Model Predictive Collaborative Motion Planning and Control of Mobile Robots including Safety Aspects

Holger Voos, University of Applied Sciences Ravensburg-Weingarten, D-88241 Weingarten, P.O.-Box 1261, Germany, voos@hs-weingarten.de

***Abstract*— One main problem in multi-robot systems is the coordinated navigation of the autonomous robots. Hereby, the robots have to fulfill their respective tasks while avoiding collisions with other moving robots. In addition, problem-specific differential constraints like limitations of the velocities and accelerations have to be considered. Coordinated navigation of such a multi-robot system therefore has to combine contradicting aspects like efficient task accomplishment and the fulfillment of safety and problem-specific constraints in parallel. This work focuses on a two-level model predictive optimizing approach. On a global long-term level, simple dynamic models of the robots are used to compute optimal paths under differential constraints where a safety distance between all robots is achieved. Since many uncertainties and unforeseen events could occur, all robots are additionally using a non-linear model predictive control approach on a local real-time level. This control approach solves the path following and the collision avoidance problem in parallel while also considering differential constraints of the single robots.**

I. INTRODUCTION

In a multi-robot system, several mobile autonomous robots are used to act together and to achieve overall common goals. Possible areas of application are flexible manufacturing environments and here we consider the industrial example of a flexible microproduction system, see also [1] for further details. In this microproduction system, stationary machine tools are interconnected by autonomous mobile robots which are transporting the micro-workpieces in palette systems in the right sequence between the different machines. Fig. 1 depicts a possible structure of the resulting highly flexible microproduction system. One main task concerning the multi-robot system is the coordinated navigation during their transportation tasks. Here we assume that each robot receives the information about the next transportation task (e.g. provided by a higher-level production planning system) in the form of start position and time, destination position and latest delivery time as well as the assigned transportation order. The robots then have to fulfill these transportation tasks in the best possible way while the special microproduction environment adds some problem-specific constraints. Since the space between the stationary machine tools is free (i.e. without further stationary obstacles) but limited, and since several robots always operate in parallel, the biggest problem for the robots is collision avoidance with other moving robots. In addition, since the robots have to transport extremely small workpieces in palette systems which should not be disordered too much, the accelerations both in and perpendicular to the travel direction as well as velocities and turning rates are limited. In addition, the robots should move energy-efficient in order to increase the operating time with one battery charge.

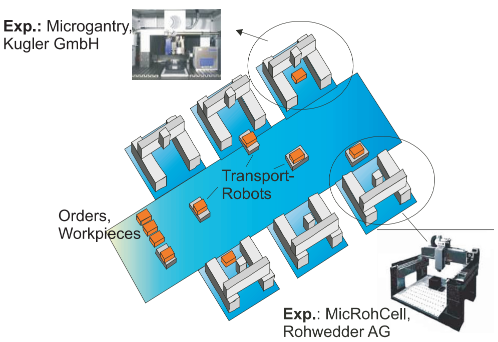

Fig. 1. A flexible microproduction system including multiple robots.

Therefore, this contribution focuses on multi-robot motion planning and control where all aspects of the special transportation task should be considered in parallel: (1) the delivery should be achieved in an optimal way, (2) safety constraints like collision avoidance must be guaranteed, (3) energy, acceleration, velocity and turn rate limitations must be considered and (4) the algorithms must be capable of responding to unforeseen changes in the environment or inaccuracies of the planning procedures. The next section first provides some more details of the considered example of the multi-robot based flexible microproduction system and describes some related work in multi-robot motion planning and control. In section 3, a two-level model predictive approach for motion planning and control of multiple robots in the microproduction system is derived while section 4 presents some first simulation and experimental results.

II. BACKGROUND AND RELATED WORK

A. *The Multi-Robot System and Experimental Testbed*

The proposed microproduction system has some special characteristics with respect to the included multi-robot system used for transportation tasks. As a manufacturing facility it will be an indoor environment with a defined structure, i.e. the machine tools and any other objects are stationary at fixed positions with free flat space in between for the navigation of the mobile robots. Since this structure is fixed it is assumed that a cartesian map of the environment is defined and available to each single robot. The only moving objects considered so far are the mobile robots while in a future extension also human workers as non-cooperative moving objects will be considered, too.

Each robot now is equipped with a suitable system for localization and communication. For localization, i.e. the determination of the robot's current location in configuration space (C-space), systems based upon local sensor information or global localization systems exist. Many possible solutions exist here, see [2] for a comprehensive overview and further references. We will neglect further details here and without loss of generality assume that a suitable localization system exist that delivers the current location of all robots in C-space with a suitable accuracy. For communication, each robot is equipped with a wireless communication system which enables the robots to exchange data with a blackboard system in a central computer. Each robot is associated with a unique ID-number for identification, and each robot's necessary data which will be detailed later will be posted at the blackboard, readable for all other robots. In order to obtain experimental results with the planned approach of the microproduction system, an experimental small-scale testbed has been developed.

B. Related Work in Multi-Robot Motion Planning

Motion planning for mobile robots is one of the fundamental and most intensively studied robotics tasks, see e.g. [2], [3], [4] for comprehensive overviews. Many of the contributions deal with path planning of one single mobile robot in an environment where collisions with static obstacles must be avoided. In this work, the main focus is on motion planning for multiple robots, and the considered obstacles are dynamic (namely the other robots). According to [2], multi-robot planning can be grouped into centralized and decoupled planning. In centralized planning, one single planner is used to compute the paths for all robots. In decoupled planning, some aspects of the planning are handled independently for each robot. Herein, no central planner is necessary and the approaches are computationally more efficient, but at the expense of extensive communication between the robots. Examples of decoupled planning are prioritized planning [5], fixed-path coordination [6], traffic rules, potential field methods or model predictive approaches [7].

In this work, we apply an adapted version of prioritized planning on the global long-term level for planning rough collision free paths defined by waypoints for all the robots. This approach fits well to the underlaying transportation problem: if any robot starts its transportation task, we generally assume that the already moving robots have a higher priority. Therefore, the considered robot computes its own collision free path with the help of a model predictive approach taking the already determined paths of the other prioritized robots as fixed. This approach then has to be extended to include differential constraints. In order to simplify the algorithms, our approach only considers velocity constraints on the global long-term planning level and more detailed differential constraints on the local real-time control level. For global motion planning, the velocities of the robots are considered as being constant but limited between two waypoints, respectively. Planning under differential constraints also has been intensively studied, see e.g. [2]. One useful approach is the discretization of the constraints by using a simplified discrete-time model of the robotic motion. In this work, the result of the global long-term decoupled planning under simplified differential constraints is a priority relation between the robots and a set of collision free waypoints for all robots from the start to the goal location with a fixed limited velocity for each way-segment between two waypoints.

However, it must be taken into account that in reality uncertainties and unforeseen events during the execution of the plans can have a strong influence on the overall resulting motion of the robots. Problems of this type are also intensively studied in the literature, see again [2] for an overview, and solutions are e.g. using probabilistic planning or dynamic re-planning on the global long-term level. In this contribution, those problems are not solved on the global long-term planning level but are combined with the solution of the path following problem and therefore solved on a local, real-time motion control level. Here, we interpret the solution of the long-term motion planning as a set of paths that must be followed by the robots with a "desired" velocity on the respective path segments. If these conditions are perfectly fulfilled this would result in collision free paths. Because of the mentioned uncertainties however, we cannot generally guarantee that no collisions occur if we only try to realize the long-term motion planning.

Therefore, all robots are continuously combining the task of path following with collision avoidance under detailed differential constraints on the local real-time motion control level. Hereby the main task for each robot is to follow the specified path with the desired velocity while continuously checking for any possible collision. This is done with the help of the blackboard and the knowledge about all current locations of the robots and therefore in a collaborative fashion. The problem of motion control like path following is also investigated in the literature, see e.g. [2] for an overview. One promising approach which motivates the proposed solution is based on model predictive control [9] for the path following or tracking problem [8] since it offers a natural way to include differential constraints. In addition, this contribution extends a non-linear model predictive path following algorithm with collision avoidance to a very efficient overall approach.

III. Model-Predictive Path Planning, Following and Collision Avoidance

A. Global Long-Term Motion Planning

The problem of multi-robot global long-term motion planning is considered here as an optimization problem under special constraints: while all robots have to fulfill their respective transportation tasks in the optimal way, the robots have to keep a safety distance from each other and also velocity constraints have to be fulfilled. In the following we assume a multi-robot system with n robots. The robots move in a cartesian x-y-coordinate system on paths given by a sequence of waypoints that are defined for a single robot $i \in \{1, \ldots, n\}$ as position vectors $\boldsymbol{r}_i(k)^T = (x_i(k), y_i(k))$

at discrete time steps $k \cdot \Delta T$ with a fixed unique time interval ΔT in between. Between the waypoints, the robot is moving with a fixed velocity vector $\boldsymbol{v}_i(k)^T = (v_{ix}(k), v_{iy}(k))$ and a simple discrete-time dynamic model of robot i is given by

$$\boldsymbol{r}_i(k+1) = \boldsymbol{r}_i(k) + \Delta T \cdot \boldsymbol{v}_i(k) \quad , \quad i \in \{1, \ldots, n\} \quad (1)$$

That modelling approach has the advantage that the positions of all robots at any given discrete time step k can be compared against each other. As previously mentioned, the transportation task of robot i is defined by the start position $\boldsymbol{r}_{iS}$ and and the destination position $\boldsymbol{r}_{iD}$ at the latest arrival time step $k = K_i$. All robots now should fulfill the transportation task in an optimal way, e.g. using a minimal amount of energy and finally minimizing the distance to the destination position. If $\boldsymbol{V}_i^T = (\boldsymbol{v}_i(0), \ldots, \boldsymbol{v}_i(K_i - 1))$ denotes the vector of all velocity vectors of robot i on its path and $\boldsymbol{R}_i^T = (\boldsymbol{r}_i(0), \ldots, \boldsymbol{r}_i(K_i))$ denotes the vector of all waypoints, this can be expressed as the following optimization problem with the objective function $J_i(\boldsymbol{V}_i, \boldsymbol{R}_i)$:

$$\min_{\{\boldsymbol{V}_i, \boldsymbol{R}_i\}} \left(J_i(\boldsymbol{V}_i, \boldsymbol{R}_i) = \sum_{k=0}^{K_i-1} |\boldsymbol{v}_i(k)|^2 + (\boldsymbol{r}_i(K_i) - \boldsymbol{r}_{iD})^2 \right) \quad (2)$$

The constraints of this optimization problem are first the equations of motion given by (1) which can be defined as a set of linear equality constraints in the form $\boldsymbol{g}_i(\boldsymbol{V}_i, \boldsymbol{R}_i) = \boldsymbol{0}$. Further constraints are the limitations of the velocities, i.e. $0 \le v_{ix}(k), v_{iy}(k) \le v_{imax}$, here simply expressed as the set of linear inequality constraints $\boldsymbol{h}_i(\boldsymbol{V}_i) \le \boldsymbol{0}$. While the constraints considered so far are local for each single robot i, there is also a set of inequalities that define the constraints of the safety distance between all robots. Therefore, each robot also has to consider the paths planned by the other robots during its own planning procedure.

In order to define a decoupled motion planning algorithm, a priority relation between all robots is defined. Herein, it is assumed that the robot which starts first has a higher priority than those robots which start later. The first robot then has the highest priority and therefore is able to plan its motion without any safety constraints. The obtained optimal path, i.e. the vectors $\boldsymbol{V}_1^*$ and $\boldsymbol{R}_1^*$ are posted at the blackboard and can be accessed by the next robot 2. This robot has to accept this path and velocities of robot 1 as given and has to optimize its path by taking further nonlinear inequality constraints into account. The next robot 3 then has to take the two higher priority path vectors into account etc. If several robots start at the same time, priority is given to them in a random fashion.

This procedure now can be generalized as follows: Assume a considered robot i where all robots $j \in \{1, \ldots, i-1\}$ have a higher priority and already determined their respective optimal path vectors $\boldsymbol{R}_j^*$. Then, the set of nonlinear inequalities considering the safety distance for robot i can be expressed as

$$|\boldsymbol{r}_i(k) - \boldsymbol{r}_j^*(k)| \ge \delta \quad \forall j, \forall k \quad (3)$$

where δ denotes the safety distance between the robots at any given discrete time step k. This can be expressed more compact as the nonlinear inequality constraints denoted by $\boldsymbol{h}_{i,\delta}(\boldsymbol{R}_1^*, \ldots, \boldsymbol{R}_{i-1}^*, \boldsymbol{R}_i) \le \boldsymbol{0}$. Regarding the optimization problem of robot i, the only variable that must be optimized then is $\boldsymbol{R}_i$, and the motion planning problem of robot $i \in \{1, \ldots, n\}$ can be written as:

$$\begin{aligned} \min_{\{\boldsymbol{V}_i, \boldsymbol{R}_i\}} \quad & J_i(\boldsymbol{V}_i, \boldsymbol{R}_i) \\ \text{s.t.} \quad & \boldsymbol{g}_i(\boldsymbol{V}_i, \boldsymbol{R}_i) = \boldsymbol{0}, \quad \boldsymbol{h}_i(\boldsymbol{V}_i) \le \boldsymbol{0} \\ & \boldsymbol{h}_{i,\delta}(\boldsymbol{R}_1^*, \ldots, \boldsymbol{R}_{i-1}^*, \boldsymbol{R}_i) \le \boldsymbol{0} \end{aligned} \quad (4)$$

This optimization problem (4) describes the optimal path planning task for each robot in the multi-robot system under the mentioned constraints on a higher level from all start to all destination positions. The solution of (4) defines the optimal path for each robot given by waypoints and also the desired constant velocities between these waypoints. However, since many unforeseen events and disturbances can occur during the movement of the robots on these paths from start to destination, these calculated paths are considered as the long term desired paths that have to be followed by controllers on a lower real-time motion control level.

B. Model-Predictive Motion Control

On the real-time motion control level, each robot has to follow the desired long-term path with the desired velocity between the waypoints. Herein the robots have to compensate any deviations from the desired path while keeping detailed differential constraints. In addition, all robots are continuously checking whether there is a threat of a collision with other robots. Because of the previously determined hierarchy of priorities, it is also fixed for the local motion control level which robots have higher or lower priority if they meet. Since all robots can access the blackboard where all current positions and velocities of all robots are posted, they consider all other robots which are currently within a certain distance limit as potential collision candidates which have to be taken into account during the local control task. However, if we have a look at the intersections of the global long-term optimal paths of the robots it becomes obvious that possible intersections of the paths mainly occur for pairs of two robots, respectively. Without any loss of generality we therefore consider only two robots 1 and 2 in the following while the approach can easily be extended to more than two robots.

It is assumed that each robot has to follow the previously calculated path, given by straight path segments between waypoints. The path following problem of the single robot 1 under consideration is depicted in Figure 2 and describes the task to follow the given path currently defined by the two waypoints $\boldsymbol{r}_1(i)$ and $\boldsymbol{r}_1(i+1)$ while the desired absolute value of the velocity (constant on that path segment) defined by the global long-term planning is denoted by $v_1(i) = v_{1D}$. In order to distinguish between the variables determined during long-term planning and real-time motion control, the variables used in real-time motion control are always denoted by a ˜-sign in the following.

For motion control we first have to specify the dynamic behavior of the robots more precisely. The mobile transport robots are equipped with two differential-drive wheels on one common axis and one castor wheel. Robots with this configuration have a restricted mobility in the sideways direction and thus have an underlying non-holonomic property. The posture, i.e. position and orientation of the robot in a Cartesian x-y-coordinate system is described by the kinematic equations

$$\begin{aligned} \dot{\tilde{x}}_1(t) &= \tilde{v}_1(t) \cdot \cos\tilde{\theta}_1(t) \\ \dot{\tilde{y}}_1(t) &= \tilde{v}_1(t) \cdot \sin\tilde{\theta}_1(t) \\ \dot{\tilde{\theta}}_1(t) &= \tilde{\omega}_1(t) \end{aligned} \tag{5}$$

Herein, $\tilde{v}_1(t)$ is the heading velocity, $\tilde{\theta}_1(t)$ is the heading angle, i.e. the angle between the x-axis and the axis of the robot 1, $\tilde{\omega}_1(t)$ is the angular velocity of the robot and $\tilde{\boldsymbol{r}}_1(t)^T = (\tilde{x}_1(t), \tilde{y}_1(t))$ is the current position vector of robot 1. Using a differential drive, the two input variables $\tilde{v}_1(t)$ and $\tilde{\omega}_1(t)$ are finally generated via the two wheel velocities of the left and the right wheel, respectively.

For a mathematical description, it is more suitable to work with a path coordinate system where $\tilde{d}_1(t)$ is the current orthogonal distance between the robot 1 and the path and $\tilde{s}_1(t)$ is the distance travelled along the path direction starting in the last waypoint. The orientation of the curent path segment between the neighboring waypoints $\boldsymbol{r}_1(i)$ and $\boldsymbol{r}_1(i+1)$ is denoted by the angle $\varphi_1(i)$, see Fig. 2. If the vector $\Delta\boldsymbol{r}_1(i) = \boldsymbol{r}_1(i+1) - \boldsymbol{r}_1(i)$ is the vector that points along the current path segment, the orthogonal distance between robot and current path segment can be calculated as

$$\tilde{d}_1(t) = \frac{|\Delta\boldsymbol{r}_1(i) \times (\tilde{\boldsymbol{r}}_1(t) - \boldsymbol{r}_1(i))|}{|\Delta\boldsymbol{r}_1(i)|} \tag{6}$$

For the description of the path following problem, it is more suitable to describe the movement of the considered robot 1 with regard to the path coordinate system in the form

$$\begin{aligned} \dot{\tilde{s}}_1(t) &= \tilde{v}_1(t) \cdot \cos(\tilde{\theta}_1(t) - \varphi_1(i)) \\ \dot{\tilde{d}}_1(t) &= \tilde{v}_1(t) \cdot \sin(\tilde{\theta}_1(t) - \varphi_1(i)) \end{aligned} \tag{7}$$

However, while following the desired path, the robots also have to avoid collisions. As previously described, the distributed global path planning algorithm results in situations where the considered robot 1 can meet a second robot

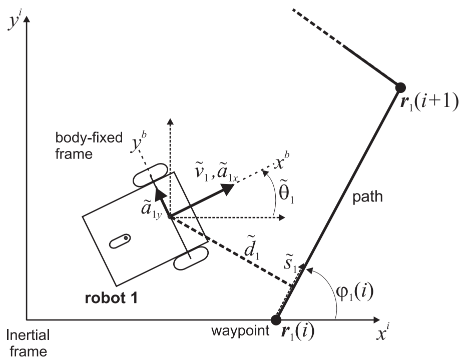

Fig. 2. The path following problem of a single robot.

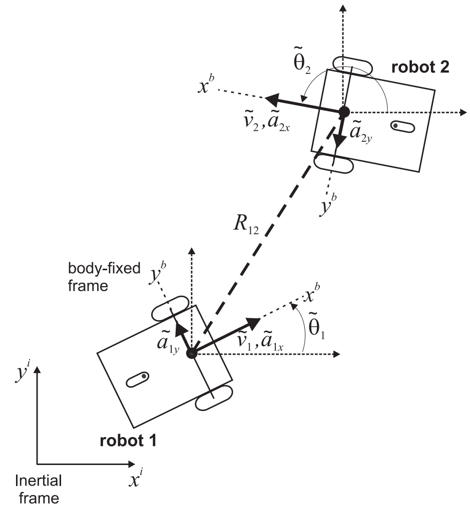

Fig. 3. The engagement geometry of two mobile robots.

2. Without loss of generality we assume that robot 2 has a higher priority than robot 1 and hence robot 1 is also responsible for the collision avoidance. The engagement geometry between two robots 1 and 2 is shown in Fig. 3. The distance $\tilde{R}_{12}(t)$ between the two robots with current local position vectors $\tilde{\boldsymbol{r}}_1(t)$ and $\tilde{\boldsymbol{r}}_2(t)$ yields

$$\tilde{R}_{12}(t) = |\tilde{\boldsymbol{r}}_1(t) - \tilde{\boldsymbol{r}}_2(t)| \tag{8}$$

From a mathematical point of view, collision avoidance means that the distance $\tilde{R}_{12}(t)$ must always be larger than the defined security threshold δ defining the constraint

$$\tilde{R}_{12}(t) > \delta \quad \forall t \tag{9}$$

Herein, the security threshold must be defined with regard to the geometry of the involved robots. As already mentioned, the microproduction environment adds some more detailed application-specific differential constraints. Since the robots have to transport extremely small parts in palette systems which should not be shaken too much, the accelerations both in travel direction ($\tilde{a}_{1x}(t)$) and perpendicular to the travel direction ($\tilde{a}_{1y}(t)$) must be limited as well as the velocities and turning rates itself:

$$\begin{aligned} &-\tilde{a}_{1y,max} < \tilde{a}_{1y}(t) = \tilde{v}_1(t) \cdot \tilde{\omega}_1(t) < \tilde{a}_{1y,max} \\ &-\tilde{a}_{1x,max} < \tilde{a}_{1x}(t) = \dot{\tilde{v}}_1(t) < \tilde{a}_{1x,max} \\ &-\tilde{\omega}_{1,max} < \tilde{\omega}_1(t) < \tilde{\omega}_{1,max} \\ &-\tilde{v}_{1,max} < \tilde{v}_1(t) < \tilde{v}_{1,max} \end{aligned} \tag{10}$$

The task of robot 1 now consists in following the desired path defined by (7) while keeping the constraints given by the kinematic equations (5), the constraints added by the collision avoidance problem (8), (9) and the problem-specific differential constraints (10). Therefore, this approach directly combines the three different and partially contradicting tasks of path following and collision avoidance under the problem-specific differential constraints. The problem is now solved by a model predictive control approach.

First we develop a discrete-time version of the underlying dynamic model on the control level. We define the vector of state variables of robot 1 as $\tilde{\boldsymbol{q}}_1(t)^T = [\tilde{x}_1(t), \tilde{y}_1(t), \tilde{\theta}_1(t), \tilde{s}_1(t), \tilde{d}_1(t)]$, and the vector $\tilde{\boldsymbol{u}}_1(t)^T = [\tilde{v}_1(t), \tilde{\omega}_1(t)]$ as the vector of input variables. The state variable differential equations are then again given by (5) and (7). Now we apply the Euler approximation to the differential quotient with time interval $\Delta\tau$ (with a small time interval $\Delta\tau \ll \Delta T$) in order to obtain a discrete-time model:

$$\dot{\tilde{\boldsymbol{q}}}_1 \approx \frac{\tilde{\boldsymbol{q}}_1(k+1) - \tilde{\boldsymbol{q}}_1(k)}{\Delta\tau} \tag{11}$$

Herein, k again denotes a discrete time step and in the following, $\tilde{\boldsymbol{q}}_1(k)$ and $\tilde{\boldsymbol{u}}_1(k)$ denote the discrete-time vectors of state and input variables of robot 1. The set of differential equations (5),(7) is then converted into a set of algebraic equations (using the notation of the input and state variables), see e.g. the conversion of the first differential equation in (5) as follows:

$$\tilde{q}_{11}(k+1) - \tilde{q}_{11}(k) - \Delta\tau(\tilde{u}_{11}(k) \cdot \cos\tilde{q}_{13}(k)) = 0 \tag{12}$$

Herein, $\tilde{q}_{1i}$ denotes the element i of the vector $\tilde{\boldsymbol{q}}_1$ of state variables. Now, also the differential constraints given by (10) can be re-formulated, see the following example of the conversion of the second equation in (10):

$$-\tilde{a}_{1x,max} < \frac{\tilde{u}_{11}(k+1) - \tilde{u}_{11}(k)}{\Delta\tau} < \tilde{a}_{1x,max} \quad \forall k \tag{13}$$

In the same way also the constraints describing the collision avoidance task (8), (9) can be re-formulated, too.

We now assume that at $t = 0$ (and hence $k = 0$) the two robots 1 and 2 have the initial vectors of state variables $\tilde{\boldsymbol{q}}_1(0)$ and $\tilde{\boldsymbol{q}}_2(0)$ and both robots have to follow a path with given current path angles $\tilde{\varphi}_1(i)$ and $\tilde{\varphi}_2(j)$, respectively. The proposed algorithm then works as follows. For a given time horizon of K time steps, robot 2 with the higher priority has to calculate its trajectories of input and state vectors $\tilde{\boldsymbol{Q}}_2 = [\tilde{\boldsymbol{q}}_2(1), \ldots, \tilde{\boldsymbol{q}}_2(K+1)]$ and $\tilde{\boldsymbol{U}}_2 = [\tilde{\boldsymbol{u}}_2(0), \ldots, \tilde{\boldsymbol{u}}_2(K)]$ in a way that the distance to the path as well as the difference between the current velocity in path direction and the desired velocity v_{2D} is minimized, using the following objective function:

$$J_2(\tilde{\boldsymbol{U}}_2, \tilde{\boldsymbol{Q}}_2) = \sum_{k=1}^{K+1} \left(\frac{\tilde{q}_{24}(k+1) - \tilde{q}_{24}(k)}{\Delta\tau} - v_{2D}\right)^2 + (\tilde{q}_{25}(k))^2 \tag{14}$$

It becomes obvious that the set of constraints with regard to the dynamics of the robot after discrete-time formulation can generally be formulated as a set of nonlinear equality constraints $\tilde{\boldsymbol{g}}_2(\tilde{\boldsymbol{U}}_2, \tilde{\boldsymbol{Q}}_2) = \boldsymbol{0}$. The problem-specific differential constraints in discrete-time formulation according to (13) can be given as a set of linear inequality constraints $\tilde{\boldsymbol{h}}_2(\tilde{\boldsymbol{U}}_2) < \boldsymbol{0}$. Therefore, the optimization problem of robot 2 finally yields

$$\begin{aligned} &\min_{\{\tilde{\boldsymbol{U}}_2, \tilde{\boldsymbol{Q}}_2\}} \quad J_2(\tilde{\boldsymbol{U}}_2, \tilde{\boldsymbol{Q}}_2) \\ &\text{s.t.} \quad \tilde{\boldsymbol{g}}_2(\tilde{\boldsymbol{U}}_2, \tilde{\boldsymbol{Q}}_2) = \boldsymbol{0}, \quad \tilde{\boldsymbol{h}}_2(\tilde{\boldsymbol{U}}_2) < \boldsymbol{0} \end{aligned} \tag{15}$$

The results are the sets of optimal input and corresponding vectors of state variables over the considered horizon given by $\tilde{\boldsymbol{U}}_2^*$ and $\tilde{\boldsymbol{Q}}_2^*$. Robot 1 now has to follow its own path while avoiding collisions with robot 2, which is assumed to be on its optimal path defined by $\tilde{\boldsymbol{Q}}_2^*$. In the collaborative approach as proposed in this work it is assumed that robot 2 communicates this planned optimal path to robot 1 via publication on the blackboard. Robot 1 now has to calculate its own optimized path while however taking the collision avoidance problem into account. This adds a further set of nonlinear inequality constraints given by $\tilde{\boldsymbol{h}}_{1,\delta}(\tilde{\boldsymbol{Q}}_2^*, \tilde{\boldsymbol{Q}}_1) \leq \boldsymbol{0}$ according to (8), (9). With the information about the future behavior of robot 2 given by $\tilde{\boldsymbol{Q}}_2^*$ robot 1 now solves the following nonlinear static optimization problem:

$$\begin{aligned} &\min_{\{\tilde{\boldsymbol{U}}_1, \tilde{\boldsymbol{Q}}_1\}} \quad J_1(\tilde{\boldsymbol{U}}_1, \tilde{\boldsymbol{Q}}_1) \\ &\text{s.t.} \quad \tilde{\boldsymbol{g}}_1(\tilde{\boldsymbol{U}}_1, \tilde{\boldsymbol{Q}}_1) = \boldsymbol{0}, \quad \tilde{\boldsymbol{h}}_1(\tilde{\boldsymbol{U}}_1) < \boldsymbol{0} \\ &\qquad \tilde{\boldsymbol{h}}_{1,\delta}(\tilde{\boldsymbol{Q}}_2^*, \tilde{\boldsymbol{Q}}_1) < \boldsymbol{0} \end{aligned} \tag{16}$$

After this calculation of the trajectories of optimal vectors of input variables $\tilde{\boldsymbol{U}}_1^*$ and $\tilde{\boldsymbol{U}}_2^*$ however, only the optimal steering commands $\tilde{\boldsymbol{u}}_1^*(0)$ and $\tilde{\boldsymbol{u}}_2^*(0)$ for the current time step are realized and the overall procedure starts again in the next time step. That means that the steering commands of the two robots are always calculated on model-based predictions of the future trajectories, but the calculated future trajectories are not fully realized. The reason for that approach is the possibility to consider disturbances of the state variables that can occur in the next time step. Thus the overall scheme is a model predictive control algorithm, see e.g. [9] for an overview, but realized by communicating robots. The full procedure can be summarized as follows:

(1) The current discrete time is set to $k = 0$, both robots 1 and 2 receive the current posture vectors $(\tilde{x}_1(0), \tilde{y}_1(0), \tilde{\theta}_1(0))$ and $(\tilde{x}_2(0), \tilde{y}_2(0), \tilde{\theta}_2(0))$ from the blackboard. (2) Both robots determine the current distance $\tilde{d}_1(0), \tilde{d}_2(0)$ to the respective paths with the help of (6). The initial value of s can be easily set to $\tilde{s}_1(0) = \tilde{s}_2(0) = 0$.

(3) Robot 2 with the higher priority solves (15) with the initial values and obtains the optimal trajectories $\tilde{\boldsymbol{U}}_2^*$ and $\tilde{\boldsymbol{Q}}_2^*$ for the time horizon of K time steps. (4) Robot 2 communicates the optimal trajectory of the state variables $\tilde{\boldsymbol{Q}}_2^*$ to the blackboard where this information is read by robot 1. (5) Robot 1 uses $\tilde{\boldsymbol{Q}}_2^*$ in order to solve the combined path following / collision avoidance problem (16) and to obtain the optimal trajectories $\tilde{\boldsymbol{U}}_1^*$ and $\tilde{\boldsymbol{Q}}_1^*$ for the time horizon of K time steps. (6) Both robots realize the optimal steering commands $\tilde{\boldsymbol{u}}_1^*(0)$ and $\tilde{\boldsymbol{u}}_2^*(0)$ for the current time step. Then they proceed again with step 1.

This model predictive motion control approach then has been implemented in both simulation environments as well as in the previously described testbed. For the implementation of the model predictive approach, the special multiple shooting based dynamic optimization package MUSCOD-II [11] has been applied. Aspects of stability are discussed in [10]. While the implemented solution always converged in our simulation and experimental tests, a proof of the stability of the described approach is currently investigated.

IV. First Results and Future Works

In the following, some first results of the derived approach are presented. In a first simulation which is intended to proof the concept of the global long-term motion planning, three robots are considered in an x-y-coordinate system. The robots start at the same time after prioritization where robot 1 has the highest, robot three the lowest priority. The result of the decoupled prioritized planning is shown in Fig. 4. Herein, S denotes the start and E the goal location, and the markers denote the calculated waypoints, respectively.

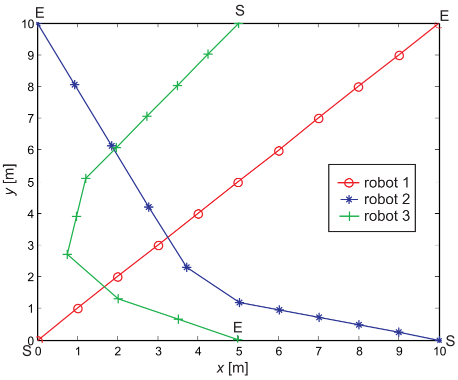

Fig. 4. Result of the global long-term motion planning.

It becomes obvious that robot 1 with the highest priority drives on the direct way from start to the goal location, herein keeping the velocity constraints. Robot 2 then has to take this path of robot 1 into account and to plan a path where the distance between these two robots is always larger than three meters. Finally, robot 3 has the lowest priority and to adapt its path to the two other already computed paths of robot 1 and 2. Also in this case, the obtained path of robot 3 keeps a distance of a least three meters between robot 3 and the other two robots. In all cases, the velocity constraints are also fulfilled. Finally, also the real-time motion control approach is tested in a simulation and in the experimental testbed described in section 2. In this experiment two robots 1 and 2 meet and robot 2 has the higher priority. Therefore, robot 2 only has to follow its desired path, as depicted in Fig. 5. Robot 1 then has to follow the path while having an initial deviation from the desired path and always has to keep a distance of a least 0.4 meters from the other robot in the testbed. For testing, the algorithms are implemented in the central computer and the steering commands are then sent to the robots, respectively. This allows the real-time solution of the optimization problems, respectively, for the robots that move with an average speed of 0.1 m/s.

The results of the model predictive approach as depicted in Fig. 5 are promising and underline its efficiency. The robot 1 first tries to minimize the deviation from the desired path, however then it has to start avoiding the approaching robot 2. That results in a deviation from the desired path of robot 1 again. After robot 2 has passed, robot 1 is again approaching the desired path. Fig. 5 also shows that the collision avoidance constraints are always fulfilled. The

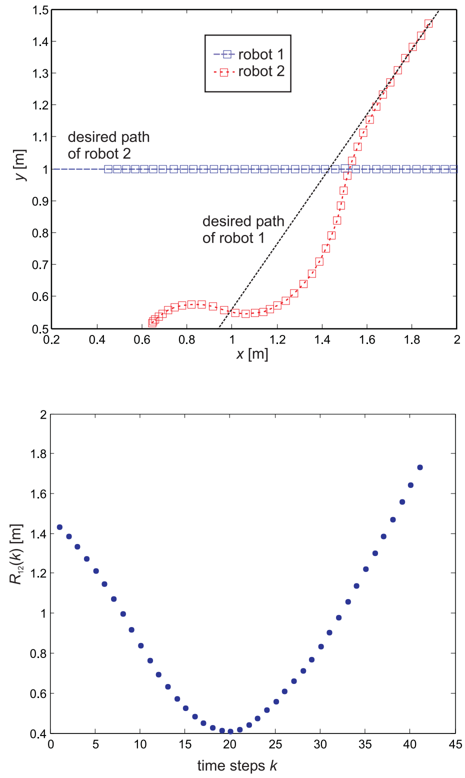

Fig. 5. Result of the local real-time motion control.

result can be interpreted as the best compromise between path following and collision avoidance while additionally keeping the differential constraints. Future work comprises the integration of the algorithms in the robots itself using embedded solutions. In addition, stability aspects are currently investigated in more detail. Finally, the approach will also be implemented in mobile robots with own local sensors for localization.

References

[1] H. Voos, *Agent-based Automation of Flexible Microproduction Systems.* In proc. of AUTOMATION 2008, Baden-Baden, Germany, 2008.

[2] B. Siciliano, O. Khatib (Eds.), *Springer Handbook of Robotics.* Berlin, Heidelberg, New York: Springer, 2008.

[3] J.C. Latombe, *Robot Motion Planning.* Boston: Kluwer, 1991.

[4] S.M. LaValle, *Planning Algorithms.* Cambridge: Cambridge University Press, 2006.

[5] J. van den Berg, M. Overmars, *Prioritized motion planning for multiple robots.* In: Proc. of IEEE Int. Conf. Intell. Robot. Systems, 2005.

[6] T. Simeon et al, *Path coordination for multiple mobile robots: A resolution complete algorithm.* In: IEEE Trans. Robot. Autom. 18(1), 42 - 49 (2002).

[7] O. A. Orqueda and R. Fierro, *Model predictive path-space iteration for multirobot coordination.* In: Cooperative Systems: Control and Optimization, D. Grundel et al (Eds.), Springer, Lecture Notes in Economics and Mathematical Systems, vol. 588, pp. 229-253, 2007.

[8] M. Bak, N. K. Poulsen, O. Ravn, *Receding horizon approach topath following mobile robots in the presence of constraints.* In: Proc. of the European Control Conference ECC2001, Porto, Portugal, 2001.

[9] F. Allgoewer et al, *Nonlinear predictive control and moving horizon estimation An introductory overview.* In: P. Frank (ed.), Advances in Control, Highlights of ECC99, Springer, London, pp. 391449, 1999.

[10] R. Findeisen, *Nonlinear Model Predictive Control: A Sampled-Data Feedback Perspective.* Duesseldorf: VDI-Verlag, 2004.

[11] M. Diehl, *Real-Time Optimization for Large Scale Nonlinear Processes*, Vol. 920 of Fortschr.-Ber. VDI Reihe 8, VDI Verlag, Duesseldorf, 2003.