S. Mejia, D. M. Stipanovic, and K. Schilling, "Control and navigation of formations of car-like robots on a receding horizon," in *Proc of 3rd IEEE Multi-conference on Systems and Control*, 2009.