

Multi-agent System for Off-line Coordinated Motion Planning of Multiple Industrial Robots

Shital S. Chiddarwar¹ and N. Ramesh Babu²

¹ Assistant Professor, Department of Mechanical Engineering, Shri Ramdeobaba Kamla Nehru Engineering College, Nagpur, Maharashtra, India

² Professor, Manufacturing Engineering Section, Department of Mechanical Engineering, Indian Institute of Technology Madras, India

Abstract This article presents an agent based framework for coordinated motion planning of multiple robots. The emerging paradigm of agent based systems is implemented to address various issues related to safe and fast task execution when multiple robots share a common workspace. In the proposed agent based framework, each issue vital for coordinated motion planning of multiple robots and every robot participating in coordinated task is considered as an agent. The identified agents are interfaced with each other in order to incorporate the desired flexibility in the developed framework. This framework gives a complete strategy for determination of optimal trajectories of robots working in coordination with due consideration to their kinematic, dynamic and payload constraint. The complete architecture of the proposed framework and the detailed discussion on various modules are covered in this paper.

Keywords Agent based systems; Coordination; Multiple robot motion planning; Optimal Trajectory Planning; Conflict resolution; Inverse Kinematics

1. Introduction

The promising epitome of sustainable manufacturing essentially demands for optimal utilization of expensive and seldom used resources in modern manufacturing systems. The desired sustainability can be efficiently incorporated in reconfigurable and flexible

manufacturing systems using multi-robot systems. Multi-robot systems have inherent advantages like efficient sharing of resources, reduction in cycle time and optimal energy consumption. These aspects are of prime importance from sustainable manufacturing point of view where emphasis is on optimal use of resources and conservation of natural resources. In order to bring out these advantages associated with multi-robot systems, there is a need for a flexible and generic framework for motion planning and control of multi-robot systems.

Motion planning of multiple robots is still challenging since it has to address numerous important aspects ranging from modeling of workcell to optimal trajectory generation. The problem becomes more challenging with increasing number of robots, complexity of tasks to be performed by robots, the number of constraints related to robots, stationary obstacles, and the presence of other robots in the workspace. Attempts made by researchers on development of motion planners for coordinated motion planning of multiple robots are reviewed in the following section to provide the basis for developing a flexible and generic approach that can address the generation of conflict free coordinated motion plan for multiple robots.

A typical motion planning system for coordinated motion of multiple robots must essentially consider various aspects ranging from modeling of robots and the environment to the generation of optimal trajectory for achieving any specific task with multiple robots. The conventional motion planning systems consider all these aspects in a sequential fashion by devising strategies dependent on (a) the geometry used for robot and environment modeling (Hwang *et al.*, 2005), (b) schemes used for collision avoidance (Bennewitz, *et al.*, 2002; van der Berg and Overmars, 2005; Fraichard, 1999) (c) strategies adopted for coordination (O'Donnell and Lozano-Perez, 1989; Cheng, 1995; Kant and Zucker, 1986). Because of interdependency of these approaches on various aspects of motion planning, these integrated approaches are compact in structure. Despite of their integrated nature, these approaches cannot be employed as generic and flexible systems for motion planning of multiple robots. To address the issues related to conservative sequential approaches, agent based systems have emerged as versatile systems due to their plug and play architecture. Fig.1 shows the sequential nature of conventional motion planning systems which are inflexible. Altering any one of the modules in the system endangers entire system whereas the agent based systems consist of independent modules which can be coupled to address a simple to complex motion planning issues.

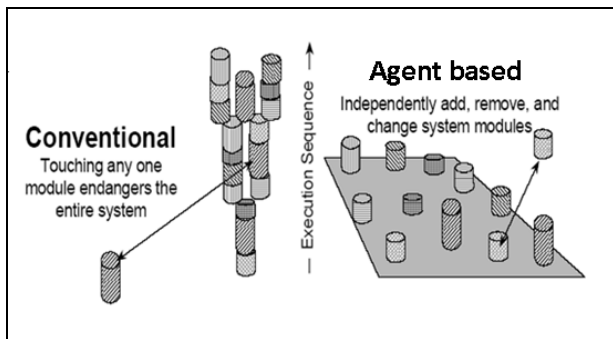


Figure 1. Flexibility associated with agent based systems

2. Agent Based Systems

The theory of agent based systems goes decades back when research in distributed artificial intelligence had been initiated. Along with the information and communication technology, the application of agent based systems is becoming very popular in manufacturing decision making and control systems. While this novel paradigm has several roots as far as its theory, enabling technologies and applications are concerned, there is a general consent about its two main concepts:

- ◎ An agent is a computational system that is situated in a dynamic environment and is capable of exhibiting autonomous and intelligent behavior.

- ◎ An agent may have an environment that includes other agents. The community of interacting agents, as a whole, operates as a multi-agent system.

The definition of an agent, agent categorization methods and the previous attempts to develop agent based systems for robot motion planning are discussed in the next section.

2.1 Agent Definition

Agent based approach for system modeling and control is relatively new concept. Agents in these systems are active and persistent components that perceive, reason, act and communicate. Generally, the agents are developed as a part of multi-agent system than a single entity. An agent can be defined as "An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives" (Wooldrige 2002). Moreover, an agent can be considered as a logical and autonomous entity which tries to contribute to the overall task achievement based on its own strategy (Dastani *et al.*, 2002; d'Inverno, 1998; Jennings and Wittig, 1992).

An Agent provides an abstraction for structuring and developing control system that offers certain benefits and is suited to certain types of applications. The effectiveness of agents in control system design relies on the mappings of the distinctive features of an agent into the properties of a control system. One of the major advantages of an agent-based system is the ability of agents to decouple from one another so as to eliminate cross coupling. Hence, an agent is a self-contained entity that achieves its goals and tries alternatives to adapt to certain changing environment. Agents require minimal supervision and human intervention.

2.2 Basic Characteristics of Agents

An agent operates in an environment from which it is clearly separated (Fig. 2). Hence, an agent (1) makes observations about its environment, (2) has its own knowledge and beliefs about environment, (3) has preference regarding the states of the environment, and finally, (4) initiates and executes actions to change the environment.

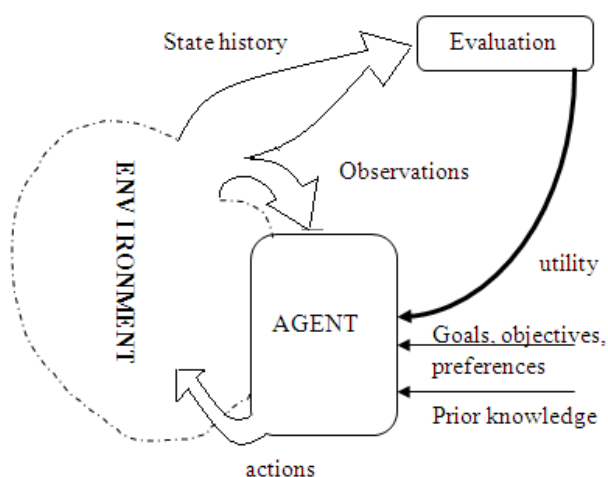


Figure 2. The agent and its environment (Russel and Norvig, 1995)

The most important common properties of agents are as follows (Monostori *et al.*, 2006):

- ❖ Agents act on behalf of their user in order to meet a particular purpose.
- ❖ They are autonomous in the sense that they control both their internal state and behavior of environment.
- ❖ They exhibit some kind of intelligence, from applying fixed rules to reasoning, planning and learning capabilities.
- ❖ They interact with their environment and other agents.
- ❖ They are ideally adaptive, i.e. capable of tailoring their behavior as per the changes in the environment.

2.3 Agent Categorization

The capabilities of an agent are determined by its design. Several researchers have given different designs of an agent to facilitate the control of robot systems. Agent can be categorized as functional agent and integrated agent. Functional Agent is capable to perform some specific function while an Integrated Agent performs holistic function for a robot system (Bastos-Filho *et al.*, 1998).

With the functional agent categorization approach, it is easy to define agent capabilities due to the direct mapping of the corresponding functionality. The interaction of agents is exclusive due to the unambiguous requirement of information for a particular behavior. The functional agent categorization approach achieves flexibility and extensibility by the addition of agents for extension of system capabilities and behavior. The functional agent categorization approach accommodates heterogeneous agent design that is incapable for homogenous agents design.

The integrated agent categorization approach is to design an agent based system with integrated behavioral agents that achieves modularity and homogeneity. Each agent consists of its own sensor, behavior and actuator. Each agent senses the environment and drives the assigned actuator with its own strategy. This approach attempts to map directly between the hardware and the control software (Lau and Ng, 2004).

2.4 Agent Based Systems for Robot Motion Control

Agent based systems are very widely used in control of various manufacturing systems and it has become inevitable for meeting the requirements of modern manufacturing systems (Papakostas *et al.*, 1999; Chrysosolouris *et al.*, 2000). Though agent based systems are gaining importance in manufacturing system modeling and control, attempts made towards application of these system for motion planning and control of robot are limited. In literature, very few attempts can be seen towards developing an agent based system for robot motion control.

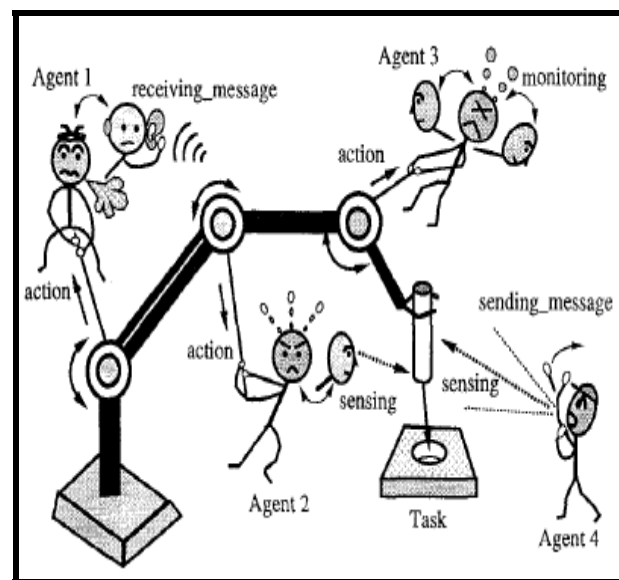
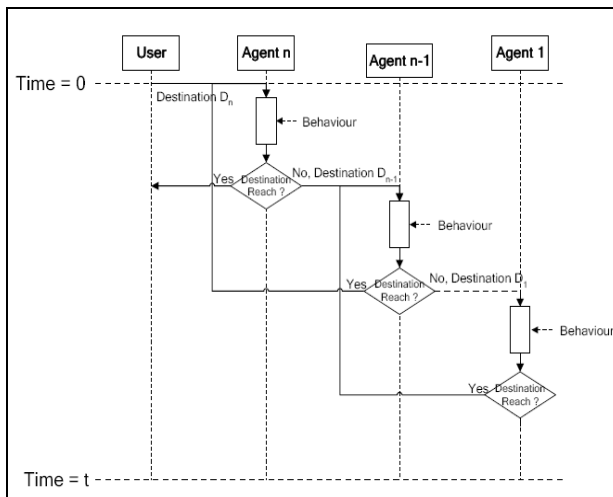


Figure 3. Task execution by 4 agents using task decomposition (2D) (Mori *et al.*, 1996)

In the agent based system proposed by Mori *et al.* (1996), four agents were deployed and each individual agent participated in execution of a peg-in-hole task with a robotic arm (Fig. 3). Three of them, could only control one joint that is assigned to them. Fourth agent monitored the execution of tasks and reported the status to other agents. The joint controlling agents decide and carry out their own behavior based on the information obtained from sensors and other agents. Precise trajectory of the manipulator was unknown beforehand because the actions are concurrently and independently performed. Thus, the behavior is a result of agent interactions.



nature and geometry of obstacles, and task specification as input and gives out collision free and conflict free trajectory for all the participating robots in order to perform the specified task. In this work, each robot and every motion planning activity is viewed as an agent. These agents are categorized as a) Functional agent b) Integrated agent based on the task to be undertaken by them. The activities like modeling of robot and its environment, generation of Configuration space Time (CST), Coordinated motion planning, estimation of inverse kinematics solutions, determination of optimal trajectory are supposed to perform some specific task and hence each of these activities is considered as a Functional Agent. Robots are responsible to perform holistic function for entire system and hence each robot is considered as an Integrated Agent. The functional agent can obtain required information from other agents as per necessity, whereas the integrated agents share the information among them using CST. The detailed description of these agents and new strategies developed for Functional Agents are explained here.

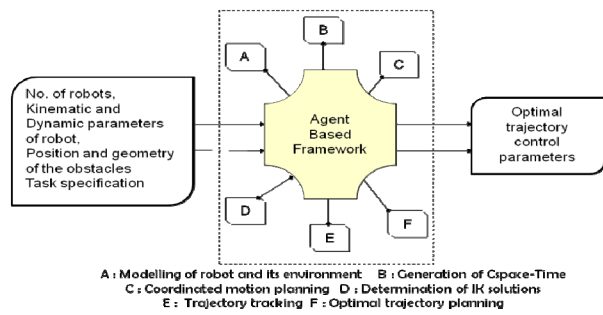


Figure 6. Outline of the proposed agent based framework for coordinated motion planning of multiple robots

3.1 Integrated Agent - Robot

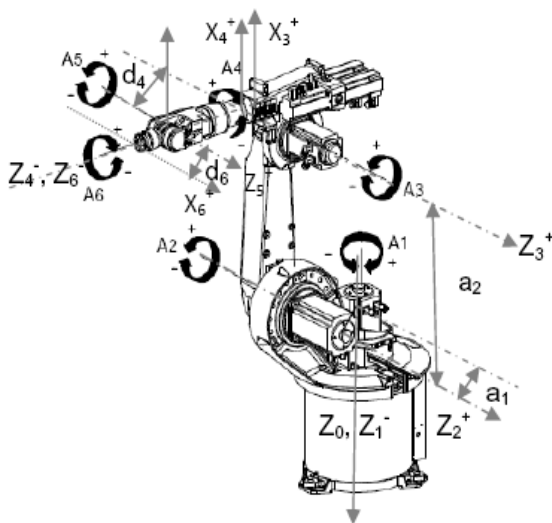


Figure 7. Coordinate frames attached to the rigid body of KUKA-Kr-15 robot

The proposed framework can consider any revolute type of robot whose kinematic and dynamic parameters are known. This is essential to plan the path and trajectory for the robots involved in coordinated task. In this work, Kuka Kr-15 six degree of freedom, serial fixed based manipulator (Fig. 7) is considered since its kinematic parameters, DH parameters and dynamic parameters are known (Verdonck,2004).

3.2 Functional Agent A – Modeling of Robot and Workspace

Effective modeling of robot and obstacles present in the workspace is important to map the workspace more accurately. By this, the accuracy of path planned for the tasks can be enhanced. In this work, Swept Sphere Volume (SSV) technique is employed to model the elements of work cell environment since this technique is capable of modeling various links of robot, objects and obstacles more realistically (Harden et al., 2005). A set of algorithms based on SSV are developed to determine the safe distance between an obstacle and a link of robot, links of robot and link of a robot and rectangular platform (Chiddarwar and Ramesh Babu, 2009a). Thus, this particular *Agent A* considers kinematic and geometric parameters of the robot, the nature and position of obstacles in the environment to model the robot and its workspace (Fig. 8).

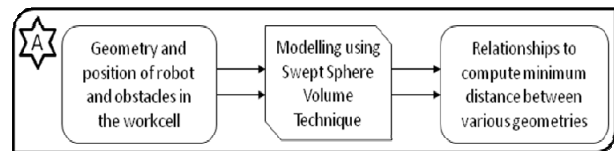


Figure 8. Modeling of robot and its environment using SSV (*Agent A*)

3.3 Functional Agent B – Configuration Space -Time Generation

It is well known that the path planning approaches consider configuration space for planning the collision free path for a single robot performing the tasks in a constrained environment. However, the coordinated path planning approaches need the configuration space with time (CST) to plan for collision free path for multiple robots since each robot is viewed as a dynamic obstacle by other robots during the task planning. By using the algorithms proposed (*Agent A*) for estimating the minimum distance between various entities, collision free configurations of each robot with respect to stationary obstacles are first determined and these configurations are then subjected to singularity checks. The collision and singularity free configurations are transformed into Configuration space-time (CST) by considering the linear velocities of end effectors of robots. All these tasks are performed by *Agent B* (Fig. 9).

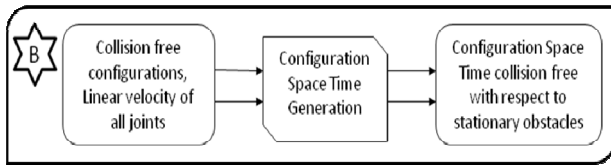


Figure 9. Determination of CST (*Agent B*)

3.4 Functional Agent C- Decoupled Coordination

The task of coordination among the robots is planned by a decoupled path planning approach proposed in this work. The proposed approach, covered with *Agent C*, consists of two phases: a) path planning phase and b) coordination phase.

Path Planning Phase: Collision free configurations of each robot, provided by Agent B, are used to determine an optimal collision free path for each robot using well known heuristic search algorithm i.e. A* algorithm. This path of each robot with respect to time and CST of each robot determined by *Agent B*, is made available as intelligent CST during motion planning of multiple robots.

Coordination Phase: Optimal collision free path obtained for each robot can safely be used only when the paths of other robots do not conflict with the path of considered robot. In case of multiple robots, conflicts can occur between two robots and more than two robots. In case of conflict between any two robots in a multi-robot system, the conflict configuration is removed from the CST of these two robots. It is then searched with incremental A* algorithm (Koenig and Likhachev, 2002) to determine the modified path for each robot. Based on the length of modified path for each robot, the path of robot having the shortest path length is chosen for modification and the path for coordination of multiple robots is obtained. In case of conflict among three robots in a multi-robot system, the present work proposes a path modification sequence (PMS) to arrive at the sequence of robots for conflict free path determination. It makes use of incremental A* algorithm for modifying the path of each robot conflicting with other robots. The cost of PMS is determined by using the length of modified path. For each PMS, cost of PMS = cost of PMS with priority – cost of PMS without priority. For this computation, the cost of respective paths were taken into account as for example $\ell(R_2R_1R_3) = [\ell(R_2R_1) + \ell(R_2R_1R_3)] - [\ell(R_1) + \ell(R_3)]$. Among all the available PMS, the minimum cost PMS is selected and this optimal PMS automatically yields the optimal set of path modification sequence to resolve a conflict situation. In case of 'n' robot agents conflicting in the environment, the issue of conflict resolution can be attempted by identifying the groups of two or three robots conflicting among the many robots (Chiddarwar and Ramesh Babu, 2009b).

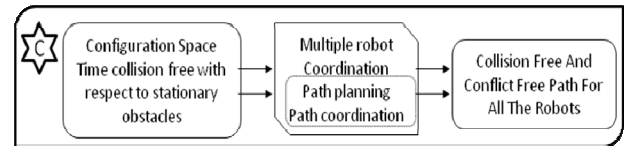


Figure 10. Schematic of two phase approach developed for coordinated motion planning of multiple robots (*Agent C*)

3.5 Functional Agent D- Computation of Inverse Kinematics Solutions

Collision and conflict free path provided by Agent C can be implemented only when this path is converted into joint angle configurations. Normally, the Cartesian path is transformed into joint angles by inverse kinematics. Unlike geometric and algebraic methods, the artificial neural network seems to be promising since it is independent of the geometry of robot and initial solution. For quick and precise transformation of Cartesian path into joint angle path, a fusion process is proposed in this work (Fig. 11). The proposed fusion approach is the blend of artificial neural network and forward kinematics relations of robot. It uses radial basis function neural network (RBF) for prediction of joint angles. In order to estimate the deviation in predicted value from the desired value, the pose at that configuration is computed using forward kinematics relations. The proposed algorithm utilizes this information and continues to improve the solution. In order to develop the neural network, the required database is obtained from forward kinematics relations. (Chiddarwar and Ramesh Babu, 2010). Thus, the task of determining the joint angle configuration for movement of robot end-effector along the path obtained by using *Agent C* is performed by *Agent D*.

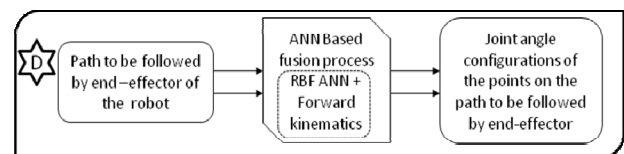


Figure 11. Schematic of fusion approach proposed for estimation of IK solutions (*Agent D*)

3.6 Functional Agent E- Trajectory Tracking

Estimation of kinematic control parameters is necessary to move the robot end-effector along the specified path. For this purpose, a neural network based approach is proposed in this work (Fig. 12). The proposed approach generates the necessary data for training of neural network by fitting fifth order B-spline to the entire CST of each robot. The corresponding velocity, acceleration and jerk are computed by taking derivative of the spline. This database is further used for training of radial basis function neural network, which takes joint angle path of robot in time domain as an input and predicts the values of velocity, acceleration and jerk at that particular configuration (Chiddarwar and Ramesh Babu, 2009c).

Hence, this **Agent E** computes the kinematic control parameters i.e. velocity, acceleration and jerk for the robot agent end-effector moving along the specified joint angle path obtained by using **Agent D**.

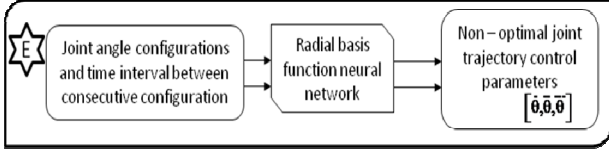


Figure 12. Outline of neural network approach developed for trajectory tracking (**Agent E**)

3.7 Functional Agent F – Optimal Trajectory Planning

The **Agent E** is capable of finding out the velocity, acceleration and jerk for the robot movement along the specified path without considering the dynamics and torque constraints of a robot. However, for multiple robot motion planning, optimal trajectory planning is important to ensure safe and fast task execution by robots along the coordinated path with minimum consumption of resources within dynamic, kinematic and payload constraints. In this work, a generic approach is proposed for optimal trajectory planning which computes optimal velocity and acceleration necessary for precise and appropriate task execution independent of nature of task. For this purpose, a multi objective optimization function as shown in the equation (1) is formulated with the consideration of objectives such as minimization of torque, travelling time, acceleration, jerk, gripping forces and maximization of manipulability subjected to various kinematic and dynamic constraints (Chiddarwar, 2010).

$$Z_{\min} = w_r * \sum_{i=1}^{vp-1} h_i + w_j * \sum_{j=1}^N \int_0^{t_f} (\ddot{\theta}_j(t))^2 dt + w_v * \sum_{j=1}^N \int_0^{t_f} (\dot{\theta}_j(t))^2 dt + w_a * \sum_{j=1}^N \int_0^{t_f} (\ddot{\theta}_j(t))^2 dt - w_s * |\det j| + w_p * \sum_{j=1}^N \int_0^{t_f} (F_1(t)v_1(t) + F_2(t)v_2(t))dt \quad (1)$$

Subject to constraints

$|\theta_j| \leq \theta_j^{\max}$ Displacement constraint, $|\dot{\theta}_j| \leq \dot{\theta}_j^{\max}$ Velocity constraint

$|\ddot{\theta}_j| \leq \ddot{\theta}_j^{\max}$ Acceleration constraint, $|\ddot{\theta}_j| \leq \ddot{\theta}_j^{\max}$ Jerk Constraint

$|u(t)| \leq u_j^{\max}$ Force / Torque constraint, $F_{g_{\min}} \leq F_g \leq F_{g_{\max}}$ payload constraint for $k=1,2$

$|\theta_{j,i}| = \theta_{j,i}^{\text{via-point}}$ Via-point constraint for $i=1, \dots, \text{number of via-points}$

$h_i = \max_{j=1, \dots, N} \left\{ \frac{|\theta_{j,i+1} - \theta_{j,i}|}{\dot{\theta}_j^{\max}} \right\}$ lower bound constraint for execution time

where,

Z_{\min} = multiobjective objective function, N = number of joints of the robot

vp = number of via points, h_i = Time interval between consecutive via points

$\theta_j(t)$ = displacement of j^{th} joint, $\dot{\theta}_j(t)$ = velocity of j^{th} joint

$\ddot{\theta}_j(t)$ = acceleration of j^{th} joint, $\ddot{\theta}_j(t)$ = jerk of j^{th} joint

The torque computation is done by considering the dynamics of robot as well as the gripping force necessary to grip the payload with two finger gripper. The formulated objective function and constraints are

represented in terms of coefficients of spline used for modeling the trajectory and are further used as optimization variables for optimization process. The values of these variables are determined by means of a multi objective genetic algorithm. The computed values of these variables are further used to estimate the values of kinematic control parameters. Thus, the function of **Agent F** is to model the trajectory using quintic B-spline and to estimate the optimal trajectory control parameters using multiobjective genetic algorithm (Fig. 13).

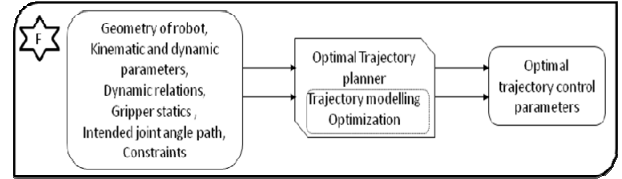


Figure 13. Schematic of the proposed approach for optimal trajectory generation (**Agent F**)

4. Results and Discussion

In this work, an agent based framework is developed for coordinated motion planning of multiple robots. The various agents of the system and their functionalities are identified and new approaches are developed to impart flexibility with the plug and play nature of modules into the framework. The performance of proposed approaches for realising the functionality of various agents is compared with commonly used approaches and the results thus obtained are discussed in this section.

In order to evaluate the performance of the proposed decoupled approach for coordinated motion planning of multiple robots (**Agent C**), a four robot workcell is considered (Fig. 14). The desired collision free coordinated path is determined for all three robots using the proposed approach as well by a) assigning fixed priorities to the movement of robot, b) varying the velocity of the robots and c) introducing delay in the startup of the robot. The typical example shown in Fig. 14 (a) consists of four robots sharing a common workspace. The objective is to move the end-effector of robot agent R1 from the start position P1(-852.53,89.06,365) to the goal position P5 (909.30,-50.47,350.16), robot agent R2 from the start position P6 (-930.80,160.79,412.93) to the goal position P8 (488.20,-751.56,455.68), robot agent R3 from the start position P9(-703.22,138.62,428.76) to the goal position P12(629.16,-4.85,456.73) and robot agent R4 from the start position P13(-350.23,-273.57,457.84) to the goal position P15 (415.93,91.9,362.25). The robot agents, various objects and obstacles in the workspace are modeled with **Agent A**. **Agent B** determines the collision free CST for each robot agent by utilizing collision checking algorithms developed by **Agent A**. This CST is utilized by **Agent C** to generate collision and conflict free

path for each robot. The two phase decoupled approach developed for *Agent C* generates the collision free path shown in Fig. 14 (b) as an output of the path planning phase. This path is further used by coordination phase in order to resolve probable conflicts arising due to sharing of common workspace by multiple robot agents. The conflict free path obtained after resolving the conflicts is shown in Fig. 15. The performance is evaluated in terms of the time taken by each agent to execute individual task, the total time required for complete task execution and the computational time. From the results presented in Table 1, it is clear that the proposed approach considerably reduced the time required for individual task execution and eventually the time required for executing the entire task. The proposed approach ensures concurrent movement of robot with optimal velocity without introducing any delay in startup of the robot. With the proposed approach, the robots have to move a little longer path. Though it takes more computational time than other approaches, the reduction in travelling time will become more prominent for coordinated motion planning of multiple robots in off-line mode.

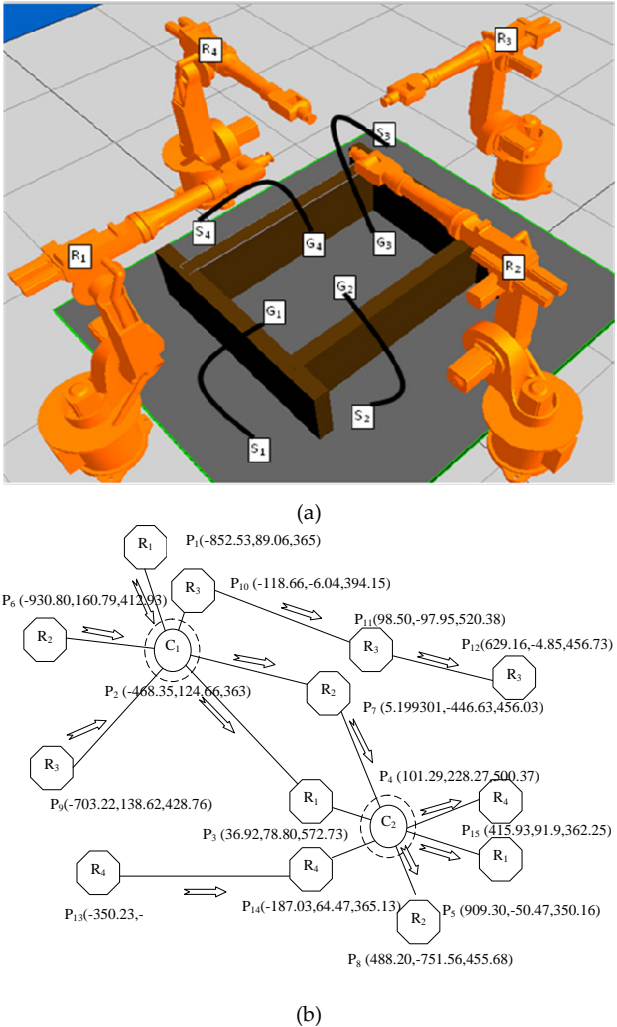


Figure 14. (a) Four robot workcell (b) Collision free path for four robots with respect to stationary obstacles

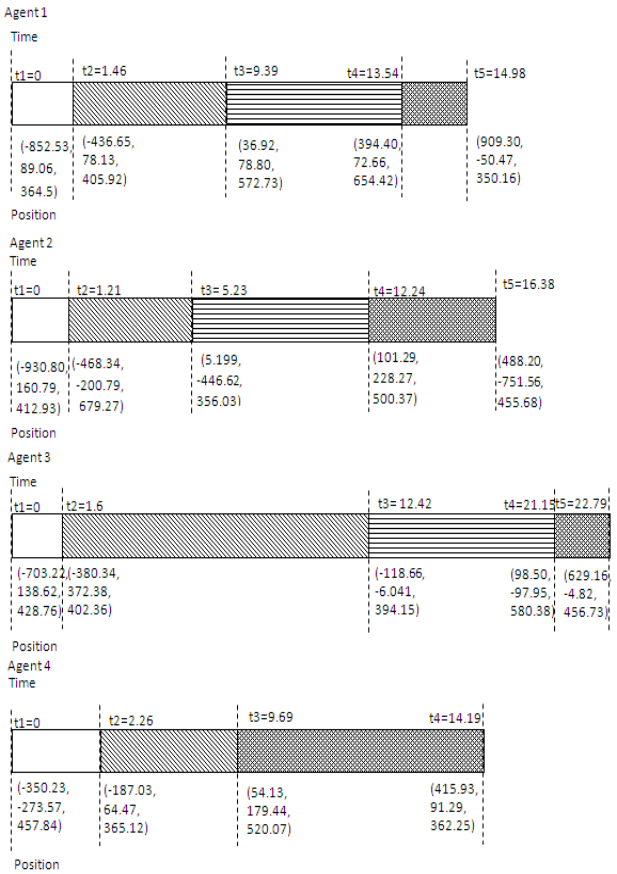


Figure 15. Conflict free coordinated path obtained for four robot agents after resolving conflicts (Output of the coordination phase of *Agent C*)

Robot	Distance travelled (m)	Coordination Techniques				
		Fixed priority	Velocity reduction	Delay in startup time	Path modification based on optimal PMS	
		Travelling time (s)	Travelling time (s)	Travelling time (s)	Distance travelled (m)	Travelling time (s)
R ₁	1.83	11.64	23.28	21.64	1.954	14.98
R ₂	2.17	28.02	16.38	16.38	2.179	16.38
R ₃	1.92	47.19	46.73	41.74	1.957	22.79
R ₄	1.72	60.93	28.06	29.03	1.729	14.19
Total Execution Time (s)		147.78	46.73	108.79		22.79
Computational time (s)		15.18	30.85	45.54		67.486

Table 1. Comparison of performance of various coordination techniques used to determine conflict free path for robot agents of four robot case study

In order to assess the performance of the proposed fusion approach for estimation of inverse kinematic solution (*Agent D*), a sample path having five nodes is considered. IK solutions at these nodes are determined by using the proposed fusion approach as well as the conventional

approaches. The fusion approach is realized using by RBF ANN as well as MLP ANN in order to determine the appropriate neural network algorithm. The performance of various neural network architectures is compared based on the RMS error in prediction of joint angles. From the results obtained, it is seen that RBF ANN predicted joint angles more accurately than those predicted with MLP ANN. Moreover, the error in prediction of joint angles using the proposed fusion approach is considerably lower than the error observed with conventional approaches. This can be attributed to the training data obtained from forward kinematics relations of the robot that helps to avoid the trapping of solution at local minima and minimizes the mapping error in IK solutions.

The requisite trajectory along the specified path is determined by using *Agent E*, *Agent F* and the control scheme developed by Falconi and Melchiorri [29] for generation of the trajectory/ kinematic control parameters. For this purpose, Kuka Kr-15 robot is modeled by considering dynamics and kinematics of the robot using SimMechanics toolbox [30]. The robot model and various functions from RobotiCad toolbox are interfaced in the Simulink of MATLAB [31] in order to estimate the trajectory control parameters along the specified path when joint angle configurations and time is given as the input. The strategy developed for *Agent E*, *Agent F* and the control scheme based on RobotiCad toolbox is implemented to determine the trajectory of the four robots working in the common workspace. The performance of these three methods is compared based on the total time (Table 2) and torque required for completion of the task (Table 3). From the results presented in Table 3, it is seen that the effect of considering the dynamics of robot on torque required to complete the task is considerable. In addition, the optimization of trajectory helps to reduce the total time and the total torque required for completion of the task.

Case 2	Distance travelled (m)	Coordination Strategy (<i>Agent C</i>) (s)	RobotiCad (s)	<i>Agent E</i> (s)	<i>Agent F</i> (s)
R ₁	1.954	14.98	12.1	13.44	13.33
R ₂	2.179	16.38	18.19	19.76	17.43
R ₃	1.957	22.79	23.47	23.78	22.89
R ₄	1.729	14.19	15.56	20.20	13.18

Table 2. Time required for the completion of the task by using velocity profiles generated by various trajectory generation techniques

Case 2	RobotiCad (Nm)	<i>Agent E</i> (Nm)	<i>Agent F</i> (Nm)
R ₁	1658.8	1621.4	910.1
R ₂	2155.8	1655.8	1348.2
R ₃	637.7	883.2	980.7
R ₄	935.8	1426.9	734.5

Table 3. Computation of the total torque by using various trajectory generation techniques

5. Major Observations

The developed agent based framework consists of various agents classified based on their functionality. In order to make the developed approach flexible, the agents responsible for coordinated motion planning, generation of inverse kinematics solutions and trajectory generation are of major importance. Hence, new approaches are developed to achieve the functionality of these agents. The major findings of this work are as follows:

- ❖ The proposed approach of dynamic determination of PMS for modifying the path to avoid conflicts and achieve desired coordination is efficient than other approaches as it does not allow any deadlock situation to happen. In addition, this approach facilitates concurrent movement of all the robots with optimal velocity. The proposed approach shows around 40% reduction in total time required for complete task execution and 12 - 33% reduction in the individual time taken by the robot as compared to other approaches liked fixed priority, velocity reduction, delay in startup time.
- ❖ The RBF ANN based fusion process for determination of IK solutions seems to be effective in terms of accuracy of solution and time required for determination of the solution. The error in prediction of joint angle is in the range of 0.06 to 0.11 degrees which is very less as compared to MLP ANN and conventional neural network architectures.
- ❖ The trajectory tracking method made use of quintic B-spline for modeling of trajectory and RBF ANN with 7by6 ANN architecture for prediction of the kinematic control parameters shows 0.4-6 % error in the solution whereas when computation is done with algebraic spline error is 0.9-18%. This method is very handy for the determination of the kinematic control parameters when the dynamics of robot need not be considered.
- ❖ The optimized trajectory obtained along the specified path by using quintic B-spline for modeling of trajectory and genetic algorithm for optimization process seems to be efficient than trigonometric spline modeling and sequential quadratic programming for

optimization. When trajectory estimation is done by using quintic B-spline for modeling of trajectory and GA based optimization technique then 11-43% reduction in total computed torque is obtained.

6. Conclusions

A systematic study on multi-robot coordinated motion planning and its important aspects showed the effectiveness of agent based framework developed in this work. Among the various agents necessary for successful implementation of the proposed framework, the agents responsible for achieving coordination, estimation of inverse kinematics solutions and optimal trajectory planning are very important. The strategies developed for these agents ensure collision free, conflict free and smooth movement of multiple robot agents sharing the common workspace. This framework can be used for motion planning of multiple robots independent of task to be performed and the number of robots sharing the common workspace. The effective utilization of resources, minimum time and energy consumption are the highlights of the proposed approach, which helps to introduce sustainability in modern reconfigurable and flexible manufacturing systems.

7. References

- [1] Hwang, K., Ju, M., and Chen, Y., 2003, Speed alteration strategy for multijoint robots in co-working environment. *IEEE Trans. on Industrial Electronics*, 50, 2, 385-393.
- [2] Bennewitz, M., Burgard, W., and Thru, S., 2002, Finding and optimizing solvable priority schemes for decoupled, path planning techniques for teams of mobile robots. *Robotics and Autonomous Systems*, 41, 2, 89- 99.
- [3] van der Berg, J. P. and Overmars, M. H., 2005, Prioritized motion planning for multiple robots. *International Conference on Intelligent Robots and Systems*, Edmonton, Canada, Aug 2-6, 2217-2222.
- [4] Fraichard, Th., 1999, Trajectory planning in a dynamic workspace using state time space approach. *Advanced Robotics*, 13 (1), 75-94.
- [5] O'Donnell, P. A. and Lozano-Perez, T., 1989, Deadlock-free and collision-free coordination of two robot manipulators, *Proc. IEEE International Conference on Robotics and Automation*, 484-489.
- [6] Cheng, X., 1995, Online collision free path planning for service and assembly task by a two arm robot. *International Conference on Robotics and Automation*, Nagoya, Aichi, Japan, 1523-1528.
- [7] Kant, K. G. and Zucker, S. W., 1986, Towards efficient trajectory planning: path velocity decomposition. *International Journal of Robotics Research*, 5, 72-89.
- [8] Wooldrige, M. *An Introduction to Multiagent Systems*, John Wiley, 2002.
- [9] Dastani, M., de Boer, F., Dignum, F., van der Hoek, W., Kroese, M. and Meyer, J.-J. Ch. (2002) Implementing Cognitive Agents in 3APL. *Proceedings of the 14th Belgian-Dutch Conference on Artificial Intelligence, BNAIC2002*, Leuven, Belgium, pp. 515-516
- [10] d'Inverno, M., Kinny, D., Luck, M. and Wooldridge, M. (1998) A Formal Specification of dMARS, In *Intelligent Agents IV: Proceedings of the Fourth International Workshop on Agent Theories, Architectures and Languages*, Singh, Rao and Wooldridge (eds.), *Lecture Notes in Artificial Intelligence*, 1365, pp. 155-176, Springer-Verlag.
- [11] Jennings, N. R. and Wittig, T. (1992) ARCHON: Theory and Practice. In: *Distributed Artificial Intelligence: Theory and Praxis*. Kluwer Academic Press, pp.179-195.
- [12] Monostori, L., Vancza, J., Kumara, S.R.T. (2006) Agent based systems for manufacturing, *Annals of the CIRP*, 55, 697-720.
- [13] Russel, S., and Norvig, P. *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995.
- [14] Bastos-Filho, T.F., Freitas, R.A.C., Sarcinelli-Filho, M. and Schneebeli, H.A. (1998) An agent-based structure for mobile robots using vision and ultrasonic sensors. *Proceedings of the IEEE International Symposium on Circuits and Systems*, May31- June 3, 6, 602 - 609.
- [15] Lau, H. Y. K. and Ng, A. K. S. (2004) Immunology-based Control Framework for Multi-jointed Redundant Manipulators; *Proc. 2004 IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Singapore
- [16] Papakostas, N., Mourtzis, D., Bechrakis, K., Chrysosouris, G., Doukas, D., and Doyle, R., 1999, A Flexible Agent Based Framework for Manufacturing Decision-Making, *Proceedings of the 9th Flexible Automation and Intelligent Manufacturing Conference*, Tilburg, The Netherlands, June 23-25, 789-800.
- [17] Chrysosouris, G., Papakostas, N., and Mourtzis, D., 2000, A Decision-Making Approach for Nesting Scheduling: A Textile Case, *International Journal of Production Research*, 38/17, 4555-4564.
- [18] Mori, A., Naya, F., Osato N., and Kawaoka, T., 1996, Multiagent-based distributed manipulator control. *Proc. of International conference on Multisensor Fusion and Integration for Intelligent Systems*, Washington DC, USA, 289-296.

- [19] Ramdane-Cherif, A., Levy, N., Djenidi, H., and Tadj, C., 2002, Agents paradigm to solve a complex optimization problem. Proc. of IEEE International Conference on Cognitive Informatics, Calgary, Alberta, Canada, 247-256.
- [20] Bohner, P. and Luppen R., 1997, Reactive Multi-agent Based Control of Redundant Manipulators, Proceedings of the IEEE International conference on Robotics and Automation, Albuquerque, New Mexico, 20-25 April 1997, 2, 1067 - 1072.
- [21] Verdonck W., 2004, Experimental Robot Payload Identification with Application to Dynamic Trajectory Compensation, Ph.D thesis, Katholieke University, Belgium.
- [22] Harden, T., Kapoor, C., and Tesar, D., 2005, Obstacle avoidance influence coefficients for manipulator motion planning. Proceedings of ASME IDETC/CIE. Long beach, USA, Sep 24-28, 1-13.
- [23] Chiddarwar, S. S. and Ramesh Babu, N., 2009, Off-line decoupled path planning approach for effective coordination of multiple robots, Robotica, doi:10.1017/S0263574709005773, Accepted for publication.
- [24] Koenig, S. and Likhachev, M., 2002, Incremental A*. Journal of Advances in Neural Information Processing Systems, 2 (14), 1539-1546.
- [25] Chiddarwar, S. S. and Ramesh Babu, N., 2009, Dynamic priority allocation for conflict free coordinated manipulation of multiple agents, Proceedings of IEEE Conference on Automation Science and Engineering, Bengaluru, Aug. 22-24, 549-554.
- [26] Chiddarwar, S. S. and Ramesh Babu, N., 2009, Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6R serial robot by a fusion approach, Engineering Applications of Artificial Intelligence, doi:10.1016/j.engappai.2010.01.028, Accepted for publication.
- [27] Chiddarwar, S. S. and Ramesh Babu, N., 2009, Agent based framework for coordinated motion planning of multiple robots, 7th Global Conference on Sustainable Manufacturing, IIT Madras, Chennai, Dec 2-4. 373-378.
- [28] Chiddarwar, S. S., 2010, Agent based framework for coordinated motion planning of multiple robots, Ph.D. Thesis, Supervisor: Prof. N. Ramesh Babu, Indian Institute of Technology Madras, Chennai, India.
- [29] Falconi, R. and Melchiorri, C., 2008, RobotiCad: an Educational Tool for Robotics. Proc. of the 17th World Congress of Automatic Control, Seoul, Korea, 9111-9116.
- [30] SimMechanics Toolbox for MATLAB: User's Guide, The Mathworks Incorporation, 2007.
- [31] Simulink Toolbox for MATLAB: User's Guide, The Mathworks Incorporation, 2007.