RESEARCH ARTICLE

ScienceAsia 36 (2010): **333–341**

doi: 10.2306/scienceasia1513-1874.2010.36.333

Collaborative nonlinear model-predictive motion planning and control of mobile transport robots for a highly flexible production system

Suparchoek Wangmanaopituk[a,*], **Holger Voos**[b], **Waree Kongprawechnon**[a]

[a] School of Information, Computer and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, P.O. Box 22, Thammasat-Rangsit Post Office Pathum Thani 12121, Thailand

[b] Automatic Control Laboratory, Research Unit in Engineering, University of Luxembourg, Campus Kirchberg, 6 rue Coudenhove-Kalergi, L-1359, Luxembourg-Kirchberg

*Corresponding author, e-mail: suparchoek_w@yahoo.com

Received 24 May 2010
Accepted 12 Oct 2010

ABSTRACT: This study is based on a new approach for an advanced microproduction system or highly flexible production systems where all necessary production and assembly processes are connected in a very flexible way using autonomous mobile transport and handling robots. Each robot has to follow its planned paths while avoiding collisions with other robots. In addition, problem-specific constraints for a defined microproduction system, such as limitations of the velocity and accelerations of the robots, have to be fulfilled. This paper 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 also use a nonlinear model predictive control approach on a local real-time level. This control approach solves the path following and the collision avoidance problems in parallel, while also taking into account differential constraints of the single robots.

KEYWORDS: mobile robots cooperation, model-predictive control, path following and collision avoidance

INTRODUCTION

In a multi-robot system, several mobile autonomous robots are used to act together to achieve overall common goals. Possible areas of application are flexible manufacturing environments, and here, an industrial example of a flexible microproduction system has been introduced[1]. The production of microsystems such as micromotors, micropumps, microgears, and microscale optical devices needs special methods, due to their special requirements and characteristics. These products most often are produced only in small quantities with special customer needs. Hence the entire manufacturing process for miniaturized and micro-structured parts must be optimized in order to develop opportunities for cost-effective and flexible microproduction. During the last few years there has been considerable progress in new manufacturing or assembly processes for single microcomponents such as shafts, toothed wheels, and lenses. However, there is hardly any solution for the automated production of overall microproducts from these components. One main reason is that microproducts require their own specialized sequence of manufacturing and assembly steps and humans still play an important role in the processes.

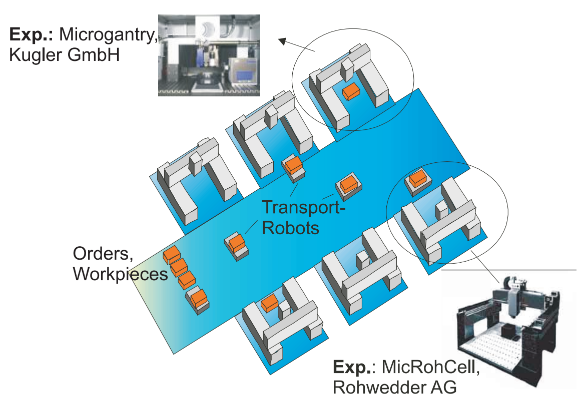

Fig. 1 Structure of the proposed microproduction system.

This paper focuses on the development of such a highly flexible automated microproduction system.

It is assumed that the system includes all necessary manufacturing and assembly processes in the form of suitable stationary machine tools as shown in Fig. 1. In order to form a highly adaptable and reconfigurable overall microproduction process, these machine tools will be interconnected by autonomous mobile robots. The multi-robot transport infrastructure allows for very flexible and even parallel interconnection of the different stationary microproduction machine tools. It is assumed that in the first solution the workpieces are fixed on palette systems for transport in order to avoid the handling of extremely small workpieces by the robots. The autonomous mobile robots have one main important task which is the transportation of the palettes in the right sequence between the different machine tools without any collision, and thus forming an adaptable and reconfigurable interconnection of single processes.

Motion planning for mobile robots is one of the fundamental and most intensively studied robotics tasks (see, e.g., Refs. 2–6 for overviews). In this paper, the main contribution is on motion planning for multiple robots[7–11], and the considered obstacles are dynamic (namely the other robots). An adapted version of prioritized planning on a global long-term level, for planning rough collision free paths defined by waypoints for all the robots, will be applied. This approach fits well to the underlying transportation problem: if any robot starts its transportation task, it can be assumed 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, this 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. Planning under differential constraints also has been intensively studied[2]. One useful approach is the discretization of the constraints by using a simplified discrete-time model of the robotic motion. In this paper, the result of the global long-term decoupled planning under simplified differential constraints is a priority relationship 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, 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[2], and solutions are obtained 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, the solution of the long-term motion planning can be interpreted 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.

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. 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, has also been investigated. One promising approach which motivates the proposed solution is based on model predictive control[12–16] for the path following or tracking problem[17,18], since it offers a natural way to include differential constraints. In addition, this contribution extends a nonlinear model predictive path following algorithm with collision avoidance to a very efficient overall approach.

MATERIALS AND METHODS

The multi-robot system and experimental testbed

The proposed microproduction system has some special characteristics with respect to the group of mobile autonomous robots. 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 manufacturing infrastructure is fixed, it is assumed that a Cartesian map of the environment is defined and available to each single robot. The only moving objects within this environment then are the robots and any human workers.

For the localization of robots and human workers, a global vision-based positioning system is assumed. That leads to the advantage that the single robots must

Fig. 2 The experimental testbed with global positioning system.

not each be equipped with their own localization system like in the simultaneous localization and mapping problem, based on visual information provided by camera or laser-scanning systems. In addition, the moving human workers can be detected outside of any sensorial range of the robots. For the design of the positioning system, a network of camera systems mounted at the top of the manufacturing facility is assumed, where the single cameras are distributed in a way that the full environment is covered by the field of view of these visual sensors. Each camera is equipped with an image processing system that is able to detect the actual position and orientation of any mobile robot and human worker in the respective field of view (Fig. 2).

In order to uniquely identify the mobile robots, these are equipped with a distinct visual pattern fixed on the robot chassis. Each distinct pattern is associated with a unique ID-number for the identification of the respective mobile robot. The position and orientation of each robot, clearly identified by the ID-number, is posted at an electronic blackboard. In addition, the position and orientation of any human workers are also posted at the blackboard without a unique ID since it would be unrealistic to assume any fixed identifiable pattern for the human workers. Each robot is able to access the blackboard and to read the information of all currently available moving objects. This access to the blackboard will be realized using wireless communication. The robots are also equipped with a camera system which is mainly intended for the control of the manipulation task. During transportation this camera system with a limited field of view in the direction of movement is used for a local emergency stop mechanism. This independent mechanism leads to an increased level of safety in case of any malfunction of the global positioning or the communication system.

Distributed Navigation

Within the multi-robot system, one main task is the fulfilment of the transportation orders assigned to the single robots. As mentioned before, these orders are defined by the start position and the destination position of the respective robots.

The overall navigation problem is structured as follows. First, each robot plans its individual optimal path according to its given transportation task. However, looking at the multi-robot system, this individual optimal planning might lead to paths that include collision points of two or even several robots, which leads to a non-optimal solution from an overall perspective. One possible solution would be a coordinated detailed path planning algorithm on the multi-robot level which would lead to optimal individual paths under the constraints that collision points are avoided.

This paper proposes a two-level distributed approach. First, all robots use a local long term planning algorithm for the calculation of individual optimal paths. This algorithm is based on a grid-map of the environment and a computation of the shortest path on the grid using efficient algorithms. It also does not include any velocity or acceleration constraints on the robots. The single robots then publish their individual optimal paths on a blackboard. All robots can access this blackboard and are looking for any points where more than two robots can meet. In those cases, the involved robots form a group and solve their problems in a way that priorities are given to the robots. The path of the robot with highest priority remains unchanged. The robots with lower priority have their local paths recalculated while the former collision grid point is blocked for them within the next shortest path calculation.

After this recalculation, the result would be a set of rough grid map based paths of the robots, where only collision points of a maximum of two robots occur. These situations are then resolved on a local level using a model-predictive approach as described

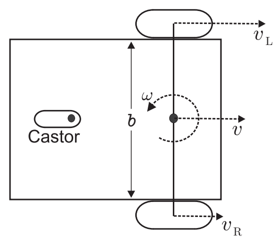

Fig. 3 Model of a mobile robot with differential drive.

in the following section.

Mathematical problem description

Mobile transport robots are robots with two differential-drive wheels on one common axis and one caster wheel (Fig. 3). Robots with this configuration have a restricted mobility in the sideways direction and thus have an underlying nonholonomic property. The posture, i.e., position and orientation of the robot in a Cartesian x-y-coordinate system is described by the kinematic equations:

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \omega, \tag{1}$$

where v is the heading velocity, θ is the heading angle, i.e., the angle between the x-axis and the axis of the robot, and ω is the angular velocity of the robot. Using a differential drive, the two input variables v and ω are generated via the two wheel velocities v_R and v_L of the right and the left wheel, respectively. If slippage can be neglected

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} 0.5 & 0.5 \\ 1/b & -1/b \end{pmatrix} \begin{pmatrix} v_\mathrm{R} \\ v_\mathrm{L} \end{pmatrix}. \tag{2}$$

The length b is the wheel base of the robot. However, if the desired values of v and ω are computed, the corresponding values of v_R and v_L can be calculated. Therefore, this equation can be inverted.

It is assumed that each robot has to follow the previously calculated path, given by straight path segments between waypoints on the grid map. The path following problem of a single robot under consideration (which will be called Robot 1) is depicted in Fig. 4 and describes the task of following the given path while the forward velocity v_R1 of the robot is not part of the control problem. It is more suitable to work with a path coordinate system where d is the current orthogonal distance between the robot and the path, and s is the distance travelled along the path direction starting from the last waypoint. The waypoint i is defined by its position vector $\mathbf{r}_i = (x_i, y_i)$ in Cartesian space and $\mathbf{r}_\mathrm{R1} = (x_\mathrm{R1}, y_\mathrm{R1})$ is the current

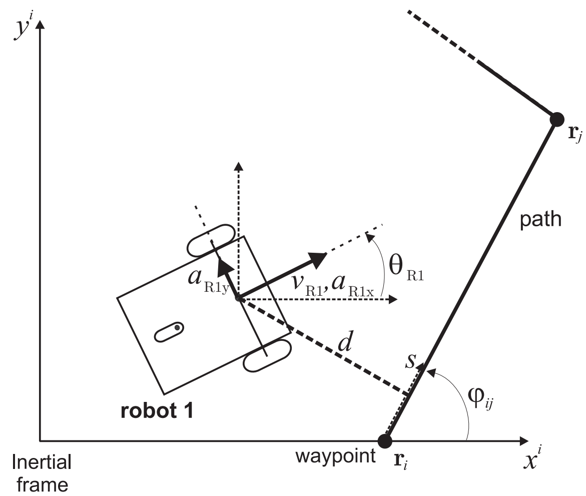

Fig. 4 The path following problem of a single robot.

position vector of Robot 1. The orientation of the path segment between the neighbouring waypoints $\mathbf{r}_i$ and $\mathbf{r}_j$ is given by the angle φ_{ij} (Fig. 4).

The vector $\mathbf{r}_{ij}$ is the vector that points along the current path segment and is given by $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$. Using this vector and the current position of the robot, the orthogonal distance between robot and current path segment can be calculated from

$$d = \frac{|\mathbf{r}_{ij} \times (\mathbf{r}_\mathrm{R1} - \mathbf{r}_i)|}{|\mathbf{r}_{ij}|}. \tag{3}$$

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

$$\begin{aligned} \dot{s}_\mathrm{R1} &= v_\mathrm{R1}\cos(\theta_\mathrm{R1} - \varphi_{ij}), \\ \dot{d}_\mathrm{R1} &= v_\mathrm{R1}\sin(\theta_\mathrm{R1} - \varphi_{ij}). \end{aligned} \tag{4}$$

However, while following the desired path from waypoint to waypoint, 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 only meet a second robot, called Robot 2. When two robots are in a collision situation, the 'right-before-left' rule is followed, i.e., the robot that comes from the right side relative to the current orientation of the two robots has priority and the other robot (coming from the left) is responsible for the collision avoidance (Fig. 5).

The distance R between the two robots with current local position vectors $\mathbf{r}_\mathrm{R1} = (x_\mathrm{R1}, y_\mathrm{R1})$ and

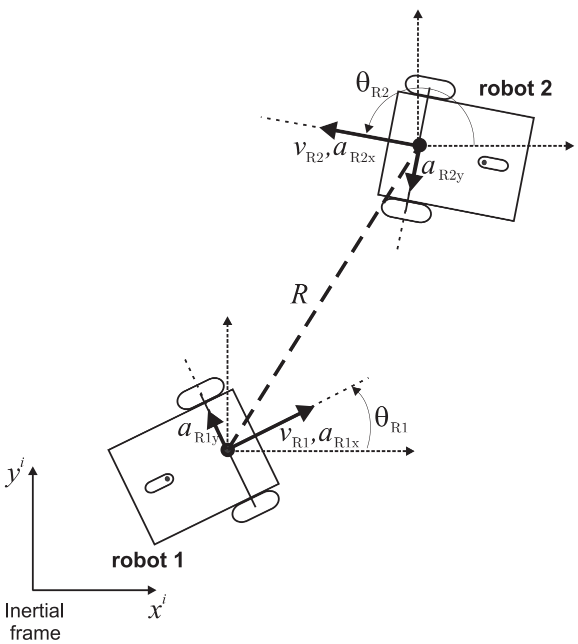

Fig. 5 The engagement geometry of two mobile robots.

$\mathbf{r}_{\mathrm{R2}} = (x_{\mathrm{R2}}, y_{\mathrm{R2}})$ is defined as:

$$R = |\mathbf{r}_{\mathrm{R1}} - \mathbf{r}_{\mathrm{R2}}| \\ = \sqrt{(x_{\mathrm{R1}} - x_{\mathrm{R2}})^2 + (y_{\mathrm{R1}} - y_{\mathrm{R2}})^2}. \tag{5}$$

Collision avoidance means that the distance R must never be smaller than a defined security threshold R_{s} defining the constraint

$$R > R_{\mathrm{s}}, \quad \forall t. \tag{6}$$

The security threshold must be defined with respect to the geometry of the involved robots. The microproduction environment results in some additional application-specific constraints. Since the robots have to transport extremely small parts in palette systems, which should not be shaken too much, the accelerations in the travel direction ($a_{\mathrm{R1}x}$) and perpendicular to the travel direction ($a_{\mathrm{R1}y}$) must be limited, as well as the velocities and turning rates:

$$\begin{aligned} &-a_{\mathrm{R1}y,\max} < a_{\mathrm{R1}y} = v_{\mathrm{R1}}\omega_{\mathrm{R1}} < a_{\mathrm{R1}y,\max}, \\ &-a_{\mathrm{R1}x,\max} < a_{\mathrm{R1}x} = \dot{v}_{\mathrm{R1}} < a_{\mathrm{R1}x,\max}, \\ &-\omega_{\mathrm{R1},\max} < \omega_{\mathrm{R1}} < \omega_{\mathrm{R1},\max}, \\ &-v_{\mathrm{R1},\max} < v_{\mathrm{R1}} < v_{\mathrm{R1},\max}. \end{aligned} \tag{7}$$

The task of Robot 1 now consists in following the desired path defined by (4) while keeping the constraints given by the kinematic equations (1), the constraints added by the collision-avoidance problem (5), (6) and the problem-specific constraints (7). Therefore, this approach directly combines these three different and partially contradicting tasks of path following and collision avoidance under the problem-specific constraints in contrast to other existing solutions. The problem is solved by a model-predictive control approach as described in the following section.

Model-predictive path following and collision avoidance

In order to derive the model-predictive approach, a discrete-time version of the underlying dynamic model will be developed. The dynamic equations that describe the kinematics of any robot n, $n \in \{1, 2\}$ as well as the dynamics with respect to the desired path are obtained from (1) and (4):

$$\begin{aligned} \dot{x}_{\mathrm{R}n} &= v_{\mathrm{R}n}\cos\theta_{\mathrm{R}n}, \\ \dot{y}_{\mathrm{R}n} &= v_{\mathrm{R}n}\sin\theta_{\mathrm{R}n}, \\ \dot{\theta}_{\mathrm{R}n} &= \omega_{\mathrm{R}n}, \\ \dot{s}_{\mathrm{R}n} &= v_{\mathrm{R}n}\cos(\theta_{\mathrm{R}n} - \varphi_{ij}), \\ \dot{d}_{\mathrm{R}n} &= v_{\mathrm{R}n}\sin(\theta_{\mathrm{R}n} - \varphi_{ij}). \end{aligned} \tag{8}$$

The vector of state variables $\mathbf{x}_{\mathrm{R}n} = [x_{\mathrm{R}n}, y_{\mathrm{R}n}, \theta_{\mathrm{R}n}, s_{\mathrm{R}n}, d_{\mathrm{R}n}]$, and the vector $\mathbf{u}_{\mathrm{R}n} = [v_{\mathrm{R}n}, \omega_{\mathrm{R}n}]$ are defined as the vector of input variables with respect to robot n.

From (7) the problem-specific constraints can therefore be written as

$$\begin{aligned} &-a_{\mathrm{R}ny,\max} < u_{1,\mathrm{R}n}u_{2,\mathrm{R}n} < a_{\mathrm{R}ny,\max}, \\ &-a_{\mathrm{R}nx,\max} < \dot{u}_{1,\mathrm{R}n} < a_{\mathrm{R}nx,\max}, \\ &-\omega_{\mathrm{R}n,\max} < u_{2,\mathrm{R}n} < \omega_{\mathrm{R}n,\max}, \\ &-v_{\mathrm{R}n,\max} < u_{1,\mathrm{R}n} < v_{\mathrm{R}n,\max}, \end{aligned} \tag{9}$$

where $u_{i,\mathrm{R}n}$ is component i of the vector $\mathbf{u}_{\mathrm{R}n}$. The collision avoidance constraint between Robot 1 and Robot 2 is given by (6).

By applying the Euler approximation to the differential quotient with time interval ΔT, the set of differential equations (8) is converted into a set of algebraic equations (using the notation of the input and state variables). The conversion of the first differential equation in (8) is

$$x_{1,\mathrm{R}n}(k+1) - x_{1,\mathrm{R}n}(k) \\ - \Delta T(u_{1,\mathrm{R}n}(k)\cos x_{3,\mathrm{R}n}(k)) = 0, \tag{10}$$

where k denotes a discrete time step, and $\mathbf{x}_{\mathrm{R}n}(k)$ and $\mathbf{u}_{\mathrm{R}n}(k)$ denote the discrete-time vectors of state and input variables.

Assume that at $t = 0$ (and hence $k = 0$) Robot 1 and Robot 2 have the initial vectors of state variables $\mathbf{x}_{\mathrm{R}1}(0)$ and $\mathbf{x}_{\mathrm{R}2}(0)$ and both robots have to follow a path with given current path angles $\varphi_{ij,1}$ and $\varphi_{ij,2}$, respectively. The proposed algorithm then works as follows. For a given time horizon of K time steps, those trajectories of input and state vectors, $\mathbf{X}_{\mathrm{R}n} = [\mathbf{x}_{\mathrm{R}n}(1), \dots, \mathbf{x}_{\mathrm{R}n}(K+1)]$ and $\mathbf{U}_{\mathrm{R}n} = [\mathbf{u}_{\mathrm{R}n}(0), \dots, \mathbf{u}_{\mathrm{R}n}(K)]$, have to be calculated in a way that the two robots travel along their paths while keeping all constraints. This can be done via the minimization of the objective function

$$
\begin{aligned}
J_{\mathrm{R}n}(\mathbf{U}_{\mathrm{R}n}, \mathbf{X}_{\mathrm{R}n}) &= \sum_{k=1}^{K+1} (d_{\mathrm{R}n}(k))^2 - (s_{\mathrm{R}n}(k))^2 \\
&= \sum_{k=1}^{K+1} (x_{5,\mathrm{R}n}(k))^2 - (x_{4,\mathrm{R}n}(k))^2.
\end{aligned}
\tag{11}
$$

The first term is included to minimize the distance to the path while the second term is included in order to move the robot forward along the path. The set of constraints with respect to the dynamics of the robot (8) after discrete-time formulation (10) can generally be formulated as a set of equality constraints with the vector function $\mathbf{g}_{\mathrm{R}n}(\mathbf{U}_{\mathrm{R}n}, \mathbf{X}_{\mathrm{R}n}) = \mathbf{0}$. The problem-specific constraints (9) can be given as a set of inequality constraints with the vector function $\mathbf{h}_{\mathrm{R}n}(\mathbf{U}_{\mathrm{R}n}) < \mathbf{0}$. The constraints describing the collision avoidance problem (5), (6) between Robot 1 and Robot 2 can finally be formulated as a set of inequality constraints defined by the vector function $\mathbf{c}_{\mathrm{R}1,\mathrm{R}2}(\mathbf{x}_{\mathrm{R}1}, \mathbf{x}_{\mathrm{R}2}) < \mathbf{0}$. It has to be taken into account that this set of inequality constraints depends on the state variables of both robots. However, since Robot 2 has priority, it can optimize its own path following problem over the horizon of K time steps without taking into account the collision avoidance problem by solving the nonlinear static optimization problem,

$$
\begin{aligned}
&\min_{\{\mathbf{U}_{\mathrm{R}2}, \mathbf{x}_{\mathrm{R}2}\}} J_{\mathrm{R}2}(\mathbf{U}_{\mathrm{R}2}, \mathbf{x}_{\mathrm{R}2}) \\
&\quad \text{s.t. } \mathbf{g}_{\mathrm{R}2}(\mathbf{U}_{\mathrm{R}2}, \mathbf{x}_{\mathrm{R}2}) = \mathbf{0}, \mathbf{h}_{\mathrm{R}2}(\mathbf{U}_{\mathrm{R}2}) < \mathbf{0}.
\end{aligned}
\tag{12}
$$

The results are the sets of optimal input and corresponding vectors of state variables over the horizon denoted by $\mathbf{U}^*_{\mathrm{R}2}$ and $\mathbf{x}^*_{\mathrm{R}2}$. Robot 1 now has to follow its path while avoiding collisions with Robot 2, which is assumed to be on its optimal path defined by $\mathbf{x}^*_{\mathrm{R}2}$. In the collaborative approach as proposed in this paper, it is assumed that Robot 2 communicates this planned optimal path to Robot 1. With the information about the future behaviour of Robot 2 given by $\mathbf{x}^*_{\mathrm{R}2}$, Robot 1 now solves the following nonlinear static optimization problem:

$$
\begin{aligned}
&\min_{\{\mathbf{U}_{\mathrm{R}1}, \mathbf{x}_{\mathrm{R}1}\}} \quad J_{\mathrm{R}1}(\mathbf{U}_{\mathrm{R}1}, \mathbf{x}_{\mathrm{R}1}) \\
&\quad \text{s.t. } \mathbf{g}_{\mathrm{R}1}(\mathbf{U}_{\mathrm{R}1}, \mathbf{x}_{\mathrm{R}1}) = \mathbf{0}, \mathbf{h}_{\mathrm{R}1}(\mathbf{U}_{\mathrm{R}1}) < \mathbf{0}, \\
&\qquad \mathbf{c}_{\mathrm{R}1,\mathrm{R}2}(\mathbf{x}_{\mathrm{R}1}, \mathbf{x}^*_{\mathrm{R}2}) < \mathbf{0}.
\end{aligned}
\tag{13}
$$

After the calculation of the trajectories of optimal vectors of input variables $\mathbf{U}^*_{\mathrm{R}1}$ and $\mathbf{U}^*_{\mathrm{R}2}$, only the optimal steering commands $\mathbf{u}^*_{\mathrm{R}1}(0)$ and $\mathbf{u}^*_{\mathrm{R}2}(0)$ for the current time step are realized, and the overall procedure starts again in the next time step. This means that the steering commands of the two robots are always calculated from model-based predictions of the future trajectories, but the calculated future trajectories are not fully implemented. The reason for that approach is the possibility of disturbances of the state variables that can occur in the next time step. Thus the overall scheme is a model-predictive control algorithm, 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 receive the current posture vectors $(x_{\mathrm{R}1}(0), y_{\mathrm{R}1}(0), \theta_{\mathrm{R}1}(0))$ and $(x_{\mathrm{R}2}(0), y_{\mathrm{R}2}(0), \theta_{\mathrm{R}2}(0))$ from the global localization system.

2. Both robots determine the current distance $d_{\mathrm{R}1}(0)$, $d_{\mathrm{R}2}(0)$ to the respective paths with the help of (3) and the internally stored global paths given by waypoints. The initial value of s is set to $s_{\mathrm{R}1}(0) = s_{\mathrm{R}2}(0) = 0$.

3. Robot 2 solves (12) with the initial values and obtains the optimal trajectories $\mathbf{U}^*_{\mathrm{R}2}$ and $\mathbf{x}^*_{\mathrm{R}2}$ for the time horizon of K time steps.

4. Robot 2 communicates the optimal trajectory of the state variables $\mathbf{x}^*_{\mathrm{R}2}$ to Robot 1.

5. Robot 1 receives $\mathbf{x}^*_{\mathrm{R}2}$ and solves the combined path following/collision avoidance problem (13) to obtain the optimal trajectories $\mathbf{U}^*_{\mathrm{R}1}$ and $\mathbf{x}^*_{\mathrm{R}1}$ for the time horizon of K time steps.

6. Both robots realize the optimal steering commands $\mathbf{u}^*_{\mathrm{R}1}(0)$ and $\mathbf{u}^*_{\mathrm{R}2}(0)$ for the current time step. Then they proceed with Step 1 again.

This model predictive motion control approach was 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 [19] was applied.

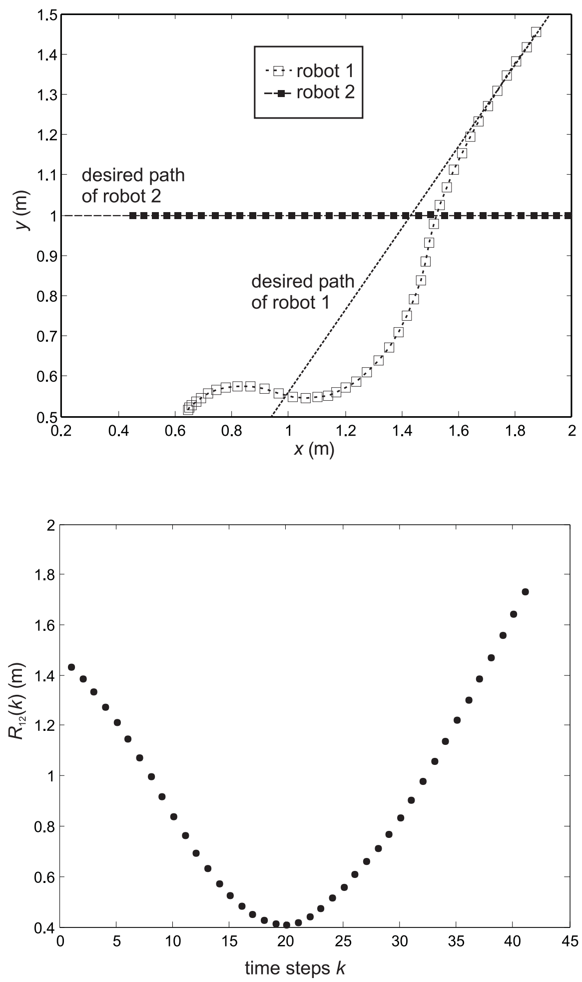

Fig. 6 Local real-time motion control.

RESULTS

The results of the model predictive approach as depicted in Fig. 6 are promising and underline its efficiency. Robot 1 first tries to minimize the deviation from the desired path. 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. 6 also shows that the collision avoidance constraints are always fulfilled. Also, the security threshold has been limited. The result can be interpreted as the best compromise between path following and collision avoidance while additionally keeping the differential constraints. In addition, for microproduction-specific constraints, the accelerations both in travel direction ($a_{\mathrm{R}1x}$) and perpendicular to the travel direction ($a_{\mathrm{R}1y}$), have been limited, as well as the velocities and turning rates.

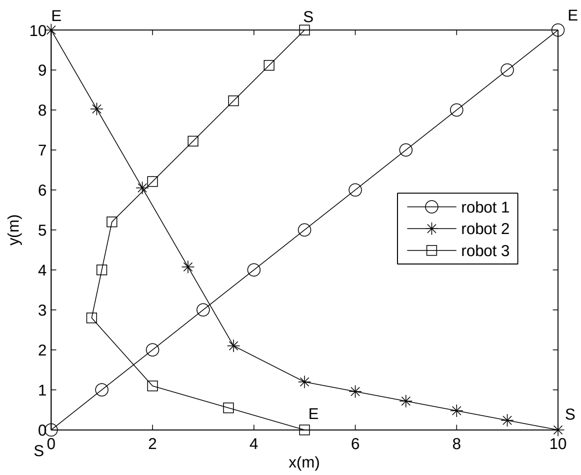

Fig. 7 Global long-term motion planning. S: start; E: goal location; markers: calculated waypoints.

Moreover, the concept of the global long-term motion planning has been simulated as well (Fig. 7). 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 priority, and Robot 3 has the lowest. The result of the decoupled prioritized planning is demonstrated.

It is clear that Robot 1 with the highest priority travels directly from start to the goal location, 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 3 m. Finally, Robot 3 has the lowest priority and has to adapt its path to the two other already computed paths of Robots 1 and 2. Also in this case, the obtained path of Robot 3 keeps a distance of a least 3 m between itself and the other two robots. Additional results are shown in Figs. 8 and 9.

DISCUSSION

While tremendous progress in the area of nonlinear model predictive control (NMPC) has been made in the recent years, there is the concern that only a small number of the existing NMPC schemes can be applied in real applications. This concern is mostly based on the high computational demand often related to NMPC. Hence computation speed issues have been addressed in this approach as well. In this proposed microproduction system, the transport robots are supposed to move with low acceleration and deceleration during the transportation because they have to carry microproducts which are very sensitive. There are also a lot of study groups (e.g., Refs. 19, 20) investigating this area to improve the performance of the NMPC for

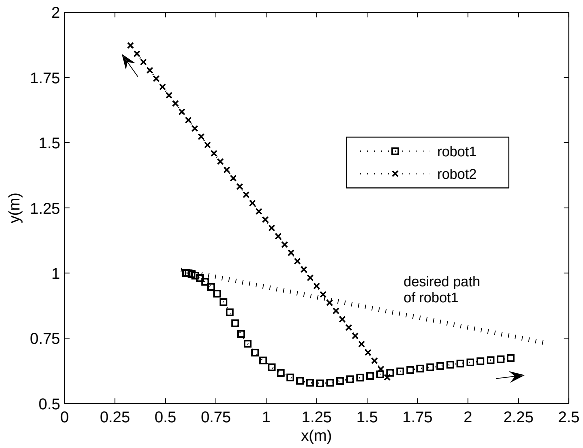

Fig. 8 Result from different scenarios.

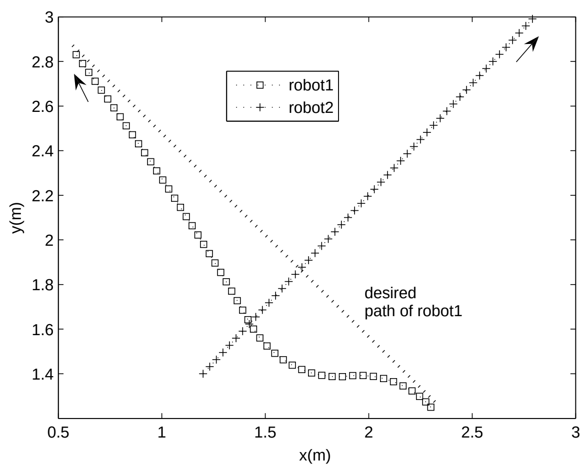

Fig. 9 Another result with lower safety distance.

other applications.

This two-level approach works properly for the proposed microproduction system, because the Microsystems fabrication laboratory is mostly a big clean room, and there are no constraints from the environment such as tables, chairs or others obstacles. In other applications, some practical problems could occur, for example, a dead-lock situation in the case of a narrow path, and so on. Then, these well-known problems require an extra approach[21], e.g., robot coordination and simultaneous scheduling.

From Figs. 6, 8 and 9, instead of applying the traffic rule (right before left), Robot 1 has to deviate from its desired path in order to avoid hard breaking, and keep the acceleration constraints at the same time. Hence Robot 1 always moves with the maximum speed. The safety distance always has to be under the setpoint value as well.

The location and orientation of human workers is known by all robots because a global vision based positioning system has been used in the proposed system. Hence the possible movement and trajectories of the human workers can be estimated. As mentioned before, the limited speed of transport robots is quite low during the handling tasks, so the proposed algorithm will have enough time.

There are a lot of possibilities to use a team of mobile robots within industries in the future. Future work comprises the integration of the algorithms in the robots using embedded solutions for higher accuracy and efficiency to enable them to run in real time. While the implemented solution always converged in the simulation and experimental tests, proof of the stability of the described approach is currently being investigated. Aspects of stability are also discussed in Ref. 15.

Acknowledgements: The authors would like to thank the School of Information, Computer and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University and the Laboratory for Mobile Robotics and Mechatronics, Hochschule Ravensburg-Weingarten University of Applied Sciences for providing inspiration and great support. The authors would like also to give special thanks to the Thai Khem Kaeng stimulus programme for financial support.

REFERENCES

1. Voos H (2008) Agent-based automation of flexible microproduction systems. In: *VDI Kongress AUTOMATION 2008*, Baden-Baden.
2. Siciliano B, Khatib O (2008) *Springer Handbook of Robotics*, Springer, Berlin.
3. Latombe JC (1991) *Robot Motion Planning*, Kluwer, Boston.
4. LaValle SM (2006) *Planning Algorithms*, Cambridge Univ Press, Cambridge.
5. Fujimori A, Teramoto M, Nikiforuk PN, Gupta MM (2000) Cooperative collision avoidance between multiple mobile robots. *J Robotic Syst* **17**, 347–63.
6. Van den Berg J, Overmars M (2005) Prioritized motion planning for multiple robots. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems, pp 430–5.
7. Fox D, Burgard W, Kruppa H, Thrun S (2000) A probabilistic approach to collaborative multi-robot localization. *Autonom Robot* **8**, 325–44.
8. Simeon T, Leroy S, Laumond JP (2002) Path coordination for multiple mobile robots: A resolution complete algorithm. *IEEE Trans Robot Autom* **18**, 42–9.
9. Jadbabaie A, Lin J, Morse A (2003) Coordination of groups of mobile autonomous agents using near-

est neighbor rules. *IEEE Trans Automat Contr* **48**, 988–1001.
10. Albagul A, Wahyudi (2004) Dynamic modelling and adaptive traction control for mobile robots. *International Journal of Advanced Robotic Systems* **1**, 149–54.
11. Mastellone S, Stipanović DM, Graunke CR, Intlekofer KA, Spong MW (2008) Formation control and collision avoidance for multi-agent non-holonomic systems: theory and experiments. *Int J Robot Res* **27**, 107–26.
12. Allöwer F, Badgwell TA, Qin SJ, Rawlings JB, Wright SJ (1999) Nonlinear predictive control and moving horizon estimation: An introductory overview. In: Advances in Control, Highlights of ECC99, pp 391–449.
13. Shim DH, Kim HJ, Sastry S (2003) Decentralized nonlinear model predictive control of multiple flying robots. In: Proceedings of 42nd IEEE Conference on Decision and Control, vol 4, pp 3621–6.
14. Wesselowski K, Fierro R (2003) A dual-mode model predictive controller for robot formations. In: Proceedings of the 42nd IEEE Conference on Decision and Control, pp 3615–20.
15. Findeisen R (2004) Nonlinear model predictive control: a sampled-data feedback perspective. *VDI-Verlag 2004*, Düsseldorf, Germany.
16. Orqueda O, Fierro R (2007) Model predictive path-space iteration for multi-robot coordination. In: Grundel D, Murphey R, Pardalos P, Prokopyev O (eds) *Cooperative Systems: Control and Optimization*, Springer, Berlin pp 229–53.
17. Bak M, Poulsen NK, Ravn O (2001) Receding horizon approach to path following mobile robots in the presence of constraints. In: Proceedings of the 6th European Control Conference, Porto, pp 1151–6.
18. Lepetič M, Klančar G, Škrjanc I, Matko D, Potočnik B (2003) Time optimal path planning considering acceleration limits. *Robot Autonom Syst* **45**, 199–210.
19. Diehl M (2003) Real-time optimization for large scale nonlinear processes. *VDI-Verlag 2003*, Duesseldorf, Germany.
20. Findeisen R, Diehl M, Nagy Z, Allöwer F, Bock HG, Schlöder JP (2001) Computational feasibility and performance of nonlinear model predictive control schemes. In: Proceedings of the 6th European Control Conference, Porto, pp 957–61.
21. Chang YC, Yamamoto Y (2008) On-line path planning strategy integrated with collision and dead-lock avoidance schemes for wheeled mobile robot in indoor environments. *Ind Robot* **35**, 421–34.