Kasetsart J. (Nat. Sci.) 48 : 283 - 293 (2014)

A Nonlinear Model-Predictive Motion Planning and Control System for Multi-Robots in a Microproduction System with Safety Constraints and a Global Long-Term Solution

Suparchoek Wangmanaopituk[1,*], Holger Voos[2] and Waree Kongprawechnon[1]

ABSTRACT

The manufacturing of microsystems such as micromotors and micropumps among other examples is a very important emerging market. One big challenge in microproduction is mass customization, that is, the automated production of a large variety of products that are highly adapted to special customer needs in small batch sizes. These requirements call for a highly flexible manufacturing system. This study focuses on the multi-robot coordination of the resulting flexible microproduction system which is solved here by the application of a multi-agent system. Additionally, all robots additionally applied the proposed nonlinear model predictive control approach on a local real-time level to solve problems associated with path-following and collision avoidance in parallel, while also considering differential constraints on single robots, such as velocity constraints, in this specific application. The global long-term motion planning approach was also considered as an optimization problem.

Keywords: multi-agent system, flexible production system, mobile robots, nonlinear model-predictive control, applied information technology

INTRODUCTION

Intelligent agents and multi-agent systems represent the next big step in the development of next-generation manufacturing systems. In a multi-agent system, the agents coordinate their behavior and solve problems in a distributed fashion without central control using only local and limited resources and information. Therefore, multi-agent systems provide all the basic properties of a distributed automatic control system. Successful industrial applications of multi-agent systems have been reported for the control of manufacturing, logistics, traffic, or telecommunication systems (Demazeau *et al*., 2010; Srinivasan and Jain, 2010).

In the last decade, multi-agent systems have been increasingly applied to solve system reconfiguration problems. Most of the studies in this area have been applied in only a small number of real world applications. Dias *et al*. (2004) gave an example for a space application. Furthermore, they illustrated how to use market mechanisms to coordinate a multi-robot team in the task decomposition, assignment and execution phases for map building, reconnaissance and

[1] School of Information, Computer and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani 12121, Thailand.

[2] Automatic Control Laboratory, Research Unit in Engineering, University of Luxembourg, Luxembourg-Kirchberg, Luxembourg.

* Corresponding author, e-mail: suparchoek.wangmanaopituk@gmail.com

Received date : 06/09/13 Accepted date : 11/12/13

perimeter sweeping. The current research in flexible manufacturing systems involved the use of automated guided vehicles (Shaikh and Dhale, 2013). These vehicles simplify the problem of navigation by following paths based on striping on the floor in some manner or by using buried cables. The state-of-the-art of mobile robot technology and predictions of future developments give a clear view that mobile robots are going to be an essential part of every manufacturing process in the not too far future (Voos, 2008) and multi-robot systems have been studied in many research areas (LaValle, 2006; Mastellone *et al*., 2008; Ducatelle *et al*., 2011; Bhattacharya *et al*., 2012). Robots now are able to intelligently move from place to place and collect parts and take them to the appropriate work cell, which opens up a new way of structuring a manufacturing environment. An industrial example of a flexible microproduction system was the focus of the current study at the request of the Federal Ministry of Education and Research (BMBF) in Thailand and by microproduction companies in Germany—namely, industrial partners Kugler GmbH and Rohwedder AG Micro Technology.

A suitable functional structure of the overall complex control task was developed. The highest level of such a functional structure comprises all long term scheduling and planning tasks for the overall production using multi-agent systems. The formal descriptions and models of the capabilities of the manufacturing and assembly modules, the necessary processing steps for any production orders and the capabilities of the mobile robots were also defined together with the partner microproduction companies. The result of any control action on this level was an overall flexible production plan for a certain time frame, taking the mobile robots into account. These control tasks on the highest control level can be solved by a multi-agent system. Therefore, suitable agent architectures and platforms were investigated and selected.

The proposed system included all necessary manufacturing and assembly processes in the form of suitable stationary machine tools (Figure1). In order to solve the scheduling and planning task in a distributed fashion, it is reasonable to assign agents to all involved entities like machines, production orders and mobile transport robots. The most important research task was the investigation of suitable communication, interaction and coordination mechanisms between the agents to determine the overall production plan in a very flexible and completely decentralized

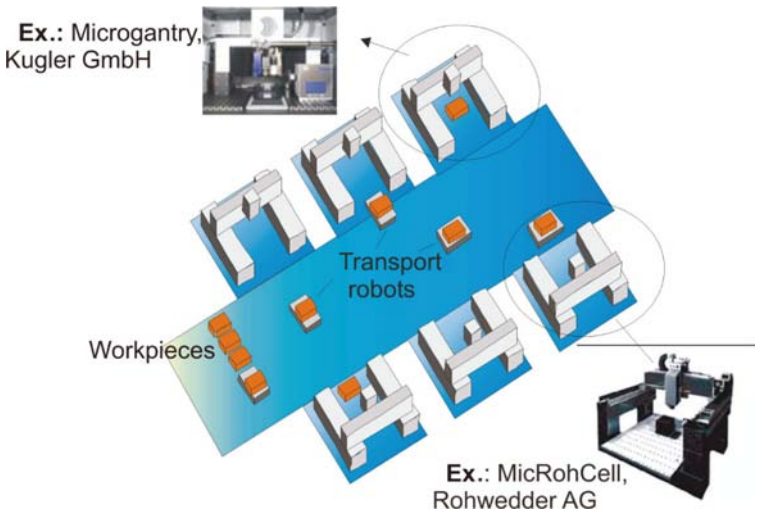

Figure 1 Structure of the proposed microproduction system with examples from two companies in Germany.

way. One special class of interaction mechanisms are market-based schemes like auctions or the distributed computation of any competitive equilibrium, which were extended and adapted here to address the problem in hand.

The next lower level of the functional control structure then included all coordination tasks on the multi-robot level that are necessary to fulfill the given production plan. The most important task here was the coordinated navigation of all robots within the given limited area of the microproduction system. Therefore, a suitable approach is the assignment of a navigation agent to each robot. These navigation agents have to communicate and coordinate the navigation task in a decentralized way in order to fulfill the production plan while also taking constraints like collision avoidance and dynamic constraints of the robots into account. Possible mechanisms for distributed agent-based navigation of a multi-robot system were developed and investigated. Since the solution of the navigation task can also influence the overall production plan, interaction between this and the higher level of the functional control structure was necessary.

The main contribution of this paper is the concept of a multi-agent system for the control of the overall flexible microproduction system. The task allocation and the robot coordination system with path planning are two main fundamental issues. Although multi-robot architectures separate them into different layers, relevant improvements may be expected from solutions that are able to concurrently handle them. This paper proposes such a complete solution for the mentioned microproduction system, which has not existed in previous works. An improved version of prioritized motion planning on a global long-term level was also integrated as the other important contribution from this paper.

MATERIALS AND METHODS

Task allocation multi-agent system

Auctions are market-based and are the most common mechanism used in task allocation approaches (Gerkey and Mataric, 2004; Jiang and Zhang, 2011). In an auction, a set of items is offered by an auctioneer in an announcement phase and the participants can make an offer for these items by submitting bids to the auctioneer. Once all bids are received or a pre-specified deadline has passed, the auction is then cleared in the winner determination phase by the auctioneer who decides which items to award and to whom. In the proposed system, the items for sale are transportation tasks. The auctioneer is the system and the transport robots are the bidders. The bid prices reflect robot costs or utilities associated with completing a task. The auction can allocate tasks to the robots with the lowest costs for performing them and the overall goal is to minimize some global cost function.

The first proposed system of the agent-based task allocation system consists of three main agents: Orders, Workers and Machines. Orders represent the manufacturing process steps that need to be done in order to produce a number of the final products with limited budgets. The total number of steps can be obtained after calculation of the elements of Orders. Workers represent the transport robots in the manufacturing system. Machines represent the machines in the proposed system. From this simple structure, a complex and flexible manufacturing system can be applied.

An initialization of the system is needed at the beginning. Some data have to be provided regarding the real microproduction system. For example, the production speed of a machine as well as the speed of a transport robot has to be set. The details for each step of production have to be defined in the configuration period as the parameters in the agent-based system. The transportation tasks will be assigned to proper mobile transport 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 by the task allocation multi-agent system. 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 is a coordinated detailed path planning algorithm on the multi-robot level which leads to optimal individual paths under the constraints that collision points are avoided.

Microproduction three dimensional simulation and experimental test bed

The proposed microproduction system has some special characteristics with respect to the group of autonomous mobile robots. The multi-robot transport infrastructure allows for very flexible and even parallel interconnection of the different stationary microproduction machine tools. As a manufacturing facility, it is an indoor environment with a defined structure, that is, 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—a "clean room" scenario. Since this manufacturing infrastructure is fixed, it is assumed that a Cartesian map of the environment is defined and available to each robot.

To demonstrate the working scenarios for a proposed microproduction system, a three dimensional (3D) simulation is needed (Figure 2). This simulation is used to test the algorithm for task allocation methods of the teams of mobile robots in order to investigate the methods more clearly to understand what is going on in the production system as well as the behavior of the motion planning approach. Webots Professional Software (Olivier, 2004) was used as a main tool with MATLAB (Version 7.11.0.584; The MathWorks, Inc.; Natick, MA, USA). Webots can be integrated with a high level multi-agent system to form a total microproduction system. Later, the solution can be adapted into a real-world system with real mobile transportation robots.

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 upon a grid-map of the environment and a computation of the shortest path on the grid using

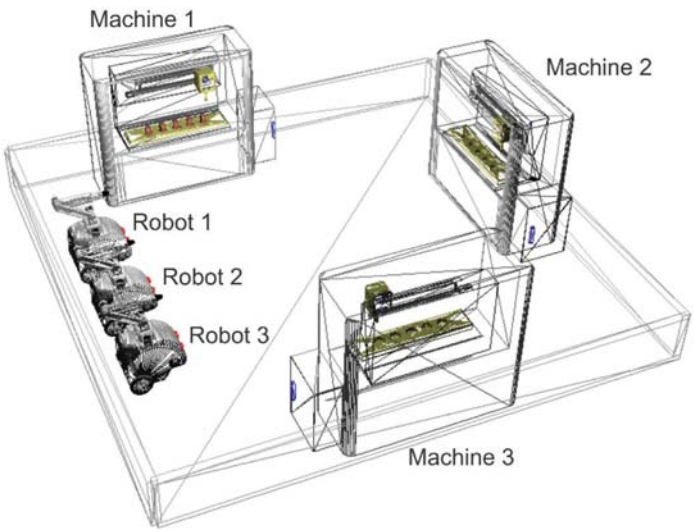

Figure 2 Test scenario using Webots: Assigning transportation tasks to each robot by a multi-agent system (Olivier, 2004).

computationally efficient algorithms. This does not include any velocity or acceleration constraints on the robots. The single robots can then publish their individual optimal paths on the blackboard. All robots can access this blackboard and are looking for any points where more than one robot 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 a new local path recalculated while the former collision grid point is blocked for them within the next shortest path calculation.

After this recalculation, the result is a set path for the robots, where only collision points of a maximum of two robots occur. These situations are then resolved on a local level using a global long-term and a model-predictive approach.

Global long-term motion planning approach

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 an optimal way, the robots have to keep a safe distance from each other and also velocity constraints have to be fulfilled. A multi-robot system is assumed 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 $r_i(k)^T = (x_i(k), y_i(k))$ at discrete time steps $k\Delta T$ with a fixed unique time interval ΔT in between. Between the waypoints, the robot is moving with a fixed velocity vector $v^i(k)^T = (v_{ix}(k), v_{iy}(k))$ and a simple discrete-time dynamic model of robot i is given by Equation 1:

$$r_i(k+1) = r_i(k) + \Delta T v_i(k),\ i \in \{1, \ldots, n\} \quad (1)$$

This modeling approach has the advantage that the positions of all robots at any given discrete time step k can be compared. As previously mentioned, the transportation task of robot i is defined by the start position r_{iS} and the destination position r_{iD} at the latest arrival time step $k = K_i$. All robots now can fulfill the transportation task in an optimal way, for example, using a minimal amount of energy and finally minimizing the distance to the destination position. If $V_i^T = (v_i(0), \ldots, v_i(K_i - 1))$ denotes the vector of all velocity vectors of robot i on its path and $R_i^T = (r_i(0), \ldots, r_i(K_i))$ denotes the vector of all waypoints, this can be expressed as Equation 2 being an optimization problem with the objective function $J_i(V_i, R_i)$:

$$\min_{\{V_i, R_i\}} J_i(V_i, R_i) = \sum_{k=0}^{K_i - 1} \left| v_i(k) \right|^2 + \left(r_i(K_i) - r_{iD} \right)^2 \quad (2)$$

The constraints of this optimization problem are initally the equations of motion given by Equation 1 which can be defined as a set of linear equality constraints in the form $g_i(V_i, R_i) = 0$. 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 $h_i(V_i) \le 0$. 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 safe 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 robot that moves first, having the highest priority can therefore plan its motion without any safety constraints. The obtained optimal path, (the vectors V_1^* and R_1^*) are posted on the blackboard and can be accessed by Robot 2. This robot has to accept the path and velocities of Robot 1 as given and has to optimize its path by taking further nonlinear inequality constraints into account. Robot 3 then has to take the two higher priority path vectors into account, and so on. 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 determine their respective optimal path vectors R_j^*. Then, the set of nonlinear inequalities considering the safe distance for robot i can be expressed as Equation 3:

$$|r_i(k) - r_j^*(k)| \geq \delta \quad \forall j, \forall k \tag{3}$$

where δ denotes the safe distance between the robots at any given discrete time step k. This can be expressed more compactly as the nonlinear inequality constraints denoted by $h_{i,\delta}(R_1^*, \ldots, R_{i-1}^*, R_i) \leq 0$. Regarding the optimization problem of robot i, the only variable that must be optimized is R_i and the motion planning problem of robot $i \in \{1, \ldots, n\}$ can be written as Equation 4:

$$\begin{aligned} &\min_{\{V_1, R_1\}} J_i(V_i, R_i) \\ \textit{such that} \quad &g_i(V_i, R_i) = 0,\ h_i(V_i) \leq 0 \\ &h_{i,\delta}(R_1^*, \ldots, R_{i-1}^*, R_i) \leq 0 \end{aligned} \tag{4}$$

The optimization problem in Equation 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 Equation 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.

Integrated model-predictive path planning, following and collision avoidance

The model predictive motion control approach is implemented in both simulation environments and on a test bed. A special multiple-shooting-based dynamic optimization package MUSCOD-II (Diehl, 2003) was applied. On the real-time motion control level, each robot has to follow the desired long-term path with the desired velocity between the waypoints. The robots have to compensate for 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, where the intersections of the global long-term optimal paths of the robots are concerned, it becomes obvious that possible intersections of the paths mainly occur for pairs of robots. Without any loss of generality, therefore, only two robots, 1 and 2 are considered, 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 Robot 1 under consideration describes the task to follow the given path currently defined by the two waypoints $r_1(i)$ and $r_1(i+1)$ 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 above a variable.

For motion control, first the dynamic behavior of each robot has to be specified 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 a sideways direction and thus have an underlying non-holonomic property. The posture, that is, the

position and orientation of the robot in a Cartesian *x*-*y*-coordinate system, is described by the set of kinematic equations defined as Equation 5:

$$\begin{aligned}
\dot{\tilde{x}}_1 &= \tilde{v}_1 \cos\tilde{\theta}_1 \\
\dot{\tilde{y}}_1 &= v_1 \sin\tilde{\theta}_1 \\
\dot{\tilde{\theta}}_1 &= \tilde{\omega}_1 \\
\dot{\tilde{s}}_1 &= \tilde{v}_1 \cos\tilde{\theta}_1 - \varphi_1(i) \\
\dot{\tilde{d}}_1 &= \tilde{v}_1 \sin\tilde{\theta}_1 - (i)
\end{aligned} \tag{5}$$

where $\tilde{v}_1$ and $\tilde{\theta}_1$ are the heading velocity and angle of the robot, $\tilde{\omega}_1$ is the angular velocity, $\tilde{r}_1^{\mathrm{T}} = (\tilde{x}_1, \tilde{y}_1)$ is the current position vector of Robot 1, $\tilde{s}_1$ is the distance traveled along the path direction starting in the last waypoint $r_1(i)$ to the next waypoint $r_1(i + 1)$ on the grid map and $\tilde{d}_1$ is the current orthogonal distance between the robot and the path. The orientation of the path segment between the neighboring waypoints is given by the angle $\varphi_1(i)$.

As previously described, the distributed global path planning algorithm results in situations where the considered Robot 1 can meet Robot 2. Without loss of generality, it is assumed 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 Robot 1 and Robot 2, where Robot 1 has to avoid the collision, is shown in Figure 3. The collision avoidance constraint between any two robots is given as the distance $\tilde{R}_{12}$, which must never be smaller than a defined security threshold δ in Equation 6:

$$\tilde{R}_{12} > \delta \qquad \forall t \tag{6}$$

The real-world microproduction environment requires more detailed application-specific differential constraints than other applications. Since the robots have to carry and transport extremely small parts in palette systems, acceleration both in travel direction ($\tilde{a}_{1x}$) and perpendicular to the travel direction ($\tilde{a}_{1y}$) must be limited, as well as the velocities and turning rates (using the four constraints listed as Equation 7):

$$\begin{aligned}
-\tilde{a}_{1y,max} < \tilde{a}_{1y} &= \tilde{v}_1\tilde{\omega}_1 < \tilde{a}_{1y,max} \\
-\tilde{a}_{1x,max} < \tilde{a}_{1x} &= \dot{\tilde{v}}_1 < \tilde{a}_{1x,max} \\
-\tilde{\omega}_{1,max} < \tilde{\omega}_1 &< \tilde{\omega}_{1,max} \\
-\tilde{v}_{1,max} < \tilde{v}_1 &< \tilde{v}_{1,max}
\end{aligned} \tag{7}$$

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, a discrete-time version of the underlying dynamic model on the control level is developed. The vector of state variables of Robot 1 is also defined as $\tilde{q}_1^{\mathrm{T}} = [\tilde{x}_1, \tilde{y}_1, \tilde{\theta}_1, \tilde{s}_1, \tilde{d}_1]$, and the vector $\tilde{u}_1^{\mathrm{T}} = [\tilde{v}_1, \tilde{\omega}_1]$ as the vector of input variables. The state variable differential equations are then given by Equation 5. Now the Euler approximation is applied 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 (Equation 8):

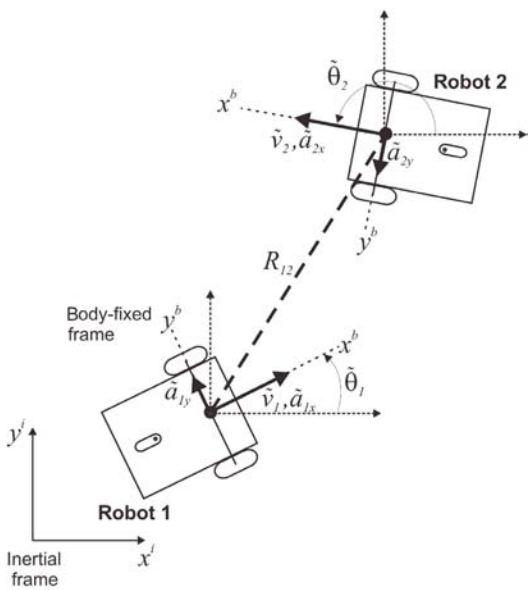

Figure 3 Engagement geometry of two mobile robots where R_{12} is the collision avoidance constraint, $\tilde{v}_1$ is the heading velocity, $\tilde{\theta}_1$ is the heading angle, and $\tilde{a}_1$ is the acceleration of Robot *i*.

$$\dot{\tilde{q}}_1 = \frac{\tilde{q}_1(k+1) - \tilde{q}_1(k)}{\Delta\tau} \tag{8}$$

where k denotes a discrete time step and in the following, $\tilde{q}_1(k)$ and $\tilde{u}_1(k)$ denote the discrete-time vectors of state and input variables of Robot 1. The set of differential equations defined in Equation 5 is then converted into a set of algebraic equations (using the notation of the input and state variables). The conversion of the first differential equation in the set defined in Equation 5 is shown in Equation 9:

$$\tilde{q}_{11}(k+1) - \tilde{q}_{11}(k+1) - \Delta\tau\left(\tilde{u}_{11}(k)\cos\tilde{q}_{13}(k)\right) = 0 \tag{9}$$

where $\tilde{q}_{11}$ denotes the element i of the vector $\tilde{q}_1$ of the state variables. The differential constraints defined in Equation 7 can be re-formulated, with the conversion of the second equation defined in Equation 7 producing Equation 10:

$$-\tilde{a}_{1x,max} < \frac{\tilde{u}_{11}\left(k+1\right) - \tilde{u}_{11}(k)}{\Delta\tau} < \tilde{a}_{1x,max} \forall k \tag{10}$$

In the same way the constraints describing the collision avoidance task defined in Equation 6 can be re-formulated.

Assume that at $t = 0$ (and hence $k = 0$) Robot 1 and Robot 2 have the initial vectors of state variables $\tilde{q}_1(0)$ and $\tilde{q}_2(0)$ and both robots have to follow a path with given current path angles $\tilde{\varphi}_1\left(i\right)$and $\tilde{\varphi}_2(j)$, respectively. The proposed algorithm then works as follows. For a given time horizon of K time steps, and trajectories of input and state vectors $\tilde{Q}_2 = [\tilde{q}_2\left(1\right),\ldots,\tilde{q}_2\left(K+1\right)]$ and $\tilde{U}_2 = [\tilde{q}_2\left(01\right),\ldots,\tilde{q}_2\left(K\right)]$, 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, the objective function (Equation 11) is applied:

$$J_2\left(\tilde{U}_2,\tilde{Q}_2\right) = \sum_{k=1}^{K+1}\left(\frac{\tilde{q}_{24}\left(k+1\right) - \tilde{q}_{24}\left(k\right)}{\Delta\tau} - v_{2D}\right)^2 - \left(\tilde{q}_{25}\left(k\right)\right)^2 \tag{11}$$

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 $g_2\left(\tilde{U}_2,\tilde{Q}_2\right) = 0$. The problem-specific differential constraints in discrete-time formulation according to Equation 10 can be given as a set of linear inequality constraints $h_2\left(\tilde{U}_2\right) < 0$. Therefore, the optimization problem of Robot 2 finally yields Equation 12:

$$\min_{\{\tilde{U}_2,\tilde{Q}_2\}}\left(J_2\left(\tilde{U}_2,\tilde{Q}_2\right)\right) \tag{12}$$

$$\textit{such that} \quad \tilde{g}_2\left(\tilde{U}_2,\tilde{Q}_2\right) = 0, \tilde{h}_2\left(\tilde{U}_2\right) < 0$$

The results are the sets of optimal input and corresponding vectors of state variables over the considered horizon given by $\tilde{U}_2^*$ and $\tilde{Q}_2^*$. 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 $\tilde{Q}_2^*$. 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 taking the collision avoidance problem into account. This adds a further set of nonlinear inequality constraints given by $h_{1,\delta}(\tilde{Q}_2^*,\tilde{Q}_1) \le 0$ according to Equation 6. With the information about the future behavior of Robot 2 given by $\tilde{Q}_2^*$, Robot 1 now solves the following nonlinear static optimization problem (Equation 13):

$$\min_{\{\tilde{U}_1,\tilde{Q}_1\}} J_1\left(\tilde{U}_1,\tilde{Q}_1\right) \tag{13}$$

$$\textit{such that} \quad \tilde{g}_1\left(\tilde{U}_1,\tilde{Q}_1\right) = 0, \tilde{h}_1\left(\tilde{U}_1\right) < 0$$

$$h_{1,\delta}(\tilde{Q}_2^*,\tilde{Q}_1) \le 0$$

After the calculation of the trajectories of optimal vectors of input variables $\tilde{U}_1^*$ and $\tilde{U}_2^*$, only the optimal steering commands $\tilde{u}_1^*(0)$ and $\tilde{u}_2^*(0)$ 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 implemented.

The reason for this 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, realized by communicating robots.

RESULTS AND DISCUSSION

The first experimental result is a task assignment for each robot and machine in order to complete the given production order, as a task assignment diagram from the agent system on the high level control. The time slot of the tasks assignment for robots and machines is based on the properties of each robot (speed, resources, etc.), and the real-world characteristic of the proposed microproduction system (sample order, production speed of each machine, position, etc.) given in the initializing state.

For a realistic simulation, a 3D simulation was developed using Webots to demonstrate working scenarios for a proposed microproduction system. Each scenario can be observed and the proper algorithm developed for the full system from high level to low level control.

The 3D simulation of the motion planning approach after giving the tasks to mobile transport robots though the time frame, as depicted in Figure 4, are promising and underline its efficiency. Robot 2 bids for a task to go to Machine 2 to transport an item, and Robot 1 wins a task to go to Machine 3 after Robot 2 starts moving. So, Robot 2 has a higher priority than Robot 1 at this time. Robot 2 moves according to its desired shortest path. Robot 1 first calculates its own path and tries to minimize the deviation from the desired path. Then it has to start avoiding the approaching Robot 2. This 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 until it reaches Machine 3. The full procedure can be summarized as steps A to F: (A) The current discrete time is set to $k = 0$. Both Robot 1 and Robot 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 (global localization system). (B) Both robots determine the current distance $\tilde{d}_1(0), \tilde{d}_2(0)$ to the respective paths and the internally stored global paths given by waypoints. The initial value of s can be easily set to $\tilde{s}_1(0) = \tilde{s}_2(0) = 0$. (C) Robot 2, with a higher priority, solves Equation 12 with the initial values, and obtains the optimal trajectories $\tilde{U}_2^*$ and $\tilde{Q}_2^*$ for the time horizon of K time steps. (D) Robot 2 communicates the optimal trajectory

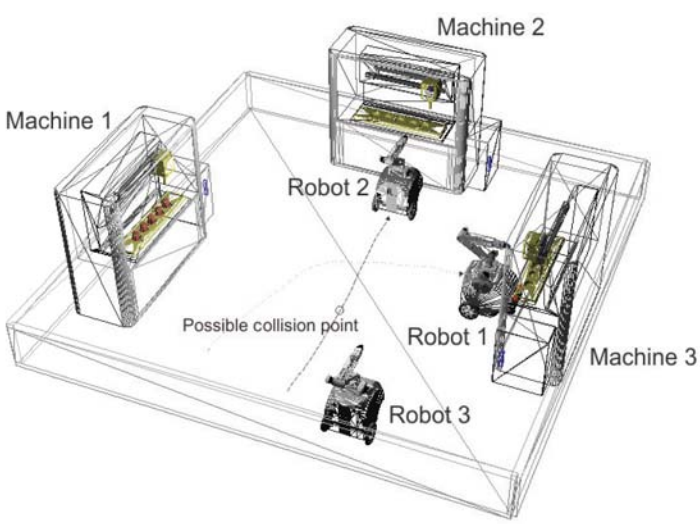

Figure 4 Result of a test scenario in Webots, where Robot 1 and Robot 2 reach their targets (Olivier, 2004).

of the state variables $\tilde{Q}_2^*$ to the blackboard where this information is read by Robot 1. (E) Robot 1 uses $\tilde{Q}_2^*$ in order to solve the combined problem, equation 13, to obtain the optimal trajectories $\tilde{U}_1^*$ and $\tilde{Q}_1^*$ for the time horizon of K time steps. (F) Both robots realize the optimal steering commands $\tilde{u}_1^*(0)$ and $\tilde{u}_2^*(0)$ for the current time step. Then they proceed with Step 1 again.

The result of the model predictive approach is depicted in Figure 5. The collision avoidance constraints are always fulfilled. In addition, the security threshold defined with regard to the size of the two robots 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_{R1x}) and perpendicular to the travel direction (a_{R1y}), have been limited, as well as the velocities and turning rates.

CONCLUSION

The approach worked well for this specific application with some limitations. The transportation problem was solved in a simple manner, so that if any robot starts its transportation task, it can be assumed that robots already in motion have a higher priority. Therefore, the 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 considered 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 (Ogay *et al*., 2012). One useful approach is the discretization of the constraints by using a simplified discrete-time model of the robotic motion. In this study, the result of the global long-term decoupled planning under simplified differential constraints was 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.

The agent-based systems sit on top of the system to provide customization and adaptation to the system. The overall system can be implemented

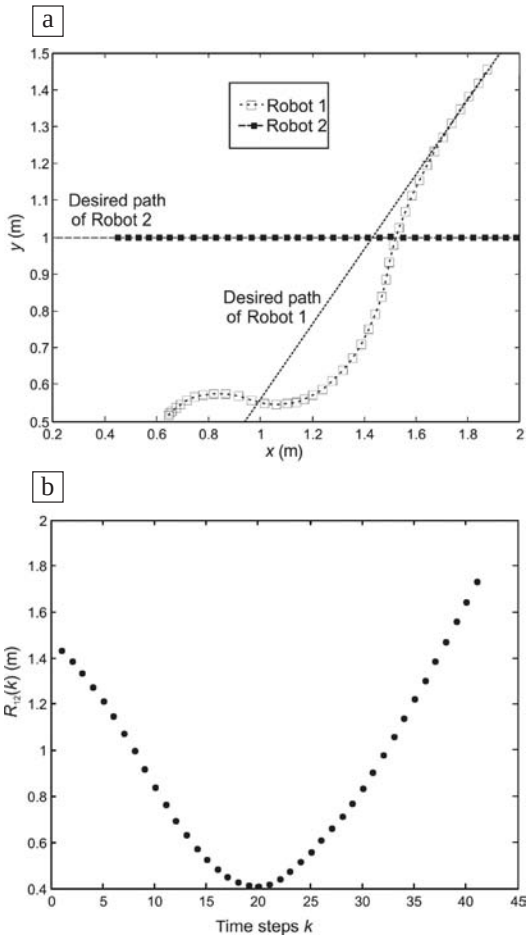

Figure 5 Result of the local real-time motion control: (a) Robot 2 moves according to its desired path. Robot 1 has to avoid the approaching Robot 2; (b) the collision avoidance constraint R_{12} are limited at 0.4 m. of k time steps.

in any flexible production systems in the future. This total solution and model predictive motion control approach has been tested by simulation, as well as on a test station platform, with many working scenarios. The results have been discussed with the microproduction companies in Germany for further implementation with real industrial robots and working machines.

ACKNOWLEDGEMENTS

The authors would like express their profound gratitude to Sirindhorn International Institute of Technology (SIIT), Thammasat University (TU) and the National Research University Project, Thailand Office of Higher Education Commission, for financial support.

LITERATURE CITED

Bhattacharya, S., R. Ghrist and V. Kumar. 2012. Multi-Robot coverage and exploration in non-Euclidean metric spaces, pp 1–16. *In* **Proceedings of the Tenth International Workshop on the Algorithmic Foundations of Robotics.** 13-15 June 2012. Cambridge, MA, USA.

Demazeau, Y., P. Pawlewski, V.J. Julián, F. Fdez Riverola, E. Corchado, R. Corchuelo, J. Bajo, J.M. Corchado Rodríguez, F. Dignum and A. Campbell. 2010. **Trends in Practical Applications of Agents and Multiagent Systems.** 8th ed. International Conference on Practical Applications of Agents and Multiagent Systems. Springer-Verlag. Germany. 700 pp.

Diehl, M. 2003. Real-time optimization for large scale nonlinear processes. **VDI-Verlag 2003.** Düsseldorf, Germany.

Dias, M.B., M. Zinck, R. Zlot and A. Stentz. 2004. Robust multirobot coordination in dynamic environments, pp. 3435–3442. *In* **Proceedings of IEEE International Conference on Robotics and Automation (ICRA).** Carnegie Mellon University. Pittsburgh, USA.

Ducatelle, F., F. Di Caro, C. Pinciroli and L.M. Gambardella. 2011. Self-organized cooperation between robotic swarms. **Swarm Intelligence** 5(2): 73–96.

Gerkey, B.P. and M.J. Mataric. 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. **The International Journal of Robotics Research** 23: 939–954.

Jiang, L. and R. Zhang. 2011. An autonomous task allocation for multi-robot system. **Journal of Computational Information Systems** 7(11): 3747–3753.

LaValle, S.M. 2006. **Planning Algorithms.** Cambridge University Press. Cambridge, UK. 1023 pp.

Mastellone, S., D.M. Stipanovi´c, C.R. Graunke, K.A. Intlekofer and M.W. Spong. 2008. Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. **The International Journal of Robotics Research** 27(1): 107–126.

Ogay, D., J.H. Ryu and E.G. Kim. 2012. A Multiresolution Approach for Real-Time Motion Planning under Differential Constraints, pp. 745–754. *In* **Proceedings of the 12th International Conference IAS-12**.

Olivier, M. 2004. Webots: Professional mobile robot simulation. **Journal of Advanced Robotics Systems** 1: 39–42.

Shaikh A. and A.D. Dhale. 2013. AGV routing using Dijkstra's Algorithm: A review. **International Journal of Scientific & Engineering Research** 4(7): 1665–1670.

Srinivasan, D. and L.C. Jain. 2010. **Innovations in Multi-Agent Systems and Application – 1**. Springer-Verlag. Düsseldorf, Germany. 298 pp.

Voos, H. 2008. Agent-based automation of flexible microproduction systems. **VDI Kongress AUTOMATION 2008.** Baden-Baden, Germany. 549 pp.