Motion Planning for Mobile Robots in Microproduction under Special Constraints

P. Ziemniak*, A.Paczynski, H. Voos***, D. Uciński***

** Department of Electrical Engineering, Computer Science and Telecommunications, University of Zielona Góra, Poland, (e-mail: pziemnia@willow.iie.uz.zgora.pl, D.Ucinski@issi.uz.zgora.pl)*
*** Faculty of Technology and Management, Hochschule Ravensburg-Weingarten, Germany, (e-mail: paczynski@hs-weingarten.de)*
**** Faculty of Electrical Engineering and Computer Science, Hochschule Ravensburg-Weingarten, Germany, (e-mail: holger.voos@hs-weingarten.de)*

Abstract: The paper outlines a method for robot motion planning. The proposed algorithm involves some movement constraints and predefined trajectories. The additional constraints allow limiting the acceleration and centrifugal forces. Limiting these forces not only helps saving energy, but helps also lowering uncertainty related to the robot movement and secures transported materials. These factors are important in microproduction applications where mobile robots are used. Various materials are transported and some might require a special treatment e.g. dangerous materials, liquids in open containers, etc. The paper considers also the use of iterative dynamic programming for optimal control. The research is based on a novel mobile robot, but in the future will be supported also on the currently under design Max3D robot that has some special attributes mentioned in the paper.

Keywords: motion, planning, mobile, robots, optimization, microproduction

1. INTRODUCTION

New areas of application in manufacturing like mass customization or microproduction require completely new manufacturing concepts with highly flexible structures and processes. In microproduction, i.e. in the production of micro-scale components and products, one of the biggest challenges besides extreme precision requirements and parts in the micrometer scale is the fact that extremely small batch sizes (up to just only one single part per order) of highly customized products must be produced. Therefore, economic microproduction requires a high degree of automation and a manufacturing system that can be reconfigured and adapted to changing production processes in real time.

One possible solution is the integration of mobile autonomous robots that are able to interconnect stationary machine tools or assembly stations in a very flexible way. This systems structure allows the fast adaptation to a broad range of possible microproducts by forming the different required sequences of microproduction and assembly processes. However, this new approach comes at the expense of the underlying integration of the autonomous robots in the overall automation concept which, in turn, requires new solutions for resource allocation, scheduling and also robot navigation and control.

One problem that especially arises in these microproduction systems is the need for a careful but nevertheless fast transport of extremely small workpieces. These workpieces should not be disordered too much and therefore the accelerations both in travel direction and perpendicular to it as well as velocities and turning rates are limited. In addition, the robots should move in an energy-efficient manner in order to increase the operating time with one battery charge and have to avoid obstacles.

Therefore, this contribution focuses on robot motion planning and control where all aspects of the special transportation task should be considered in parallel, especially the differential constraints such as limitations imposed on accelerations and velocities. The paper is organized as follows. Sections 1 provides a theoretical background for the problem and outlines possible solutions. The second section briefly presents objects (robots) that are used for testing the proposed solution. The next section provides details of the method together with examples and sample results. The last two sections present plans for further research.

2. THEORETICAL BACKGROUND

The motion planning algorithms can be roughly divided into two main groups. One of them covers local navigation, or reactive planning and the other covers global navigation or deliberate planning [Siegwart and Nourbakhsh, 2004; Dudek and Jenkin 2000; Choset et al. 2005; Chen et al. 2008]. The first group is well known for algorithms like the bug algorithm or potential fields method [Choset et al. 2005]. They generate locally optimal paths and adopt well and quickly to changes in the environment. The other group applies mainly optimal control techniques [Bryson, 1999], and occasionally methods related to computational geometry, e.g. Voronoi diagrams, or artificial intelligence. Those methods, on the other side, usually construct a path that does lead to a global minimum but requires extensive computations. Since algorithms in both groups have their advantages and disadvantages, a hierarchical approach that is a compromise between them is proposed. In our approach, we try to develop a methodology based on the hierarchical approach. First of all, we wish to reduce the complexity of path optimization by introducing some predefined trajectories. Consequently, some constraints might be removed speeding up the path generation process. The predefined trajectories might be customized for a particular robot or task for better performance. Second, we try to impose some additional constraints on the optimization that improve the reliability of the robot following the trajectory. Our research is supposed to save the space necessary for maneuvers, what is very important in production halls filled with lots of machines and various stands. Two other important factors are time and energy regarding both the robot or the whole production. In the current phase of our research we focus on optimizing the time, but in a typical production scenario the time to get from one station to another may depend on many factors. In such situations, a robot can save energy by travelling with limited speed and acceleration.

3. RESEARCH OBJECTIVE

Our considerations concern a special design production robot that is characterised by a very high manoeuvrability. In the current phase we consider the first version of the robot, but later we would like to continue with its 3D version that is able to additionally change the platform inclination. The main goal is to optimize the path regarding not only the time required to follow it, but also by limiting external forces acting on the transported materials (acceleration, centrifugal force, etc.) and limiting the total energy consumption.

Current research is supported by experiments on a currently developed Production Robot. The robot has the size of a Euro Palette and is dedicated to production environments (Fig. 1(a)). Thanks to an innovative steering principle, the robot is distinguished by a very high manoeuvrability which allows for an extraordinarily high number of manoeuvres which can be performed from any initial pose.

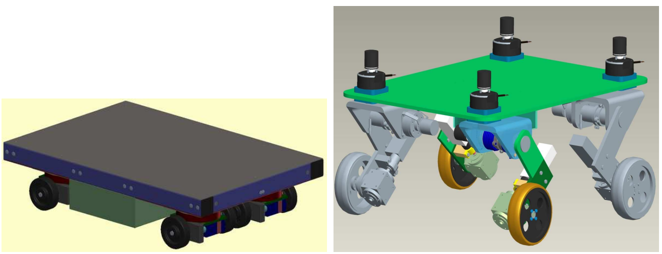

Fig. 1. (a) The Production Robot, (b) The Max 3D robot

Future research will be supported by the currently under design robot Max 3D (Fig. 1(b)). The robot will have an additional functionality allowing for a platform inclination. Such an ability will push us forward since it will be possible to actively counteract acceleration and centrifugal forces by properly inclining the platform at each point of the trajectory.

Both robot designs originate from an innovative and patented steering solution that has been thoroughly investigated for the recent years [Stetter et al. 2008]. The solution results in high manoeuvrability of the robot that can be well utilized in limited spaces that are typical for microproduction.

4. CURRENT RESEARCH

We have two main assumptions that form a basis for our research. The first relies on limiting the total acceleration that is composed from linear and tangential accelerations (centrifugal force) and can be expressed as follows:

$$\sqrt{\left(\left|\vec{a}_{\text{normal}}\right|\right)^2 + \left(\left|\vec{a}_{\text{tangential}}\right|\right)^2} \le \left|\vec{a}_{\text{total}}\right| \tag{1}$$

This limitation implies an ability to limit the forces acting on the transported materials. It is important to note that in the current phase we try to optimize only the length of the acceleration vector. In the future we will consider also the change in its direction. Both the acceleration/braking force and the centrifugal force can be limited by a value that is appropriate for the robot at hand, materials being transported or the environment the robot moves in. For solid materials the acceleration might be high, but for e.g. liquids transported in open containers the value will have to be small. Similarly, when the environment is characterized by low traction, the coefficient has to be small in order to limit the possibility of slipping.

This limitation also concerns the relation between the velocity and the curvature:

$$V_{\max} = \sqrt{\left|\vec{a}_{\text{total}}\right| R} \tag{2}$$

The equation yields a maximal velocity that is feasible for the current curvature. The larger curvature (smaller radius R), the smaller velocity can be used. Obeying this maximal velocity will ensure that total acceleration will not be outstripped and materials can be safely transported. Transforming the above equations, it is possible to calculate a feasible acceleration while already on a curve with radius R and having velocity $V < V_{max}$:

$$\left|\vec{a}_{\text{tangential}}\right| \le \sqrt{\left(\left|\vec{a}_{\text{total}}\right|\right)^2 - \left(\frac{V^2}{R}\right)^2} \tag{3}$$

Owing to this assumption, it is possible to limit unwanted forces and make transportation more reliable. The second assumption concerns the shape of the trajectory. Preliminary research showed that the fastest point-to-point trajectory with maintaining the same velocity is a circular trajectory. Other trajectories require braking and accelerating in order to comply with the first assumption, i.e., adjusting velocity to the curvature radius given the maximal total velocity. The whole trajectory, when composed out of circles, should be smooth and should make it easier to apply the first assumption. In our research we will tend to trajectories that have as few radius changes as possible in order to keep them smooth.

Fig. 2 presents a circular trajectory from point (-1, 0) to (1, 0) with an obstacle at the origin of the coordinate system. In order to omit the obstacle, the trajectory has to cross the point (0, 1). The final and initial velocities are zero. The linear acceleration is limited by the value of a maximal total acceleration. The figure presents both the trajectory and its parameters such as velocity, tangential acceleration, total acceleration. It can be observed that velocity gradually raises to the maximal velocity allowed for this curvature and then is constant near the end where gradually decreases to zero. The total acceleration is constant and equal to a predefined value of *0.3*.

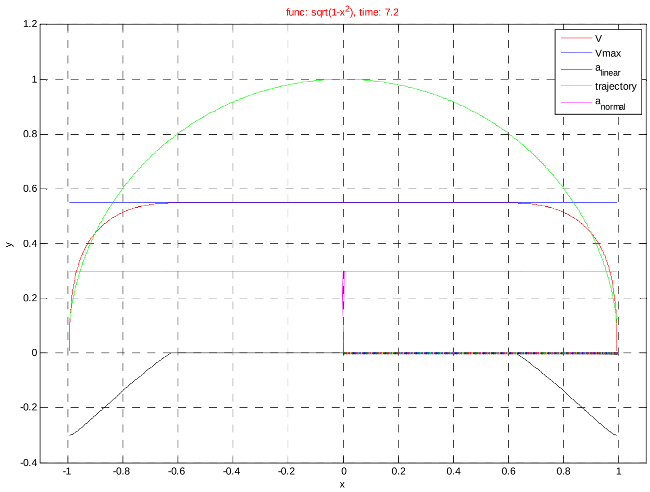

Fig. 2. Circular trajectory and its parameters

Similar analysis was performed for the same initial conditions for different curvatures like straight lines, cosine based, parabolic, etc. For our research we assumed circular trajectories that perform best having a constant radius and no unnecessary braking and accelerating sequences.

4.1 Base circular trajectory

In next phase we tried to find an optimal path from the origin of the coordinate system and with an arbitrary velocity vector to the point on the *x* axis with the same velocity but with its direction parallel to the *x* axis. The approach uses circular trajectories as presented in Fig. 3. Two tangent circles can be found that make the complete trajectory from point *A* to point *B*. There are however several scenarios that can be considered. First, the circles can be chosen so that the curvature is maximal for the actual velocity. In this scenario no further acceleration is possible. In order to account for the acceleration, a smaller curvature has to be taken and then an acceleration may occur.

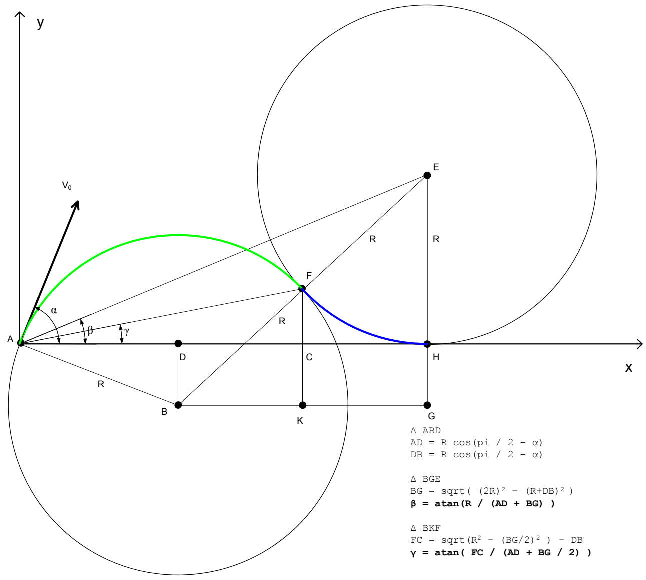

Fig. 3. Trajectory composed of two tangent circles

Following figures present two scenarios, one when acceleration starts immediately (Fig. 4) and the other when it starts right after the circular trajectory is over (Fig. 5). No method is optimal in all situations, as the result strongly depends on the initial velocity vector and radius taken.

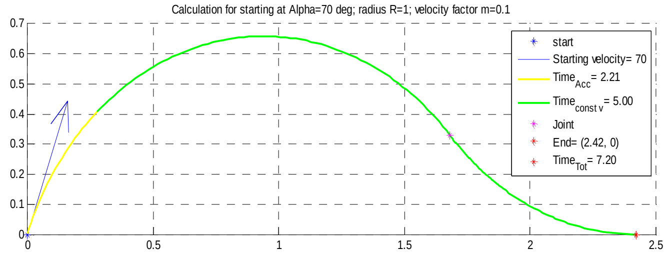

Fig. 4. Accelerating at the beginning of trajectory

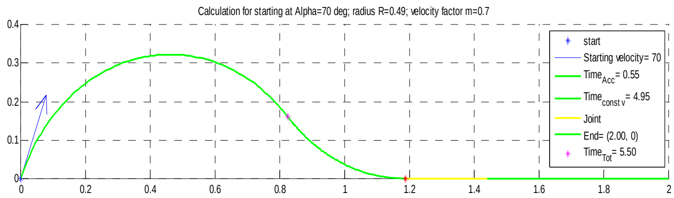

Fig. 5. Accelerating at the end of trajectory

4.2 Circular trajectory in pose reaching task

The next phase concerns the situation when from an arbitrary starting position and with a given starting velocity vector, the robot has to reach a certain position with a specified velocity vector. This situation is typical for a production scenario where a robot has to achieve the appropriate pose in order to load or unload goods from or to a machine. Depending on the machine, the robot might need to stop or can drive with a lower velocity while the robotic arm transports materials. The situation is typical also for a problem of catching an invader. The robot might be required to achieve some position and some velocity that is predicted to help stopping the invader. In such a scenario, besides good planning and prediction also a fast robot response is required.

The problem defined in this manner is in fact very common. The robot can be at any pose and can have any velocity while a new goal point is set (an obstacle encountered on the trajectory, a need to perform a different task, e.g. battery recharging). It might be important to stop the current task and follow to the new goal as soon as possible but taking into consideration the material that might be transported. There are several options how to act in this situation and choosing the appropriate one is essential in order to act optimally. A first option might be to stop, turn in place and follow the goal. Our research revealed, however, that the more braking in the trajectory, the less likely it is for the trajectory to be optimal. Braking requires time and costs energy. Another option is to enter the circular trajectory with the smallest radius that is appropriate to the actual velocity and accelerate to the maximal velocity as the trajectory becomes linear. An alternative option is to start accelerating on the curvature. This requires, however, taking a longer path (a bigger radius in order not to exceed the total acceleration limit). There are few more alternatives related to the exact point where the acceleration should start and what radius should be taken.

Our current studies concern the following situation: the robot travels from point P_S to P_K starting with velocity V_S and finishing with V_K as presented in Fig. 6. The trajectory is composed of two circles with radii (curvatures) corresponding to the starting and final velocities. The tangent line is also part of the trajectory, and if possible, only here the acceleration or braking can occur. In an ideal situation we start with V_S and go over the first part of the trajectory (first circle). Then, we start the second part (line) where we try to change the velocity from V_S to V_K (accelerating or braking). Finally, we enter the third part (second circle) with the correct velocity V_K and we can reach the final point P_K. The next figure (Fig. 7) presents the same situation after some period of time. The change in the starting position and velocity vector is a result of following the trajectory. The final position and velocity vector are dependent on the target the robot has to reach and it does not need to be constant.

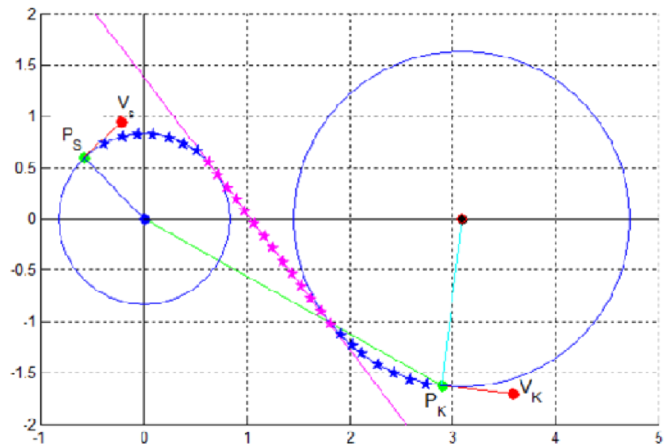

Fig. 6. Reaching required position and velocity

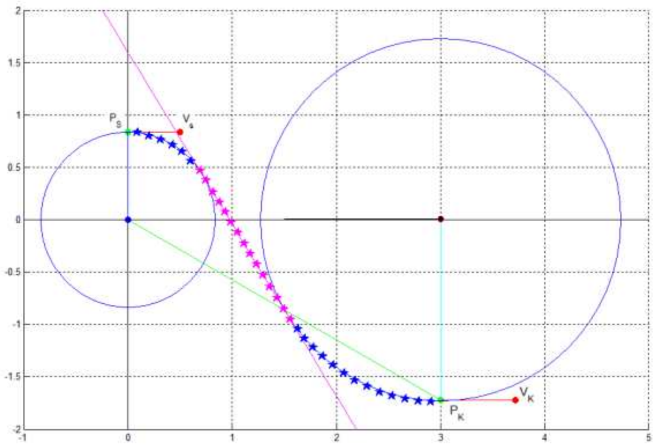

Fig. 7. Reaching required position and velocity (second step)

At any time instant, the situation is analogical. Only the positions and velocities may change, resulting in different curvatures and different motion scenarios.

4.3 Iterative dynamic programming

Since the problem of motion planning can be formed as an optimization task, a number of methods can be used to solve it. In our research we try to utilize a variant of approximate dynamic programming, i.e., Iterative Dynamic Programming (IDP). Dynamic programming is an approach whose main advantage lies in dividing an optimal control problem into a sequence of simpler problems that can be solved separately [Luus 2000]. It is well known that the approach is however prone to the course of dimensionality and finds application only for low dimensional problems. IDP, however, incorporates some approximations which make it possible to solve high dimensional problems, though at the cost of no guarantee of finding the global minimum [Luus 2000].

The problem we wish to solve by IDP can be formulated as follows [Luus 2000]. We consider a system described by a system of ordinary differential equations (they are going to describe dynamic properties of our mobile robot):

$$\dot{x}(t) = f(x,u,t) \tag{4}$$

where the initial state *x(0)* is known, x is an *(n x 1)* state vector and u is an *(m x 1)* control vector bounded by

$$\alpha_j \le u_j(t) \le \beta_j, \quad j = 1,2,\ldots, m \tag{5}$$

The optimal control problem is then to choose the control *u(t)* in the time interval $0 \le t \le t_f$ that minimizes the performance index [Bryson, 1999]:

$$\mathrm{I}[\mathrm{x}(0),\ \mathrm{t}_f] = \Psi(x(t_f)) + \int_0^{t_f} \phi(x,u,t)dt \tag{6}$$

Our IDP algorithm consists of the following steps:

1) Divide the problem into smaller ones. This is achieved by dividing the time intervals into P stages of equal lengths. In our motion planning task, stage $p=1$ represents the initial pose and $p=P$ the pose that has to be achieved by the robot.

2) Choose the number of test values for control u denoted by R, an initial control policy, an initial region size vector r_{in}, a region contraction factor γ, and the number of grid points N
3) Choose the total number of iterations B.
4) Set the region size vector $r^j = r_{in.}$
5) Repeat the following for each j from $1..B$:

Using the initial control policy generate N-1 alternative control sets.
Integrate the system (4) over the whole time interval for each set of controls generating N state trajectories. Store x at the beginning of each stage.
for each stage ***p*** backward from $P..1$
  for each grid point ***n*** from $1..N$
    Generate R alternative controls for current point and stage.
    for each control ***r*** from $1..R$
      Integrate (4) over one stage starting from p (use stored system state at the beginning of stage p)
      for each of the remaining stages ***q*** from p+$1..P$
        Find the grid point in stage q that is closest to the current system state and use already known best control for this point to integrate (4) again.
      end for
    end for
    Store the control that had the lowest performance index
  end for
end for
Store the trajectory that has a lowest performance index as initial control policy
shrink the region size vector $r^{j+1} = \gamma\, r^j$

The formula for generating alternative control vectors is given as [Luus 2000]:

$$u(P-1) = u^{*j}(P-1) + Dr^{j} \tag{7}$$

where $u^{*j}(P-1)$ is the best value found in previous iteration and D is a diagonal matrix composed of random numbers between -1 and 1.

A big advantage of the algorithm is the ease of including additional constraints and control vectors. The biggest problem is however to find the appropriate dynamic model of the robot. At the current stage of our research we provide results for the very simple model

$$\begin{aligned}\dot{x}(t) &= v(t)\cos\theta(t)\\ \dot{y}(t) &= v(t)\sin\theta(t)\\ \dot{\theta}(t) &= w(t)\end{aligned} \tag{8}$$

to illustrate the behaviour of the IDP algorithm in this type of problem. The synthesis of the model for the Max 3D is now under way. For the model (8) only constraints regarding maximal velocities v and w are considered (they are treated as controls). The performance index we used constitutes a combination of energy and the robot path length. Additionally, a penalty term is added to avoid obstacles. Fig. **8** presents the convergence of the algorithm over 30 iterations with the number of stages P=15, the number of test controls R=15, the number of grid points N=10, and the region contractor factor γ=0.9. The initial control region size was set to r_{in}=$[2, pi/4]$ where the first value corresponds to the robot velocity and second to its angular steering velocity. Additional control bounds were set up: $0..5$ for velocity and $-\pi/4..\ \pi/4$ for angular velocity.

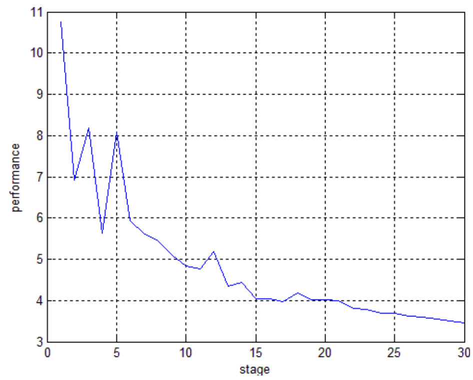

Fig. 8. Convergence of IDP over 30 iterations

Consecutive trajectories for the above-mentioned parameters are presented in Fig. 9. The initial control policy led to a linear trajectory that achieves the final point but crosses the obstacles (red circles) (Fig. 9(a)). The black lines are alternative trajectories that are generated from the initial control policy. The next panels in Fig. **9** present the current optimal trajectories after the first, sixth and fifteenth iteration. The trajectory with lowest performance index is marked by magenta colour.

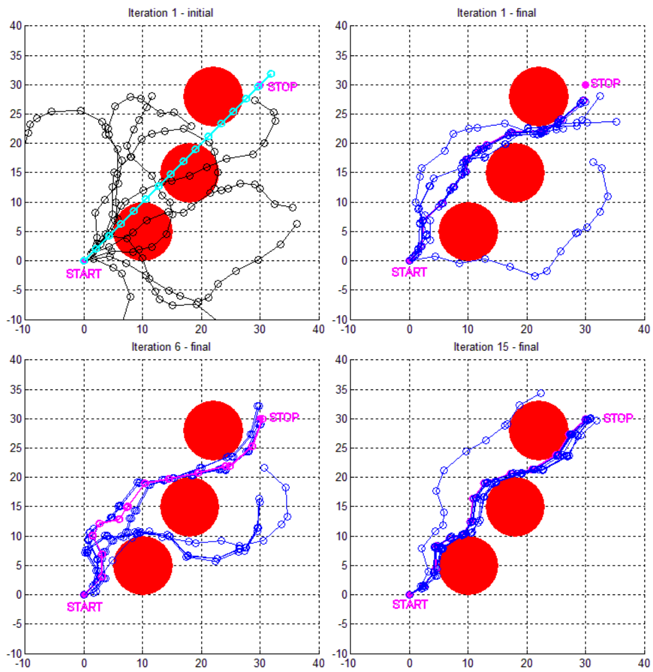

Fig. 9. Four selected steps of IDP; (a) initial trajectory; (b) iteration 1; (c) iteration 6; (d) iteration 15

It can be observed that trajectories differ much and explore quite well the region. Although not seen here, at each grid point (a small open circle) R test controls are evaluated, which in fact covers the region much more precisely. As time elapses, the region explored gets narrower and the trajectory becomes smoother. The final trajectory, however, is seldom smooth as the control region is composed of discrete values. The problem of non-smoothness and long computation time is known for the IDP. The algorithm, however, allows us to use any model and works well with models with long control vectors. We expect that the results will be more interesting after supplying the algorithm with the models of the real robots mentioned earlier. The robots are characterized by high manoeuvrability and high flexibility. We expect the generated trajectory to utilize well the advantages of the robots.

5. FURTHER RESEARCH

In the current phase of our research we have made several simplifications in order to achieve preliminary results as a basis for a more thorough analysis. Currently, we optimize only the length of the acceleration vector. In the future, however, we would like to extend the approach to include the change in the direction of the acceleration vector and to include system uncertainties. Those uncertainties originate from sensors inaccuracy or changes in the environment. In the production environment, it is expected that several robots (or also humans) will operate at the same space. In such a situation, in order to plan an optimal trajectory, it is necessary to analyze trajectories of other moving objects. Fig. 10 presents how the trajectory might change when an unknown object is approaching. Here, also the uncertainties are marked. The safe trajectory is characterised by the longest distances to the obstacles and a change in the trajectory of the approaching object. In the case of other trajectories, which are in a narrower confidence interval, the adaptation is not necessary and the trajectory is unchanged.

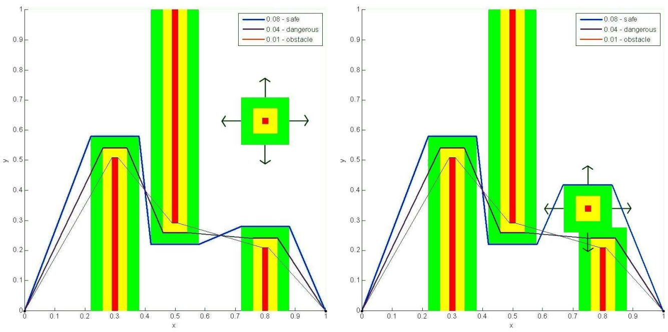

Fig. 10. Trajectory change caused by the moving object

The other simplification is the use of a predefined maximal value of total acceleration. Since it is dependant on many factors (e.g. traction, load) some special adaptation techniques or appropriate estimation is necessary as the parameter might vary largely over time.

From the other side we also want to continue our research on the IDP algorithm. After supplying it with the correct robot model we want to utilize all the benefits of special design of the robots. Besides constraints that can be added to the optimization process, there are also many parameters, improvements and variants for this algorithm that have to be analyzed and tested on the real robots.

As it was already mentioned in the article, the pose reaching task might be similar to the task of capturing an invader. In our future work, we want also to extend our research to this area. The final pose can be replaced by the predicted pose that can help catching the invader. The prediction of movement and learning the behaviour of other objects is a wide topic but can be an interesting combination with our approach for motion planning. Our approach provides easy and safe trajectories from one point to another, including the final velocity vector and additional constraints.

5. SUMMARY

The article presented an approach for generating trajectories for mobile robots. The contribution is targeted initially for microproduction purposes but can be further extended. The generated trajectories are characterized by limited tangential and normal acceleration which has positive impact not only on the movement of the robot but also on the way in which materials are transported. Thanks to the additional constraints imposed on the path optimization procedure, it is possible to limit the slip of the wheels and prevent manoeuvres that might lead to dangerous situations (e.g. entering a significant curvature with an excessive velocity). The same constraints increase also robot chassis stability during transportation. In this way transported materials does not necessarily have to be secured well for transportation. By providing correct parameters, it will be also possible to transport with ease liquids in open containers. The approach is very useful with robots that are able to achieve fast velocities and in limited spaces as it requires many manoeuvres to be performed. Future research will extend the methodology but will still focus also on iterative dynamics programming and go into a wider area of application.

REFERENCES

Bryson A.E.: Dynamic Optimization, Addison-Wesley, Menlo Park, CA, 1999

Chen, C.; Li H.C.; Dong, D. Hybrid Control for Robot Navigation - A Hierarchical Q-Learning Algorithm. In *Robotics & Automation Magazine, IEEE* , vol.15, no.2, pp.37-47, June 2008

Choset H., Lynch K., Hutchinson S., Kantor G., Burgard W., Kavraki L., Thrun S.: Principles of Robot Motion. Theory, Algorithms, and Implementation, The MIT Press, Cambridge, MA, 2005

Dudek G., Jenkin M.: Computational Principles of Mobile Robotics, Cambridge University Press, Cambridge, MA, 2000

Luus R: Iterative Dynamic Programming, Chapman & Hall/CRC, Boca Raton, FL, 2000

Siegwart R., Nourbakhsh I.R.: Introduction to Autonomous Mobile Robots, The MIT Press, Cambridge, MA, 2004

Stetter, R.; Paczynski, A.; Stania, M. and Zajac, M. Autonomes Fahrzeug mit innovativen, patentierten Lenksystem. EMA 2008 Elektromobilausstellung, Aschaffenburg, Germany 2008

Voos, H.: Automation of flexible Microproduction Systems. In Proc. of the VDI-GMA Conference „Automation 2008“, Baden-Baden, Germany, 2008.