Int j simul model 12 (2013) 1, 27-38
ISSN 1726-4529 Original scientific paper

COORDINATION OF ROBOTS WITH OVERLAPPING WORKSPACES BASED ON MOTION CO-EVOLUTION

Curkovic, P.; Jerbic, B. & Stipancic, T.
Faculty of Mechanical Engineering and Naval Architecture,
Ivana Lucica 5, 10000 Zagreb, Croatia
E-Mail: petar.curkovic@fsb.hr, bojan.jerbic@fsb.hr, tomislav.stipancic@fsb.hr

Abstract
The level of autonomy is the most important feature by which the modern robotic systems development is directed. Furthermore, if the robots are supposed to work together in order to solve a complex task, their workspaces are shared. In this case, the robots present dynamic obstacle to each other. This paper presents a solution of the problem of motion coordination of two robots with overlapping workspaces based on co-evolutionary algorithm for simultaneous motion planning of the two robots. A method for exact calculation of the solution coding chromosome length based on physical limitations of the robots is proposed. The algorithm is evaluated in a simulation environment developed in Matlab. Implementation to the real industrial FANUC Lr Mate 200*i*C robots is performed. The simulation and implementation show high potential in terms of convergence robustness and time.
(Received in March 2012, accepted in August 2012. This paper was with the authors 1 month for 1 revision.)

Key Words: Motion Planning, Industrial Robots, Co-evolutionary Algorithm

1. INTRODUCTION

A possible way to increase manipulability and dexterity of a technical system is to increase the number of the DOF of the system. For example, human hand, considered as paramount for flexibility has 27 DOF in total, with 19 DOF excluding the wrist [1]. The complexity of building similar or adequate technical system increases with the number of the DOF as well as necessary computational power to control them [2-4]. Humans and all higher primates have symmetrically distributed limbs. This can be regarded as an evolutionary achievement or optimization, which allows us to perform complex tasks that are not possible to be completed using only one hand.

In this paper, a robotic system is built of two standard industrial six DOF robots that work with high level of coordination in circumstances where the workspaces of the robots overlap, see Fig. 1. It is important to note that separated controllers, with separated time domains, control the two robots. The motivation for such setup is to increase the performance of the system compared to capabilities two independent six DOF robots each, but at the same times to leave a certain level of autonomy of each robot. This enables the robots to work together, and separately, ignoring the other robot when needed.

The problem with having two robots that can enter each other's workspaces is that they present dynamic obstacle to each other. Standard industrial approach to programming robotic systems where the workspaces overlap is to ensure that the robots actually never enter the area of interference in the same time.

This can be achieved by tracking the position of a *master* robot, and force the *slave* robot to adapt its movement not to interfere with the *master*. It becomes immediately evident that consequences of such an approach are time delays. Moreover it is an unnatural way of motion execution, with lot of pauses, stop and go intermittent motions etc. There are approaches with implementation of simple break-away screws or external shock sensors. The problem is that,

in this scenario, the collision actually occurs, with its physical consequences and a need for replacement is necessary, what introduces production delays and increases costs. Dedicated commercial solutions with special software and hardware installations exist as an option provided by recognized robot manufacturers. These solutions are expensive, and at the same time the collision actually occurs. Despite high monitoring frequencies, production delays are inevitable. Another drawback of similar approaches is that the monitoring is possible in the system only on isolated discrete position, where the special sensor is installed.

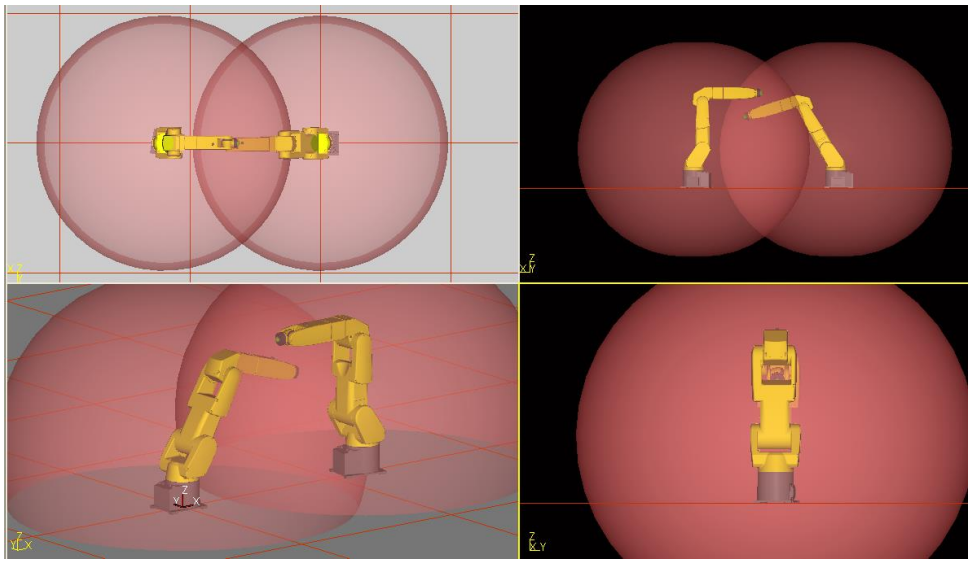

Figure 1: Arrangement of the two robots. Spheres indicate workspaces of the robots.

Many important contributions to the problem of path planning in recent years have been made, each one possessing its own merits and disadvantages. Comprehensive survey can be found in [5]. Practical multi-robot motion planning problems are often decoupled in the sense that the robots trajectories are planned for only one robot at a time in priority order. Then in the second phase, velocities are modulated so that collisions between the robots are avoided [6]. If a problem of path planning is formulated as an optimization problem, robust optimization techniques, such as evolutionary algorithms have proven suitable for finding feasible solutions, since the problem is NP hard. Rana and Zalzala [7, 8] propose an evolutionary planner to evolve near time optimal collision free trajectories for multi-arm robot manipulators. Davidor [9] also applies evolutionary algorithms to the trajectory generation by searching the inverse kinematics solutions to pre-defined end effector robot paths. Pires et al. [10] employ multi objective genetic algorithm to evolve joint-space strings of manipulator configurations. Five indices, namely manipulator joints travelling distance, joint velocity, Cartesian distance, Cartesian velocity and energy are used to qualify the evolving trajectories. Toyoda and Yano [11] used multi-purpose genetic algorithm to optimize movement of multi-joint robotic arms in presence of stationary obstacles. Venegas and Marcial-Romero [12] present preliminary results of Constructive Solid Geometry based approach to path planning of multiple robot arms. They used two phase genetic algorithm to obtain a plan for the robotic arms by using a strategy that combines the exploration of the free collision space while looking for the target position from each previously explored area. Abo-Hammour et al. [13] developed continuous genetic algorithm and implemented it to solve the problem of path planning of Cartesian manipulator in the workspace containing stationary obstacles. Majority of papers dealing with evolutionary-based path planning consider single agent operating in an environment without obstacles, or in an environment containing static, usually simplified obstacles [14, 15]. The problem then boils down to finding suitable set of interior points, to be

interpolated to formulate the polynomial of given order representing the trajectory. Another common drawback of methods based on evolutionary algorithms is the time required to find a feasible solution, thus very limited number of implementations to real robots is presented.

An approach to a path planning of a dual-arm reconfigurable robot is presented in [16]. The problem deals with a SCARA-like configuration of a robot controlled by a single controller. Real-time optimum is achieved for given start and end positions of the tool centre point (TCP). The algorithms were successfully applied to an industrial application of gear assembly, with some limitations in terms of reliability.

There are a number of research approaches to compliance control of a coupled dual arm robotic systems. Such systems are based on two robots which have a contact between reference points, either by caring a common object or directly. These methods are based on impedance control, to maintain interaction forces within some predefined level [17, 18]. A probabilistic approach to path planning based on dense tree exploring is presented in [18]. The bottleneck of proposed approach was high-dimensionality of configuration spaces for exact path verification, thus limiting the performance of proposed method.

Co-evolutionary algorithms offer great potential for concurrent multiagent domains [20, 21]. Concepts of co-evolution were also successfully coupled with other heuristic optimization techniques [22, 23] to solve complex optimization problems whose search-spaces are connected. The main problem reported concerning the co-evolutionary algorithms is their game-theoretic background and resulting pathologies, namely cyclic dynamics, loss of fitness gradient and evolutionary forgetting [24].

The main contribution of this paper is the development of modified co-evolutionary algorithm, and its implementation to the real system composed of two industrial robots. We modify the co-evolution to approach the real-time usability of the system, developing methods of selection of best collaborators based on combination of Pareto front exploration and hall of fame creation to save best collaborators – instances of both populations whose combinations yield best results. The results are evaluated in genotype space as motion of the two robots, with following evaluation criteria: number of collisions between robot links or stationary obstacles, trajectory length, TCP velocity profile, and the total angle of rotation in robot joints. Eight instances of co-evolution were implemented and statistically evaluated. The results show that modified co-evolutionary algorithm outperforms the standard instance of the algorithm in terms of convergence speed what is a critical parameter if the algorithms are to be implemented in real-time.

The remainder of the paper is organized as follows: In section 2, main features of system architecture are described. Section 3 presents simulation environment developed for calculation, together with main steps of the proposed algorithm. In section 4 an implementation to the real robots for experimental verification is described. Section 5 contains final discussion and conclusions with future work insights.

2. FEATURES OF THE PROPOSED SYSTEM

The system used in our work is built from the following components: two Fanuc Lr Mate 200 *i*C robots, equipped with force / torque sensors, cameras, Karel programming language with fast Ethernet cards etc., the setup is presented in Fig. 2.

The robots are connected via fast Ethernet to each other and to a PC, running Matlab R2008. The Matlab is extended with the MuPAD symbolic engine, optimized for operating on symbolic mathematical expressions and custom graphics.

The calculation begins after the robots have successfully received start and end configurations; main steps of the algorithm are illustrated in Fig. 3. These data is provided from the robot controller as a query message to the PC. The initial and final configurations

can be given via the teach pendant or can be obtained from the vision system after the object of interest has been recognized.

The algorithm checks whether initial and final configurations cause collision of arbitrary part of the robot 1 and robot 2. If there is collision present, the planning cannot be executed and the error message is provided indicating that different set of initial conditions has to be provided.

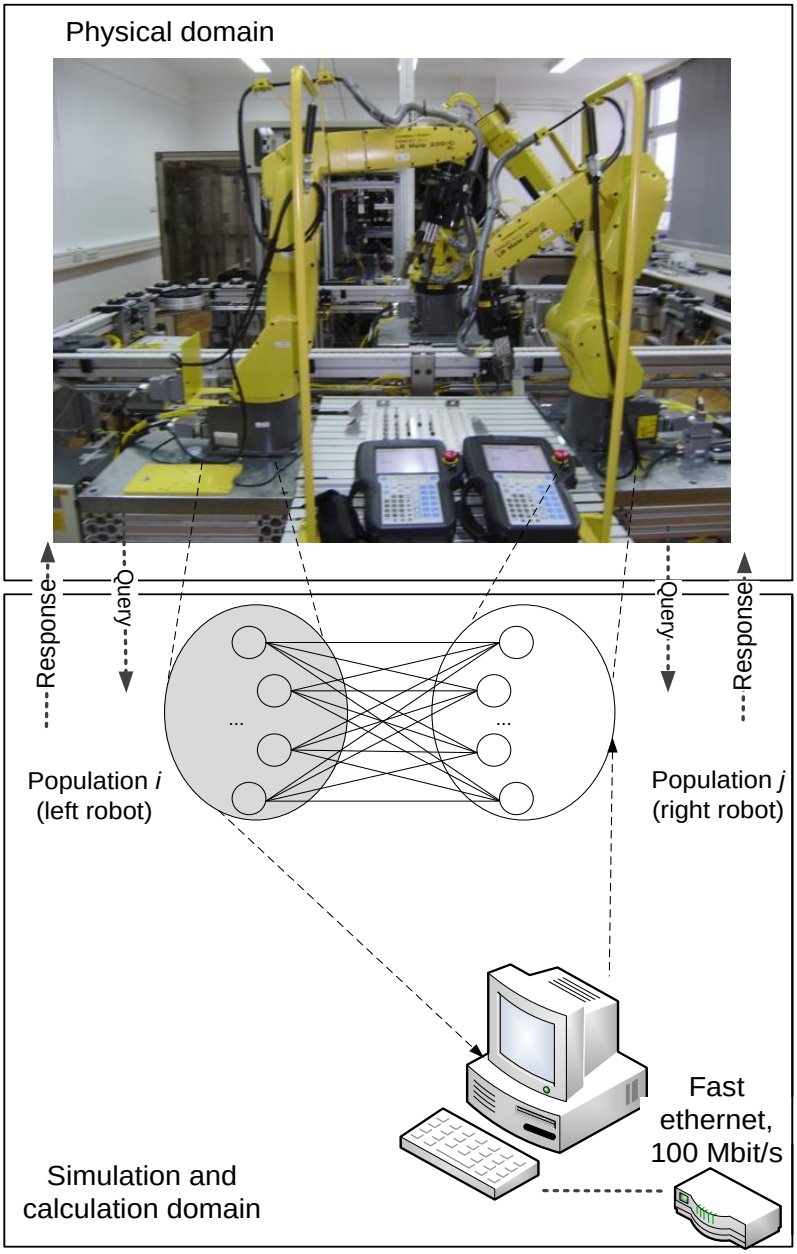

Figure 2: System architecture.

The planning is processed in the *ZX* of the robots, reducing the search space significantly. Two evolutionary algorithms that are coupled by a common fitness function perform planning. The main criterion, which connects the two populations, is criterion of collision. Optimal value of this criterion is known in advance, and for the solution to be feasible and acceptable it should be equal to zero.

The search starts with random initialization of two populations, each containing a set of codified configurations for one robot in a finite number of time steps. The performance of

each instance of the populations depends on the current state of the robots (encoded as chromosomes) from other population, since the two must coordinate the motion, to achieve continuous collision free movement.

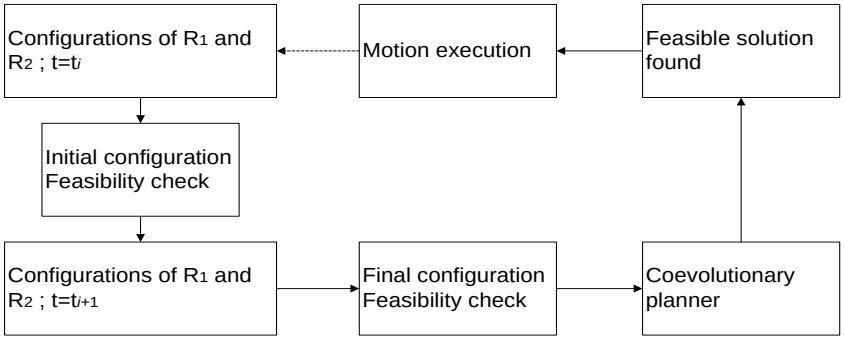

Figure 3: Main steps of the proposed algorithm.

In the canonical co-evolutionary algorithm, all individuals from both populations should be mutually evaluated – this is called complete mixing. This way maximum amount of information is extracted from the current solution pool. It is obvious that this procedure is extremely time consuming and not applicable to the problems where convergence time is a critical parameter. To speed-up the evolution process, modified co-evolution is introduced. In the modified version, each individual is evaluated by a finite set of the top collaborators from the other population, based on scores from previous generation; this is called *hall of fame*. Pareto-front based selection of the populations is combined with hall of fame individuals, which is a modification of the approach proposed by the Sims [25], where "…the most "interesting" results occurred when the all vs. best competition pattern was used". The successive generations are reproduced on the basis of roulette wheel, to fasten the evaluation process. Standard single point crossover operator is used. Individual in a population is represented as chromosome containing real-valued vector in the joint space (1):

$$\left[\left\{q_{11}^{(\Delta t,G)},..,q_{ij}^{(\Delta t,G)}\right\},\left\{q_{11}^{(2\Delta t,G)},..,q_{ij}^{(2\Delta t,G)}\right\},..,\left\{q_{11}^{((n-2)\Delta t,G)},..,q_{ij}^{((n-2)\Delta t,G)}\right\}\right] \quad (1)$$

where i denotes the robot i = 1, 2, j is the number of DOF, since the problem is reduced to *planar* plane, j = 1, but generally it is expandable to the full six DOF space, Δt is sampling time between two consecutive configurations, q is angle between the link and positive x axis, G is current generation.

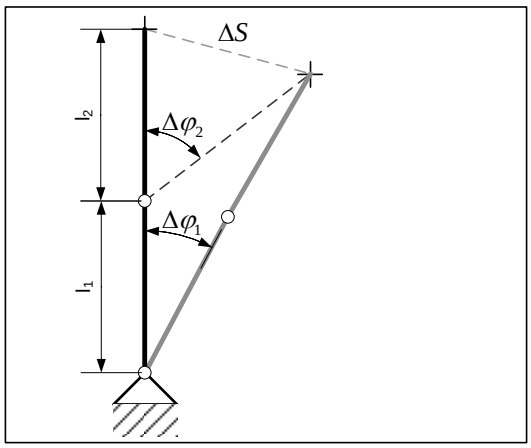

Figure 4: Analysis of the movement of the robots *TCP*.

At the beginning of the evolutionary process, joint values are randomly initialized, but the initial and final configurations are not encoded into the string because they remain constant throughout the search process. Without the loss of generality, adopted is normalized sampling time with $\Delta T = 0.01$ s.

Adopting the form of an individual in the population according to (1), what is a discrete form of individual and results with discrete information given to the robot controller, there is a trade-off between the granularity of the discretization and the real time computational requirements. There is an important question – how to decide what is the appropriate length of the chromosome representing the individual and defining the granularity of discretization?

It is assumed for this calculation that the robot moves as follows, see Fig. 4: if the second joint of the robot is static, the joint displacement of the first joint is, $\Delta\varphi_1$, if the first joint of the robot is static, the joint displacement of the second joint is $\Delta\varphi_2$. In both cases the distance travelled by the reference point equals ΔS. Now the formula can be derived: $\Delta\varphi_1 \cdot (l_1 + l_2) = \Delta\varphi_2 \cdot l_2 = \Delta S$ where l_1 and l_2 are for the lengths of the robot links. If T is assumed as sampling time, the distance travelled by the reference point is $\Delta S = V_{max} \cdot T$, and the angles travelled can be obtained from (2):

$$\Delta\varphi_{1,max} = \frac{V_{max} \cdot T}{l_1 + l_2}; \ \Delta\varphi_{2,max} = \frac{V_{max} \cdot T}{l_2} \ , \tag{2}$$

where V_{max} is for maximum velocity of the *TCP*. For this calculation we will assume that $V_{max} = 5$ m/s, $T = 0.01$ s and $l_1 = l_2 = 0.25$ m. The angles calculated are: $\Delta\varphi_1 \leq 11.5°$, $\Delta\varphi_2 \leq 23°$, thus the discretization space is bounded. The calculation of the chromosome length L follows:

$$L \geq 2 \cdot \frac{\Delta\overline{\varphi}_i^G}{\Delta\varphi_{1,\max}}, \ \ l \in \mathbb{N} \ \wedge \ l \neq 0 \tag{3}$$

where $\Delta\overline{\varphi}_i^G = \left|\varphi_i^{(n-2,G)} - \varphi_i^{(1,G)}\right|$ is the difference between the initial and final angle of rotation of the ith robot link, L is the length of the chromosome rounded up to the first larger integer. Number 2 is for the couples of values in the chromosome, two angles for one configuration. For the worst case, when robots initial and final configuration are on the boundaries of the upper semiplane, what will actually never happen in reality, the resulting chromosome length is $L = 32$, what means 16 configurations are needed at upper bound.

Since the evaluation of populations composed of the chromosomes of length 32 is extremely computationally complex and is thus not possible to perform it in real time, we keep the length $L < 10$, what is experimentally proven to be achievable in real time, calculating less intermediate points. A simple interpolation procedure is developed to reconstruct the missing configurations based on polynomial of at least third order. In that case, the solutions are obtainable in real time.

The problem analysed in this paper presents a multi objective optimization problem with conflicting criteria. All the criteria are to be minimised, but they are conflicting in nature.

A concept of Pareto [26-28] dominance is implemented into the evolutionary algorithm to keep a set of good solutions, which are good approximates of the Pareto front. It is obvious, that the proposed algorithm can found solutions on the Pareto front in an early phase of evolution, after only 30 generations, see Fig. 5. This is beneficial property of the algorithm, whose consequence is increased likelihood of finding global – optimal area of the multi-objective search space.

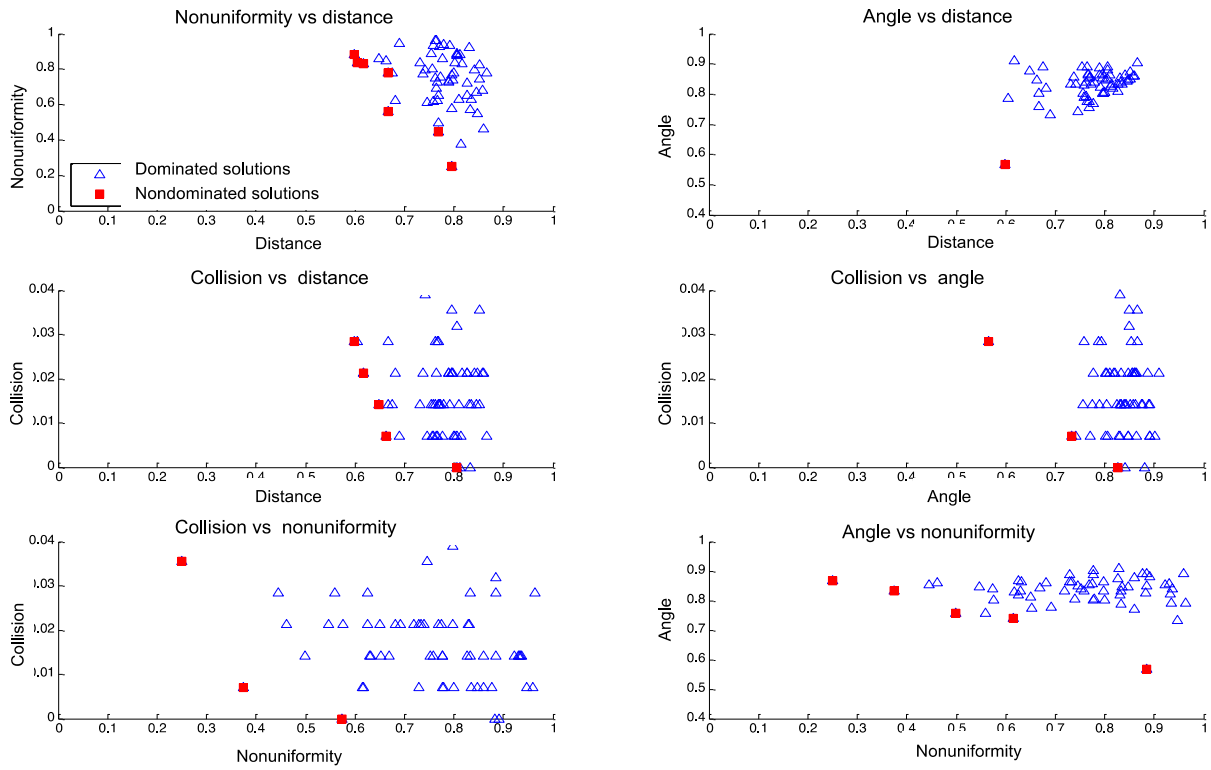

Figure 5: Pareto fronts in early phase (G = 30) of evolution for different pairs of optimization criteria.

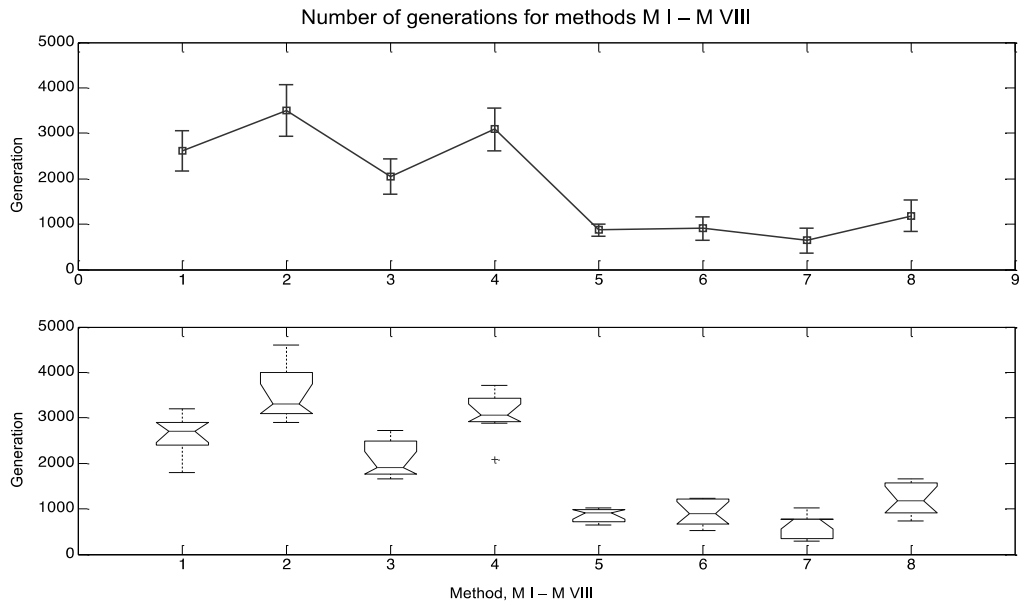

Figure 6: Impact of the selection method to the speed of the convergence.

The time required to find a feasible solution is accepted if it is in range less than 5 s. It is possible, however, since the algorithm is heuristic that no feasible solution is found in the predefined time range. The good thing is that it is easy to check whether the algorithm has converged, based on critical criterion of collision. If no convergence has occurred, and collision is present, the algorithm generates new populations and starts again. Since the planning takes place only once, at the beginning of the motion, a pause, 5 s max. is noticeable during the calculation, but the results proven acceptable for the real time application. Often, the time is significantly less than maximal acceptable time. Actually, every motion solution is

acceptable if the resulting number of collisions equals to zero. This solution might not be optimal in terms of all optimization criteria, but if the time is critical, it can be adopted and executed.

Regarding the number of generations needed for the convergence, we have compared several variations of standard approach based on roulette wheel selection, methods I-IV in Fig. 6. To increase the speed of the convergence, we introduced Pareto based selection. The idea is to keep not only best collaborators, as pairs of values from both populations in the hall of fame, but to include the whole Pareto front of the current generation for evaluation. The methods based on Pareto domination are labelled methods V-VIII in Fig. 6. It is visible that significant impact on the speed of the convergence is achieved introducing the new selection method.

The differences between methods are following: I – IV are instances of standard roulette wheel selection process, with different values of evolutionary parameters, such as size of the populations, mutation and crossover probabilities. The methods V – VIII are based on Pareto selection, with different parameters of population sizes, crossover and mutation operator values. The methods are evaluated with 20 runs for each method, and the corresponding results are illustrated with standard deviation, top of Fig. 6, and box-plot, at the bottom of Fig. 6. We experimented with the *depth* of the hall of fame procedure. Namely, if the hall of fame is not changed during the process of evolution, an opposite effect in terms of convergence speed can occur since the number of good individuals tends to grow. The consequence is that more calculations per generation are necessary. In an early phase of evolution this is not important since the hall of fame is empty, or contains a small number of solutions. The parameter *depth* of the hall of fame indicates the memory effect, namely after how many generations will new members start to replace the old ones. For our purposes, we obtained best results if a hard limit of 15 % of the population size was set to limit the size of hall of fame. Pareto front is changing from generation to generation. We obtained best results keeping the complete front for evaluation. It is important to note that the size of the front changes during the evolution since global optimum for some criteria might be approached or found.

3. SIMULATION RESULTS

Simulation environment on the computational level is developed and implemented in Matlab. The links of the robots are presented as lines. A finite set of reference configurations is evolved because of time constraints. Full path reconstruction with arbitrary time step follows, as shown in Figs. 7 and 8.

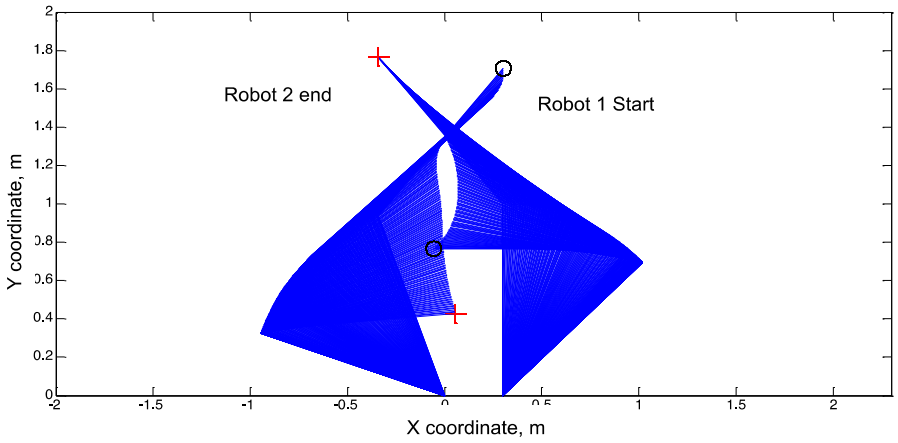

Figure 7: Motion of the two robots in developed simulation environment.

Important is to monitor the change of rotation angle of the joints in time, since this value has an upper bound in terms of achievable accelerations of the real robot. Peaks of the gradient should be avoided, for the smooth motion of the robot, and for the motion to be applicable to the real system.

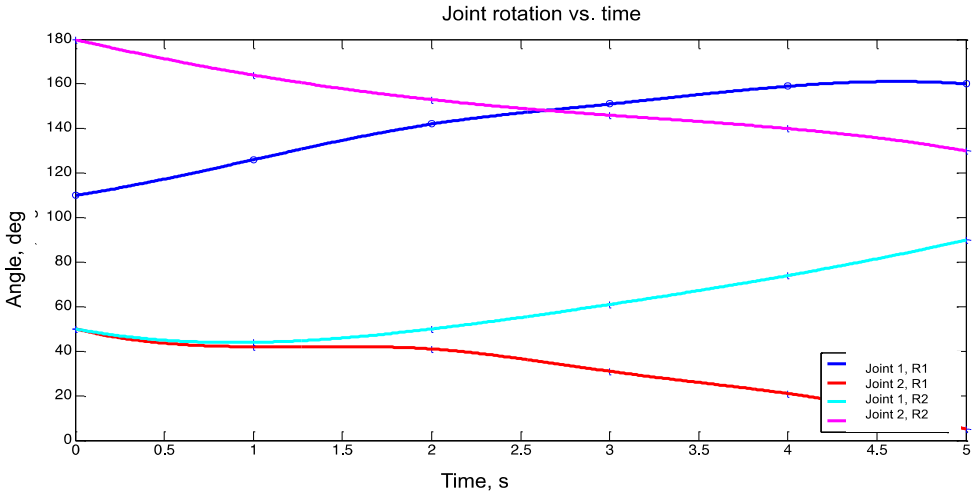

Figure 8: Joint rotations for the two robots.

After the solution is found, it is possible to implement it on the real robots. For safety reasons, an intermediate step is introduced – implementation of the robot movement in Fanuc's physical simulation environment Roboguide. An illustration of the robot motion is shown in Fig. 9.

A special case is illustrated, with left robot being stationary. This is a simpler variant of general case with both robots moving. The algorithm successfully converged and found a set of transition configurations to ensure smooth, collision free motion.

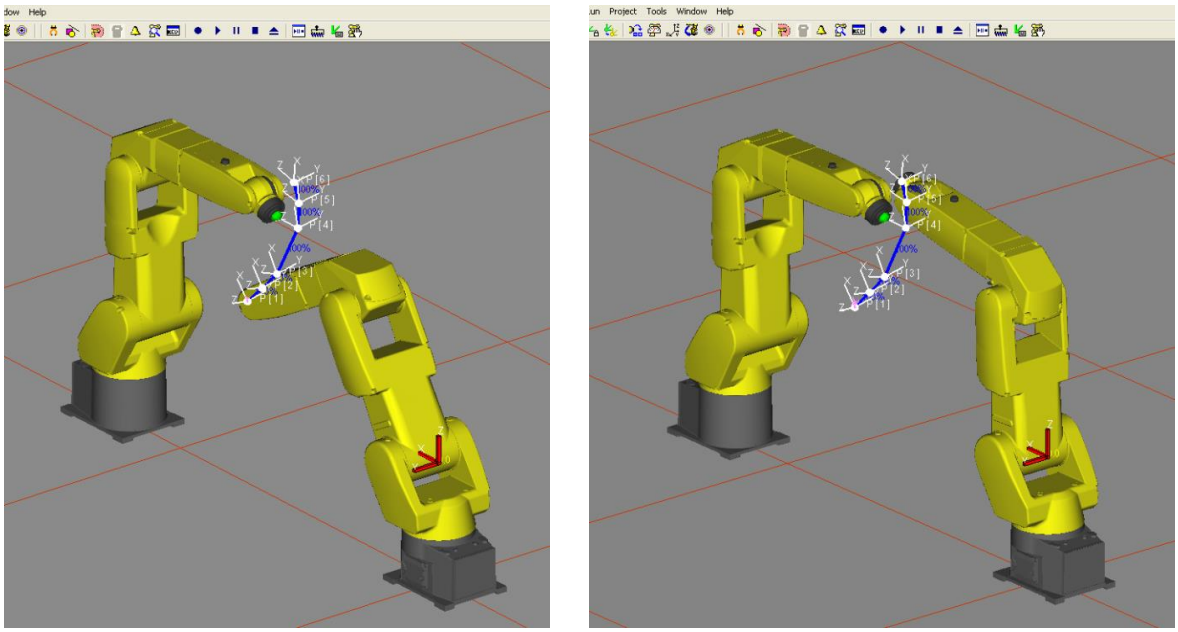

Figure 9: Motion execution in Roboguide physical simulator.

A problem was noticed during the physical simulation in sense that the TCP used as a reference point for the Matlab calculation is surrounded by the flange of the robot and located in the centre of the sixth DOF link of the robot.

Actually, that means that there is a certain probability of collision if, for example, solution presented by Fig. 7 would be implemented. Applying a variable safety factor around the TCP, virtually shifting the trajectory in every point towards the global coordinate system origin, the problem was solved. Since the factor applied is variable, mounting different tools to the robots flange is possible, without the risk of collision.

The values of the safety factor are experimentally determined according to the physical characteristics of the robot and the tool mounted on the flange.

4. APPLICATION TO THE REAL ROBOTIC SYSTEM

The application to the real robots was, after tuning of safety factor discussed in previous section, a task related to solving a set of communication / synchronization problems between the robots, PC, and a programmable logic controller. Procedures were developed to synchronize Matlab execution with the robots through robots native programming language Karel.

To enable TCP IP communication protocols in the system, we assigned following IP addresses: Robot 1 (left robot): 192.168.123.26; Robot 2: 192.168.123.25, and PC: 192.168.123.40. Two communication channels are opened. The first channel is for sending data from robot to the PC. In that case, the robot has a role of the server, while PC is a client receiving the desired data. The second channel is for sending of the response from PC to the robot controller. Now the PC has a role of the server and robot controller is a client receiving the desired data. The messages that are sent are codified and can have different forms, indicate whether the motion is possible or not (i.e. intersection of the robot links in initial or final configurations). If the motion is possible, a list of coordinates is sent.

It is important to ensure that the time delay between the motion start of the both robots is kept to minimum. This is done through directly connected I/O signals between the two robot controllers. These signals have the role to trigger the motion in desired moment on both robots.

5. CONCLUSIONS

In industrial environments where several autonomous robots work in shared environment, it is essential to find feasible, collision free motions. In this paper, we have presented a method based on co-evolutionary algorithms that finds optimal or near-optimal trajectories for two six DOF robots. Exact values for the granularity of calculation, through chromosome length, are calculated, based on the physical limitations of the robotic system used for experimental verification. This approach is scalable, in terms of adapting it to different dimensions of robots. Also, it is general, implying the possibility of adding new DOFs or new robots in the system.

Standard approaches for co-evolutionary algorithms are implemented and evaluated, and modifications including Pareto based selection and collaborations transition to the new generations are presented. The methods are compared and significant speed-up in convergence is achieved.

The routine developed in this research is expected to serve as a low-level procedure ensuring safe motion of the robots requested to perform high-level tasks. To solve the problem, we have developed simulation environment that analyses movement of the simplified robots models. The next step is implementation in the physical simulation environment, where interaction between physical constraints and simplified model occurs and tuning of the model can be performed to satisfy real-world constraints.

Finally, full implementation is performed on two Fanuc Lr Mate 200*i*C robots. We see further work in adding more robots and more DOFs into the model. The model is developed to be scalable and allows inclusion of additional features, although important questions regarding computational power are expected. The main bottleneck of the approach remains its computational complexity, despite effort put in algorithmic simplification. A possible way to overcome this problem is to implement co-evolutionary process to the distributed computer architecture and to perform independent evolutionary operators in parallel.

At this point of research, the planning is performed only once, at the beginning of the motion of the two robots. This is fine if nothing unexpected occurs during the motion of the two robots. If one robot would unexpectedly stop during the motion, there is a possibility of collision with other robot, since the robots are not aware of each other after the motion starts. This is a limitation, but future work will consider permanent motion sampling for the two robots. If an interaction of motion of one of the robots is detected via a reference signal value, this robot could be treated as a stationary obstacle. Thus a simple replanning procedure would be executed eliminating the possibility of collision during the operation cycle.

ACKNOWLEDGEMENTS

Authors would like to acknowledge support of Croatian Ministry of Science, Education and Sports, through projects no. 120-1201948-1941, Autonomous Multiagent Automated Assembly, and joint project *IGRAMO* conducted with Kungliga Tekniska högskolan *KTH*, Stockholm.

REFERENCES

[1] Agur, A. M. R.; Lee, M. J.; Grant, J. C. B. (1999). *Grant's Atlas of Anatomy*, Lippincott Williams and Wilkins, 10th ed., Ambler

[2] Poppeova, V.; Urícek, J.; Bulej, V.; Sindler, P. (2011). Delta robots – robots for high speed manipulation, *Technical Gazette*, Vol. 18, No. 3, 435-445

[3] Skorc, G.; Cas, J.; Brezovnik, S.; Safaric, R. (2011). Position Control with Parameter Adaptation for a Nano- Robotic Cell, *Strojniski vestnik - Journal of Mechanical Engineering*, Vol. 57, No. 4, 313-322, doi:10.5545/sv-jme.2009.017

[4] Meneses, J.; Castejon, C.; Corral, E.; Rubio, H.; Garcia-Prada, J. C. (2011). Kinematics and Dynamics of the Quasi-Passive Biped "PASIBOT", *Strojniski vestnik - Journal of Mechanical Engineering*, Vol. 57, No. 12, 879-887, doi:10.5545/sv-jme.2010.210

[5] Latombe, J. C. (1991). *Robot motion planning*, Kluwer Academic Publishers, Boston doi:10.1007/978-1-4615-4022-9

[6] Saha, M.; Isto, P. (2006). Multi-Robot Motion Planning by Incremental Coordination, *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference*, doi:10.1109/IROS.2006.282536

[7] Rana, A. S.; Zalzala, M. S. (1995). An Evolutionary Algorithm for Collision Free Motion Planning of Multi-arm Robots, *IEEE Genetic Algorithms in Engineering Systems: Innovations and Applications*

[8] Rana, A. S.; Zalzala, M. S. (1996). An Evolutionary Planner for Near Time-Optimal Collision-Free Motion of Multi-Arm Robotic Manipulators, *UKACC International Conference on Control*, doi:10.1049/cp:19951036

[9] Davidor, Y. (1991). *Genetic Algorithm and Robotics; A Heuristic Strategy for Optimization*. World Scientific, Singapore

[10] Solteiro Pires, E. J.; Tenreiro Machado, J. A.; Moura Oliveira, P. B. (2004). Robot Trajectory Planning Using Multi-objective Genetic Algorithm Optimization, *Proceedings of the GECCO*, LNCS, Springer, Heidelberg, doi:10.1007/978-3-540-24854-5_64

[11] Toyoda, Y.; Yano, F. (2004). Optimizing Movement of A Multi-Joint Robot Arm with Existence of Obstacles Using Multi-Purpose Genetic Algorithm, *Industrial Engineering and Management Systems*, Vol. 3, No. 1, 78-84

[12] Venegas Montes, H. A.; Raymundo Marcial-Romero, J. (2009). An Evolutionary Path Planner for Multiple Robot Arms, *Evo Workshops*, LNCS, Springer, Heidelberg, doi:10.1007/978-3-642-01129-0_39

[13] Abo-Hammour, Z. S.; Alsmadi, O. M. K.; Bataineh, S. I.; Al-Omari, M. A.; Affach, N. (2011). Continuous Genetic Algorithms for Collision-Free Cartesian Path Planning of Robot Manipulators, *International Journal of Advanced Robotic Systems*, Vol. 8, No. 6, 14-36, doi:10.5772/50902

[14] Curkovic, P.; Jerbic, B.; Stipancic, T. (2008). Hybridization of adaptive genetic algorithm and ART 1 neural architecture for efficient path planning of a mobile robot, *Transactions of FAMENA*, Vol. 32, No. 2, 11-21

[15] Curkovic, P.; Jerbic, B. (2007). Honey-bees optimization algorithm applied to path planning problem, *International Journal of Simulation Modelling*, Vol. 6, No 3, 154-165, doi:10.2507/IJSIMM06(3)2.087

[16] Fey, Y.; Fuqiang, D.; Xifang, Z. (2004). Collision-free motion planning of dual-arm reconfigurable robots, *Robotics and Computer-Integrated Manufacturing*, Vol. 20, No.4, 351-357, doi:10.1016/j.rcim.2004.01.002

[17] Surdilovic, D.; Yakut, Y.; Nguyen, T-M.; Pham, X. B.; Vick, A.; Martin, M. (2010). Compliance Control with Dual-Arm Humanoid Robots: Design, Planning and Programming, *IEEE-RAS International conference on Humanoid Robots*, Nashville, TN, USA, doi:10.1109/ICHR.2010.5686273

[18] Caccavale, F.; Natale, C.; Siciliano, B.; Villani, L. (2001). Achieving a cooperative behaviour in a dual-arm robot system via a modular control structure, *Journal of Robotic Systems*, Vol. 18, No. 12, 691-699, doi:10.1002/rob.8107

[19] Vahrenkamp, N.; Kaiser, P.; Asfour, T; Dillmann, R. (2011). RDT+: A Parameter-free Algorithm for Exact Motion Planning, *IEEE International Conference on Robotics and Automation (ICRA*), 482-488, doi:10.1109/ICRA.2011.5979777

[20] Panait, L.; Luke, S.; Wiegand, R. P. (2006). Biasing Coevolutionary Search for Optimal Multiagent Behaviors, *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 6, 629-645

[21] Curkovic, P.; Jerbic, B. (2010). Dual-Arm Robot Motion Planning based on Cooperative Coevolution, Camarinha-Matos, L. M.; Pereira, P.; Ribeiro, L. (Eds.), *Emerging trends in technological innovation*, Springer Verlag, Heidelberg, 169-179, doi:10.1007/978-3-642-11628-5_18

[22] Sinha, A. K.; Zhang, W. J.; Tiwari, M. K. (2012). Co-evolutionary immuno-particle swarm optimization with penetrated hyper-mutation for distributed inventory replenishment, *Engineering Applications of Artificial Intelligence*, Vol. 25, No. 8, 1628-1643, doi:10.1016/j.engappai.2012.01.015

[23] Kou, X.; Liu, S., Zhang, J.; Zheng, W. (2009) Co-evolutionary particle swarm optimization to solve constrained optimization problems, *Computers & Mathematics with Applications*, Vol. 57, No. 11-12, 1776-1784, doi:10.1016/j.camwa.2008.10.013

[24] Zhao, Q.; Hammami, O.; Kuroda, K.; Saito, K. (2000). Cooperative co-evolutionary algorithm – how to evaluate a module?, *IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, 150-157, doi:10.1109/ECNN.2000.886230

[25] Sims, K. (1994). Evolving 3D Morphology and Behavior by Competition. *Artificial Life IV Proceedings*, MIT Press, doi:10.1162/artl.1994.1.353

[26] Castillo, O.; Trujillo, L. (2006). Multiple objective optimization genetic algorithms for path planning in autonomous mobile robots, *Soft Computing*, Vol. 11, No. 3, doi:10.1007/s00500-006-0068-4

[27] Censor, Y. (1977). Pareto Optimality in Multiobjective Problems. *Applied Mathematical Optimization*, Vol. 4, No. 1, 41-59, doi:10.1007/BF01442131

[28] Kalyanmoy, D. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester, England