Budapest University of Technology and Economics
Department of Control Engineering and Information Technology
Budapest, Hungary

OBJECT MANIPULATION PLANNING FOR DEXTROUS ROBOT SYSTEMS

Summary of PhD thesis by

Gábor Vass

Supervisors:

Prof. Dr. Béla Lantos
Budapest University of Technology and Economics
Dept. of Control Engineering and Information Technology
Budapest, Hungary

Dr. Shahram Payandeh
School of Engineering Science
Simon Fraser University
Burnaby, BC, Canada

Budapest, May 2005

1 Introduction

Moving an object with a robotic hand is a difficult problem. First collision free paths for all fingers of the hand have to be found, then forces in the contact points must be generated. Finally the generated path is executed considering the robot geometry and joint parameters.

Object manipulation first requires that the object be located within its environment. Then, the object must be initially grasped by the agents (cooperating robots and/or fingers of a dextrous hand). When the grasp is stable, the manipulation process can be started.

The manipulation task (called object reconfiguration problem) is stated as the following: given an initial grasp of the object find the motion's trajectories of the agents to move the object to the desired configuration. In general collision free paths for all agents must be found toward the contact points on the object (pre-grasp configuration) and the grasping and manipulation forces should then be exerted on the object by the agents. These forces are determined first to ensure a stable grasp, then to manipulate the object.

Several aspects of the basic mechanics of manipulating objects are explored by [16] and [17] including the contact modeling, hand mechanisms, force application and velocity analysis, stiffness control and sensing, manipulator grasping and pushing operations.

The fingertips of a dextrous robotic hand can be treated as manipulating agents with motion constraints, thus object manipulation algorithms with multiple agents can be applied to dextrous robotic hands.

Recently numerous object manipulation algorithms have been developed (see [12]). The topic of object manipulation with agents or robotic hands can be classified by the mode of the relative motion of the agents and the object. Pushing, rolling and sliding are examples of relative motions which the agents can produce with respect to the object. The relative motion between the fingertips and the object determines the whole motion of the system, if the desired motion of the object is known.

The object manipulation task is divided into three categories:

- the goal of the *object manipulation* task is to reach the desired object configuration without considering the contact configuration;

- the *grasp adjustment* attains the desired contact configuration by disregarding the object configuration;
- *dextrous manipulation* leads the robotic hand to its final state taking into account the desired object and contacts configuration, respectively.

The Department of Control Engineering and Information Technology (CEIT, former Process Control Department, PC) at the Budapest University of Technology (BUTE) launched a project on the construction and manufacturing of a new type of dextrous gripper in 1995. Details of mechanical construction, hand kinematics and dynamics were elaborated by Ludvig in 1997 [13]. The mechanical interface of the TUB-PC hand is rather universal, therefore the hand can be fitted any PUMA or SCARA type robots. The early control programs of the hand including a calibration tool, a path-planning program and decentralized PID controllers of the hand was developed by the author as an M.Sc thesis [18].

2 Scope and Goal of the Research Work

This Ph. D. Thesis deals with object manipulation and trajectory planning algorithms for dextrous robot systems including cooperating robots and multi-fingered hands. Three methods have been developed for object manipulation design and real-time trajectory planning.

The objective of the thesis is to develop a framework of a manipulation planner in the presence of obstacle constraints between the initial and goal configuration of the object. In this approach, fingertips can loose and regain contact with the object at some other planned next location. The controller devises a strategy where the environmental constraints (e.g. obstacles) can also be used as passive contact surfaces supporting the manipulation of the object. The algorithm can also be applied to the design of object reconfiguration with multiple agents including cooperating robots. Only pure rolling and pure sliding relative motions between the fingertips and the object and finger relocation are assumed. The planner allows breaking contact between a fingertip and the object due to finger relocation.

In the thesis the following problems are examined:

- A quasi-static relative motion generation method between the object and the robot, using simulated annealing algorithm.
- Possible finger relocation on the object during the motion.
- Usage of surfaces of the environment to support the motion of the object.
- A near time-optimal trajectory design algorithm, which considers the maximum velocity, acceleration and jerk values of each joint of the robot.

In the thesis a model based motion planner algorithm for manipulating agents for object re-configuration is presented. The motion sequence is represented by a relative velocity matrix. The motion of the agents relative to the object can be pure sliding and pure rolling. The described method uses simulated annealing (SA) for generating the relative motion between the object and agents. The algorithm can be used for example to move a known shaped object to a different position and orientation with a robotic hand.

The thesis describes two extensions for the object reconfiguration method. The first contribution is a method for contact point relocation, which allows the manipulating agent to break contact and later find a new contact point on the object being manipulated. The second extension is the usage of static obstacles in the environment of the manipulating system as pseudo-agents. The two extension can be used together to improve the robustness and flexibility of the planner.

Path planning for a robot is the generation of subsequent robot positions which must be followed during the motion. Beside that, the time properties of the motion should be produced. This is the task of the trajectory planner. Executing the planned spatial trajectories of the motion of the robot and object, the dynamic properties of the robot must be considered. The maximum velocity, acceleration and jerk of each joint of the robot determine additional limits of the motion. The real-time trajectory generation should produce near time-optimal motion considering all of these limits.

After computing a sequence of desired object and robot positions, the distribution of the time parameter along the path is required for the realization of the motion. During the motion these positions are given to the robot controller, which should be able to follow the nominal path if it satisfies the motion constraints.

The thesis proposes a trajectory planning algorithm which takes into account the maximum jerk (the first derivative of acceleration). The goal on this field is the extension of the Minimum-time Spline-based Reduced State space (MSRS) approach of robot trajectory planning. The presented algorithm modifies the motion trajectories generated by the MSRS approach in order to decrease the time required for the robot motion. Modification of the spatial trajectory on the fly or adding new parts can be performed real-time. The algorithm is fast enough to use in real-time applications.

3 Methodology and Appliances of the Research

The object manipulation algorithms presented in this work were tested in simulation. For the work, MATLAB and C++ code were used. For graphical simulation OpenGL was applied. The simulated environment consisted of PUMA 560 arms and the TUB-PC hand.

For the description of the relative motion of two bodies in contact the Montana equations were used [14]. The forces exerted in the contact points are computed with the linear programming tool of MATLAB Optimization Toolbox.

The trajectory planning algorithm was tested in simulation and with a physical system as well. The robot used for test was a Cartesian arm manufactured by Wittmann (Austria). The high-level control system ran under RTOS-32 operation system on an industrial PC.

4 Summary of New Results

Thesis Group 1

I have developed a new object manipulation algorithm for cooperating agents using artificial intelligence based on simulated annealing and A search.*
Publications covering this topic: [1], [2], [3], [4], [8], [9], [11]

1.1. I have proposed a quantized relative velocity matrix V in the local level of the manipulation planner to represent the motion sequence of the manipulating agents. The upper level, the global planner generates the motion of the object using A* search and heuristics. The lower level, the local planner deals with the motion of the agents relative to the object and the design of the contact forces. The relative motion was described with the contact point motion on the surface of the object and the V matrix.

The complete motion of the manipulating agents relative to the object has $n * k$ parameters, where k is the number of agents, n denotes the number of motion phases. Hence the complete motion can be parameterized by a $n * k$ size matrix of the relative velocities denoted by $\mathbf{V} \in \mathrm{R}^{\mathrm{n} \times \mathrm{k}}$. Let the v_{ij} variable be defined as the relative velocity of the i^{th} agent in the j^{th} interval. In order to decrease the search space the relative velocity domain is quantized:

$$v_{ij} \in \{0, v_{x_{max}}, -v_{x_{max}}, v_{y_{max}}, -v_{y_{max}}, \omega_{x_{max}}, -\omega_{x_{max}}, \omega_{y_{max}}, -\omega_{y_{max}}\}. \quad (1)$$

$v_{x_{max}}, \omega_{y_{max}}$ denotes the x direction relative linear velocity and the y direction relative angular velocity. Fig. 1 illustrates the motion of the object and the agents parametrized by the relative velocities. Orientations are denoted by radial lines and forces by arrows.

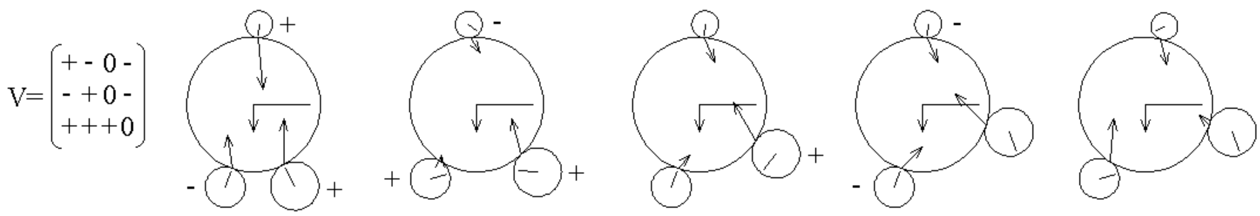

Fig. 1: *The relative motion of the object and agents, rolling clockwise (+) and counter-clockwise (-)*

1.2. I have developed a simulated annealing based local planner to find an appropriate motion sequence which satisfies the motion constraints and provides a quasi-optimal solution. The forces at the contact points are resolved by the solution of a linear program (LP), where friction cones are approximated by pyramids.

I have developed a new energy function for object manipulation design considering the following aspects:

a) Force equilibrium applied on the object.

b) No collision between agents.

c) The absolute sum of contact forces for all agents in all intervals has to be small.

d) Small number of relative velocity changes.

The energy function (objective function, cost function, performance index) can be written as:

$$E(\mathbf{V}) = K_c \sum_{j=1}^{n} c_j + K_e \sum_{j=1}^{n} e_j + K_f \sum_{i=1}^{k} \sum_{j=1}^{n} |f_{ij}| + K_g \sum_{i=1}^{k} g_i \qquad (2)$$

The first sum is the penalty due to collision (the number of collisions multiplied by a K_c constant), the second is the penalty due to no force equilibrium. The third part is the absolute sum of the contact forces, the fourth is the number of velocity changes of the agents. The definitions of the variables and constants are the following:

- E is the energy function to be minimized.
- $\mathbf{V}$ contains the relative velocities.
- K_c, K_e, K_f, K_g are positive weight constants for collision, equilibrium, forces and velocity changes respectively.
- $c_j = 0$ if there is no collision in the j^{th} interval, otherwise 1.
- $e_j = 0$ if there is force equilibrium in the j^{th} interval, otherwise 1.
- f_{ij} is the contact force applied by the i^{th} agent at the j^{th} interval.
- g_i is the number of relative velocity change of the i^{th} agent between the two subgoals.

Let T denote the temperature value for simulated annealing, the desired final temperature is T_{end}. The pseudo-code of the local planner is shown in fig. 2.

1.3. I have shown that the object manipulation algorithm can be applied for cooperating robot manipulators as mobile agents and the combo of a robotic arm and a multifingered hand. In the first case the algorithm was applied to three cooperating 6-DOF robotic arms in 3D. The motion of the end effectors relative to the object can be sequences consisting of pure sliding and pure rolling. Fig. 3 illustrates the motion sequence of the object and the three cooperating robots.

The second application of the algorithm is an object manipulating system for the 6-DOF PUMA 560 arm with a three fingered TUB-PC hand, each finger having three 3 degree of freedom.

```
Get 2 subgoals (start,goal)
Generate random relative velocities(V)
E_old = Compute_energy(V)
WHILE (T > T_end) DO
   Select (i, j) randomly
   Generate a random velocity (v_rnd)
   E_new=Compute_energy(V, v_ij = v_rnd)
   ΔE = E_new − E_old
   IF E_new < E_old THEN
       LET v_ij = v_rnd, E_old = E_new
   ELSE IF e^(−ΔE/T) > Random(1) THEN
       LET v_ij := v_rnd
   END IF
   Decrease(T)
END WHILE
```

Fig. 2: *Local planner pseudo-code*

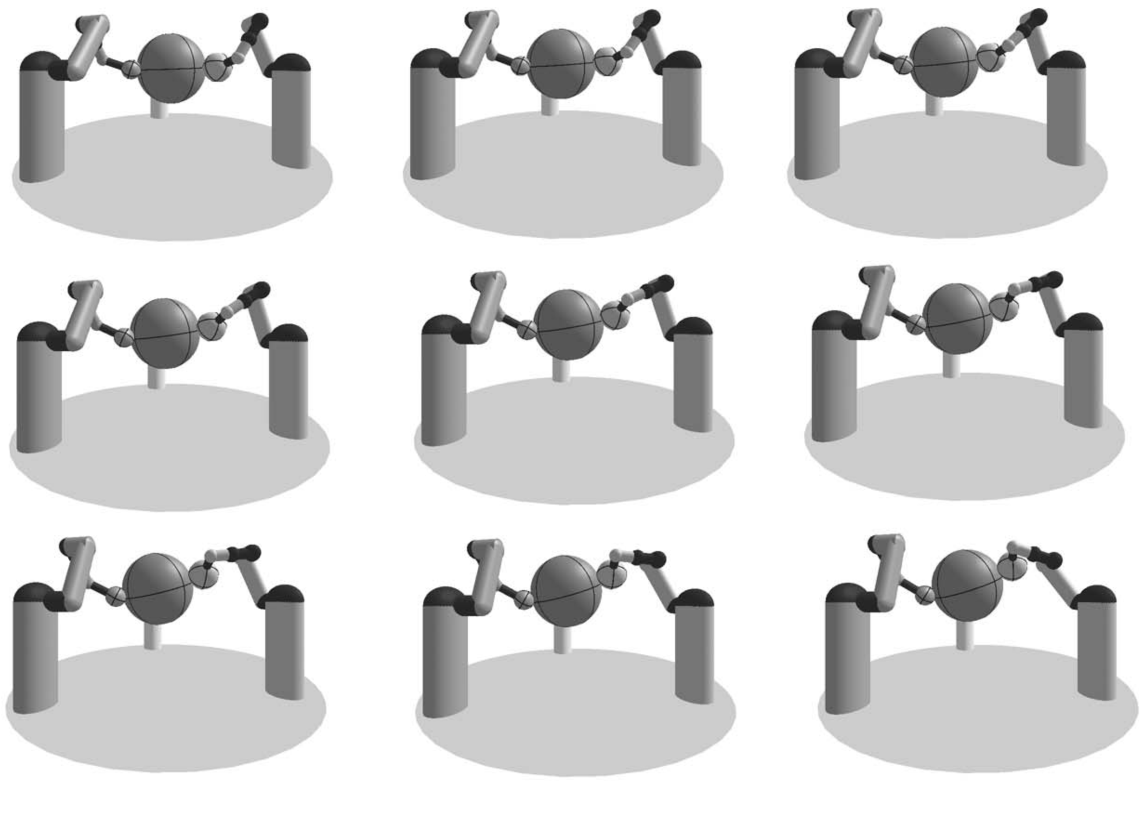

Fig. 3: *Motion sequence with three cooperating manipulators. Top row first, left to right.*

Thesis Group 2

I have proposed further improvements in the manipulation design, including the application of pseudo-agents and a new contact point relocation method for

cooperating agents in order to reconfigure the grasp.
Publications covering this topic: [7], [5], [6]

2.1. Extending the simulated annealing based object manipulation algorithm, I have developed a contact point relocation method for manipulating agents in order to reconfigure the grasp. The benefit of the method is more stable manipulation and more flexibility for the agents. Fig. 4 illustrates the contact point relocation with 3D agents.

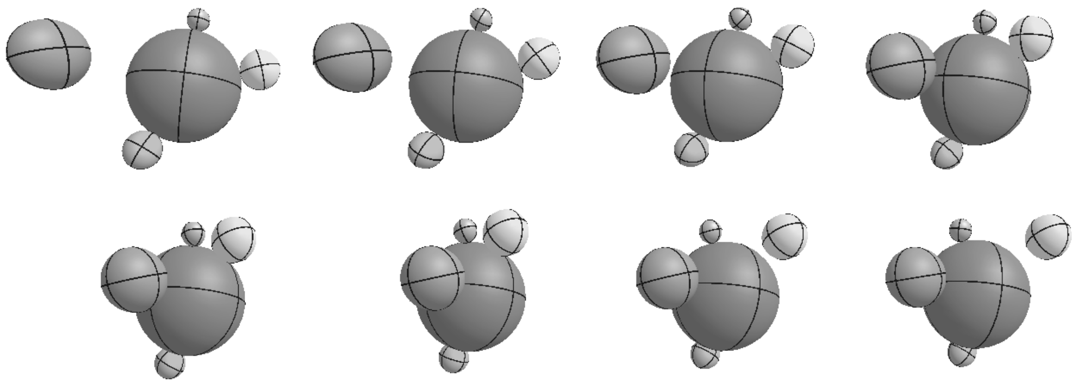

Fig. 4: *Contact point relocation with 3D agents*

2.2. In addition to the object manipulation algorithm, I have introduced the application of static contact surfaces to provide additional contact point. In some cases (heavy objects, etc.) it can be necessary to provide a temporary contact point on the object by a static obstacle of the environment. With the help of this pseudo-agent, the real agents can relocate the current contact points resulting in a better position for the further manipulation. Fig. 5 shows the system flowchart for handling pseudo-agents.

Thesis Group 3

I have generalized the Minimum-time Spline-based Reduced State space (MSRS) approach of trajectory planning, a real-time method for motion planning with jerk (the first derivative of acceleration). The new method optimizes in time robot trajectories of one or more robotic manipulators simultaneously and results in an important improvement in time for 24-hours robotic applications and production systems.
Publication covering this topic: [10]

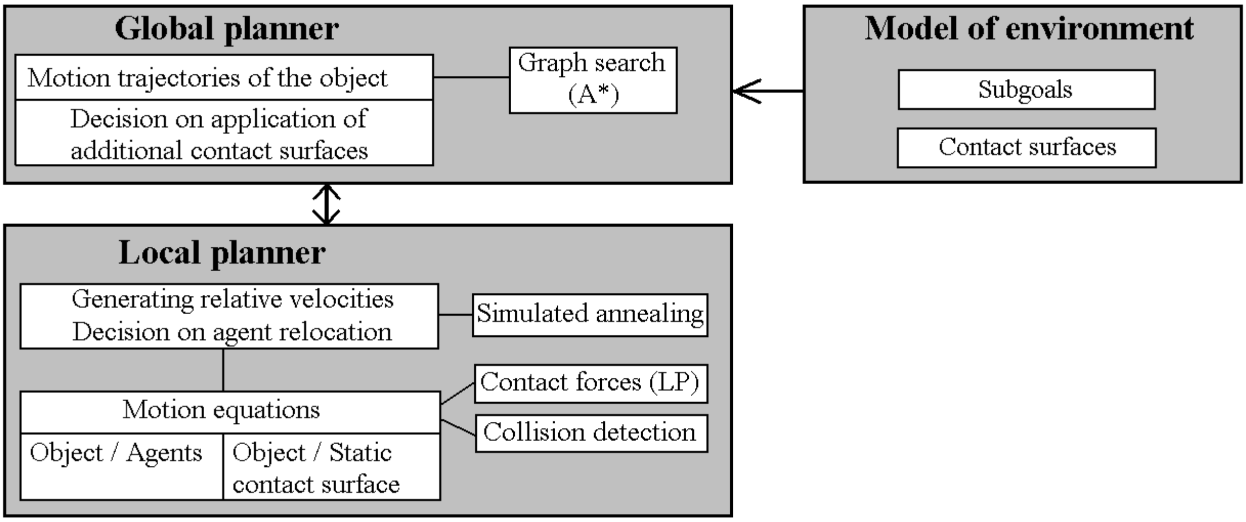

Fig. 5: *System flowchart for handling pseudo-agents*

3.1. I have developed an improved new algorithm which generalizes the original MSRS method introducing new spline types and optimizing the spline list.

MSRS generates a desired reference trajectory considering time-depending constraints for the robot movement, having the geometrical trajectories planned. It represents the motion trajectories in time by splines.

A drawback of the original MSRS approach is the requirement of zero acceleration at the beginning and end of each spline. It is a disadvantage when the velocity along the path is increasing through several subsequent splines.

The improved algorithm of the thesis further optimizes the spline list generated by the MSRS approach. The start and end accelerations of each spline are attempted to be modified. The improved method uses more spline types than the original MSRS method. The acceleration at the end of each spline can differ from zero, thus the velocity at these points is not constant. The velocity along the path can be increased by bigger steps.

The brief description of the algorithm for general 3-D robots is the following:

1. Run the original MSRS algorithm.
2. Search for sequences with monotonously increasing or monotonously decreasing velocity along the path.
3. For each sections, adjust the subsequent splines by gradually changing the acceleration profiles using a trial-and-error process in order to increase path velocity.

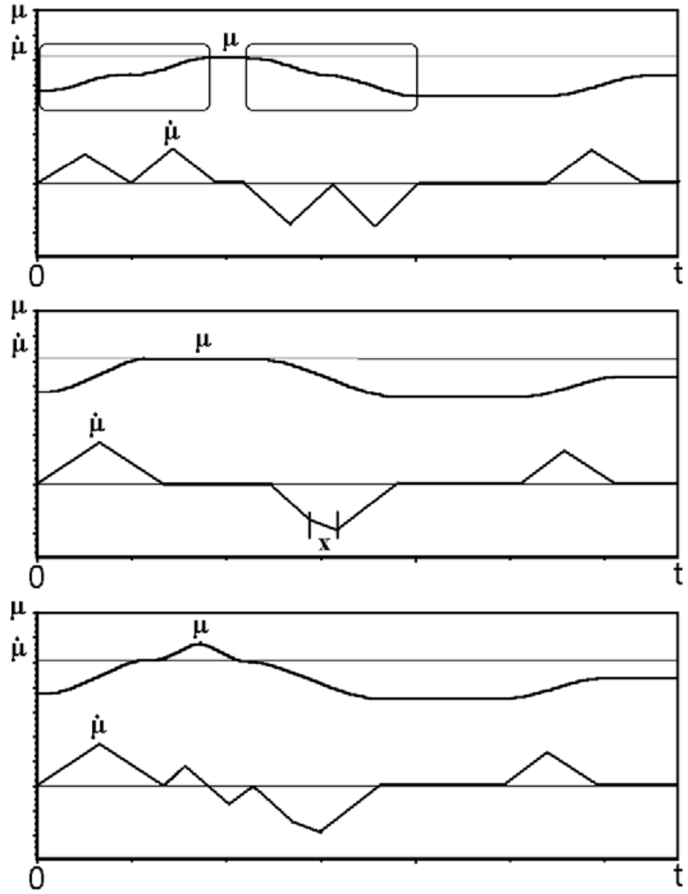

Fig. 6: *Steps of the optimization*

4. Go to the first step if refinement is needed.

Fig. 6 shows the steps of optimization (μ and $\dot{\mu}$ are the velocity and acceleration parameters, respectively).

3.2. The developed method can be applied to optimize in time robot trajectories of one or more robotic manipulators simultaneously. Modification of corner point at the end of the list or adding new ones can be performed real-time.

During the motion all joints are synchronized as a function of one common variable λ. The algorithm optimizes the movement in time by optimizing the characteristics of $\lambda(t)$ using splines to represent the time functions of λ.

Fig. 7 illustrates the velocity (v) and acceleration (a) of the end effector of a Cartesian robot before and after optimization, in the latter case realizing a 6% faster trajectory.

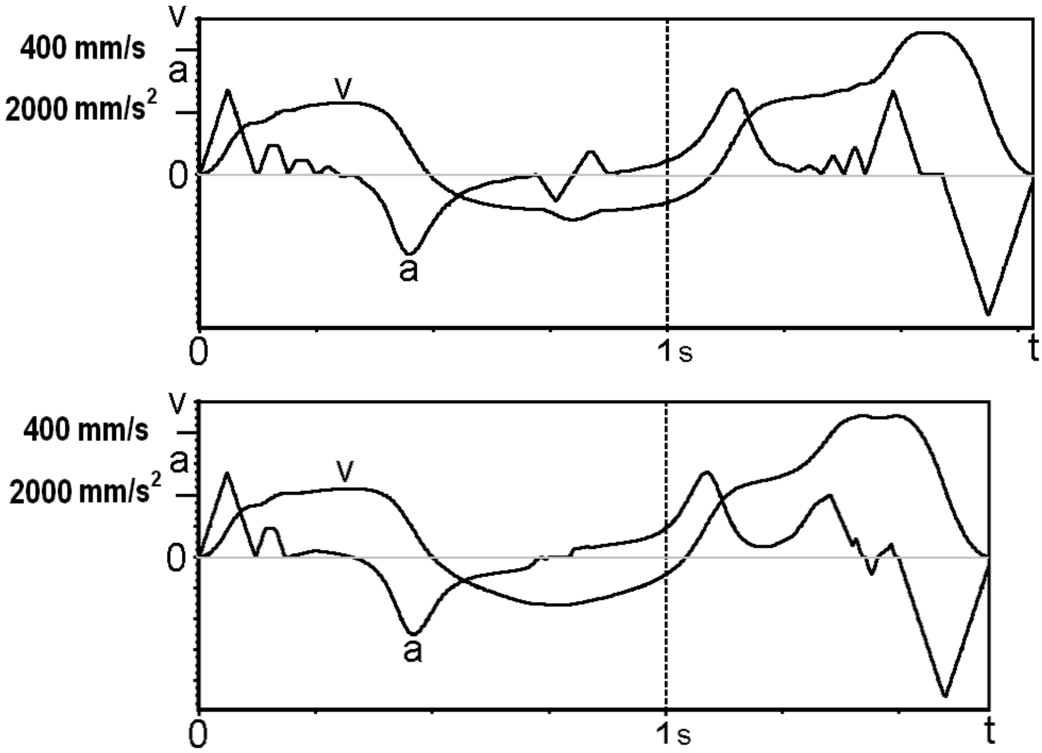

Fig. 7: *Velocity and acceleration of a joint before and after optimization*

5 Application of the Results

Within the frame of the research programs I received the opportunity for a practical application of my theoretical results developed at the Department of Control Engineering and Information Technology at Budapest University of Technology and at the School of Engineering Science at the Simon Fraser University, Canada within national research projects both in Hungary and Canada.

Based on the object manipulation algorithms a software system was developed which can be used for both motion and contact force design for a large number of robotic applications including multifinger dextrous manipulation and cooperating robots. The system provides joint positions and force reference signals for the robot controllers. Byproduct of the software is a graphical interface for the user supporting the high level implementation of the results. The applicability of the algorithm was shown through simulations with 2-D and 3-D setups.

The improved MSRS algorithm was developed and implemented for general robots having analytically solvable inverse geometry under RTOS-32 real-time operation system in 2001/2002. The following data illustrate real-time trajectory planning results for a Cartesian robot. The sampling time of the system was 3 msec. The response time to user commands (such as Emergency Halt) must be so short that the user would not sense considerable delay. For this reason the planning must be carried out within 100 msec. Since there are several tasks running on the same processor under RTOS-32 real-time operation system, the time interval available for the planner module is only a fraction of 100 msec. The running time of the algorithm was between 10 and 40 msec on an Intel Celeron 500 MHz processor depending on the length of the path. Generally, the optimization resulted in an average of 2 to 10% improvement in time which is an important improvement for 24-hours robotic applications and production systems.

The research results were published in journal paper and proceedings of international conferences and were partly included into research programs provided by the Natural Sciences and Engineering Research Council of CANADA (NSERC) under grant No. 611205 and the Hungarian National Research Programs under grant No. FKFP 0417/1997, INTCOM TEMPUS JEP 12555-97, OTKA T 029072 and OTKA T 042634.

Publications on the Topic of the Thesis

[1] G. Vass. Motion design for multi-fingered robotic hands. In *Proceedings of Conference on the Latest Results in Information Technology, Budapest, Hungary*, pp. 117–120, 1998.

[2] G. Vass, S. Payandeh and B. Lantos. On controlled manipulation of objects within multiple dexterous agents. In *Proceedings of the Tenth world congress on the theory of machines and mechanisms, IFToMM, Banff, Canada*, Volume 3, pp. 1103–1108, 1999.

[3] G. Vass, S. Payandeh and B. Lantos. Application of simulated annealing for object manipulation with multiple agents. In *Proceedings of the Second IASTED International Conference, Control and Applications, Oulu, Finnland*, pp. 566–571, 1999.

[4] G. Vass, S. Payandeh and B. Lantos. Toward application of simulated annealing for object manipulation with multiple dexterous agents. In *Proceedings of Tempus, INTCOM '99, Symposium on Intelligent Systems in Control and Measurement, Budapest, Hungary*, pp. 91–96, 1999.

[5] G. Vass, S. Payandeh and B. Lantos. Object reconfiguration with multiple agents using static obstacles. In *Proceedings of Intelligent Autonomous Systems 6, IAS-6, Venice, Italy*, pp. 180–187, 1999.

[6] G. Vass, S. Payandeh and B. Lantos. Object manipulation with dextrous robotic hand using simulated annealing. *Periodica Polytechnica, Serial Electrical Engineering 43*(3), 199–206, 1999.

[7] G. Vass, S. Payandeh and B. Lantos. Object reconfiguration with multiple robotic arms using an additional static contact surface. In *Proceedings of the 6th IFAC Symposium on Robot Control, SYROCO'00, Vienna, Austria*, Volume 2, pp. 669–674, 2000.

[8] G. Vass, S. Payandeh and B. Lantos. A planning system for object reconfiguration with robotic hand and the real time realization of the motion. In *Proceedings of AIRTC, IFAC Symposium on Artifical Intelligence in Real Time Control 2000, Budapest, Hungary*, pp. 295–300, 2000.

[9] G. Vass and E. Tóth. Optimal contact point placement and reconfiguration for dextrous manipulation. In *Proceedings of the 9th Mediterranean Conference on Control and Automation, MED'01, Dubrovnik, Croatia*, pp. 180–187, 2001.

[10] G. Vass, S. Payandeh and B. Lantos. Real-time optimized robot trajectory planning with jerk. In *Proceedings of the 7th Symposium on Robot Control, SYROCO'03, Wroclaw - Poland*, Volume 1, pp. 295–300, 2003.

[11] B. Kiss and G. Vass. Tárgymanipuláció robotkézzel - object manipulation with robotic hand. In *Proceedings of the International Scientific Conference MICROCAD'01, Miskolc, Hungary*, 2001, Vol. J, pp.105-110, ISBN 963 661 457 1.

Other references

[12] M. Cherif and K. K. Gupta. Planning quasi-static motions for reconfiguring objects with a multi-fingered robotic hand. In *Proceedings of IEEE International Conference on Robotics and Automation*, Volume 4, pp. 986–991, 1997.

[13] L. Ludvig. Construction, modeling and intelligent control of robotic hands (in Hungarian). *CSc Thesis, Hungarian Academy of Sciences,* 1997.

[14] D. Montana. The kinematics of contact and grasp. *International Jornal of Robotics Research*, Volume 7, pp. 17–31, 1988.

[15] T. Rieswijk. Robot trajectory planning - a model based geometrically constrained approach. Ph.D Thesis. Technise Universiteit Delft, 1992

[16] J. K. Salisbury and M. T. Mason. Robot Hands and the Mechanics of Manipulation. *MIT Press*, 1985.

[17] R. M. Murray, Z. Li and S. S. Sastry. A Mathematical Introduction to Robotic Manipulation. *CRC Press*, 1994

[18] G. Vass Control system for a dextrous robotic hand (In Hungarian). *M.Sc. Thesis. Process Control Department, Technical University of Budapest*, 1997.