Industrial Robot: An International Journal
40/4 (2013) 329–336
Emerald Group Publishing Limited [ISSN 0143-991X]
[DOI 10.1108/01439911311320831]

Control and Programming of a Multi-Robot-Based Reconfigurable Fixture

C. Zieliński, W. Kasprzak, T. Kornuta, W. Szynkiewicz, P. Trojanek, M. Walęcki, T. Winiarski, T. Zielińska

Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland

Abstract
Purpose: Machining fixtures must fit exactly the work piece to support it appropriately. Even slight change in the design of the work piece renders the costly fixture useless. Substitution of traditional fixtures by a programmable multi-robot system supporting the work pieces requires a specific control system and a specific programming method enabling its quick reconfiguration.
Design/methodology/approach: The multi-robot control system has been designed following a formal approach based on the definition of the system structure in terms of agents and transition function definition of their behaviour. Thus a modular system resulted, enabling software parameterisation. This facilitated the introduction of changes brought about by testing different variants of the mechanical structure of the system. A novel approach to task planning (programming) of the reconfigurable fixture system has been developed. Its solution is based on constraint satisfaction problem approach. The planner takes into account physical, geometrical, and time-related constraints.
Findings: Reconfigurable fixture programming is performed by supplying CAD definition of the work piece. Out of this data the positions of the robots and the locations of the supporting heads are automatically generated. This proved to be an effective programming method. The control system on the basis of the thus obtained plan effectively controls the behaviours of the supporting robots in both drilling and milling operations.
Originality/value: The shop-floor experiments with the system showed that the work piece is held stiffly enough for both milling and drilling operations performed by the CNC machine. If the number of diverse work piece shapes is large the reconfigurable fixture is a cost-effective alternative to the necessary multitude of traditional fixtures. Moreover, the proposed design approach enables the control system to handle a variable number of controlled robots and accommodates possible changes to the hardware of the work piece supporting robots.

Keywords Industrial robot, Programming, Machineability, Sheet metal, Reconfigurable fixtures, Multi-robot systems, Multi-agent systems, Robot control

Paper type Research paper

1. Introduction

Although currently the robotics research effort is focused on service and field robots, there are still many complex industrial tasks that need the application of robots. One of such areas is the substitution of the current costly fixtures by multi-robot systems supporting large-sized work pieces in the aircraft industry. For technological reasons, if large-sized thin-sheet components have to be manufactured they are first pressed into an appropriate 3D shape and then subjected to machining (e.g. drilling, milling). Machining requires rigid support of those flexible complex-shaped sheets (panels), thus either equally complexly shaped support fixtures have to be manufactured for each type of work piece or manually reconfigurable fixtures have to be used. The former are extremely expensive and the latter require considerable reconfiguration time resulting in the work cell being idle, hence non-productive for a long period.

The existing flexible fixtures for thin-walled work pieces, i.e. modular flexible fixture systems (MFFS), and single structure flexible fixture systems (SSFFS) provide limited reconfigurability (Sela ,2007; Shirinzadeh, 1995). However, most MFFSs still require some human intervention to reconfigure and few are suitable for thin sheets. Another approach to automate reconfiguration is the use of an SSFFS of the pin-bed type, with a matrix of supports, which provides support comparable to a mould-like fixture. The main disadvantages are high cost, and a lack of modularity, which makes them difficult or inefficient to use for parts of differing sizes. Robotic fixtureless assemblies (RFAs) replace traditional fixtures by robot manipulators equipped with grippers that can cooperatively hold the work piece (Bi, 2001; Kang, 2002). Using RAFs different parts can be manufactured within one work-cell and transitions to other work pieces can be done relatively quickly. However, existing and proposed RAFs are suited only for rigid and relatively small parts, as they use a limited number of traditional non-mobile robot manipulators. The proposed fixturing system combines the advantages of RFAs with those of MFFSs, namely: the ability to distribute the support action, adaptability to part shapes in a larger range and high stiffness of the provided support.

It was noticed that in reality the work piece has to be fixed very firmly only in the vicinity of the location that the machining takes place. This observation led to an idea of producing a self-reconfigurable fixture composed of a swarm of robots able to relocate themselves under the sheet cumulating the support only in the current place of machining (fig.1) (Molfino, 2009). The number of the supporting robots may differ for panels of different size; however the control system should remain unchanged. This paper shows what should be the features of the controller of such a system. As the number of the supporting robots cannot be specified a priori, a multi-agent approach to the design of the control system structure was adopted. The specification was based on earlier works (Zieliński, 2006; Zieliński, 2010) fostering agent approach and transition function description of the behaviour of each agent, following the operational semantics (Slonneger, 1995) approach to the description of programming languages.

2. The concept of a reconfigurable fixture

The proposed system works in the following way. The panels (thin sheet work pieces) are introduced into the fixture manually. Static clamps fix the panel initially and locate it accurately, except for sagging due to gravity. The supporting robots move to the vicinity of the place of machining to rigidly support the panel so that the CNC machine can perform its operation in that location. The robots consist of three elements (fig.1): mobile base translocating the robot under the panel, manipulator, being a parallel kinematic machine (PKM), rising and lowering the third element, i.e. the supporting head. The robots relocate themselves over a bench, which contains docking elements. The docking elements form a mesh of equilateral triangles. Once the robot is securely locked to the bench the PKM lifts the supporting head to the prescribed location, thus providing support to the machined metal sheets. Initially the head is soft, but once it has reached the prescribed location it is solidified and vacuum is applied to suck the sheet to the head, thus holding it firmly.

An off-line program, on the basis of CAD geometric data describing the panel and CAM prescribed machining procedure, generates the plan of relocation of the robots. The panels are either subjected to drilling or milling. Drilling is performed while the supporting robots are immobile. Milling requires the robots to relocate during machining. The supervisory controller uses the plan produced off-line to control the robots during machining. It also influences the controller of the CNC machine, thus synchronizing the actions of the robots and the CNC machine.

3. Devices requiring control

The mobile base of each robot contains three legs located in the vertices of an equilateral triangle (Molfino, 2009). Each leg can be lowered or lifted by a two-state pneumatic actuator. Its pneumatic valves are electrically operated. All the legs are able to rotate around the vertical axes through the actions of one electric motor coupled to the legs through a common gear. The legs are locked to the pins located in the bench by Schunk docking devices, which are pneumatically operated. When the robot supports the metal sheet its three legs are docked to the pins. During translocation two locks are opened, the legs are withdrawn (lifted), and the remaining leg docked to the bench is rotated. The manipulator placed on top of the mobile base can be rotated independently by a second electric motor. Hence the mobile base requires the control of two motors, three pneumatic actuators and three pneumatic locks. The manipulator (PKM) (Neumann, 1988; Zoppi, 2010) has 6 degrees of freedom equally divided between the parallel kinematic structure and the spherical wrist. Thus six motors need to be controlled. The head contains a magneto-rheological fluid that is either solidified or liquefied by shifting a permanent magnet, which is translocated by a pneumatic actuator (Avvenente, 2010). Moreover, the application of vacuum has to be controlled. Both pneumatic devices require just on-off control. The head can also be rotated, thus one additional motor must be controlled. The bench, besides providing the support to the robots, is responsible for delivering to them the power supply and compressed air. Both are provided through the supporting pins. This is done by on/off devices.

4. Control system structure and system behaviours

All of the motors of a robot are controlled by MAXON EPOS2 motion controllers connected to its on-board computer through a CAN bus. EPOS2 controllers are dual-powered in order to prevent resetting during anticipated (but never noticed) short breaks of power supply: motor windings are supplied with 48V DC which is provided separately of stabilized 24V DC power supply to the control logic. All devices requiring binary control are operated by the binary outputs of the EPOS2 motion controllers. Thus 9 such controllers suffice for each robot (6 for the PKM motors, 2 for the mobile base motors and 1 for the head motor). The power supply switches located in the bench are operated separately by a digital input-output card in the host computer, while the pneumatic valves controlling the airflow to the chip cleaning nozzles are operated through a separate CAN bus connected to the host computer. The host computer and the on-board computers communicate through the wireless Ethernet. The software distributed over this network is agent-based.

Each robot (j) is controlled by three agents (a_j), each one of them is dedicated to one of the above described three components: manipulator (a_{j1}), mobile base (a_{j2}), and the head (a_{j3}) -

fig.2. In general j=1,…,n, but in the presented case n=2. A monolithic control system could have been designed, but this would impede independent development of the software for each of the components. Moreover, initially it was anticipated that the target system will be frequently modified with the evolution of the project. Under such circumstances modularity of the software is a necessity. As the three components were developed separately three separate agents have been created for each robot. Thus the total number of agents is three times the number of robots plus a system coordinator and an extra agent that directly controls the activities of the bench (a_{n+1}). The software of the control system was formally specified describing the behaviours of the agents in terms of transition functions (Zieliński, 2006). The robot-based agents, in principle, can communicate with each other, however in this case only agent-coordinator communication suffices.

The agent controlling a manipulator (a_{j1}):
1) moves the PKM in such a way that the head is shifted to a predefined support position,
2) supports the panel,
3) lowers the head. In the folded configuration the agent remains dormant. In that state its mobile base can relocate the robot.
The first and the third operation are executed by the same behaviour – motion to a position defined by the argument of the command issued by the coordinator. The second one does not require any activity from the agent – the robot remains motionless.
The agent controlling a mobile base (a_{j2}):
1) unlocks two legs from the pins,
2) withdraws those two legs
3) rotates itself on the leg docked to the pin,
4) lowers the two legs,
5) docks the two legs.
In the locked configuration it remains dormant, thus five behaviours are needed. In this state the manipulator can move the head or support the panel. The mobile base operates on outstretched legs.
The agent controlling a head (a_{j3}):
1) rotates the head into the desired configuration,
2) switches on the vacuum,
3) solidifies the contents of the head,
4) supports the panel,
5) switches off the vacuum,
6) liquefies the contents of the head.
Supporting the panel does not require any action, thus again five behaviours suffice.
The agent controlling the bench (a_{n+1}):
1) switches on or off the power supply delivered through each of the connecting pins in the bench to the robots attached to those pins,
2) turns on or off the pressure of the air used for blowing off the chips falling on the individual pins.
The agent (a_0) coordinating the actions of the whole system is responsible for:
1) interaction with the operator,
2) reading-in of the plan formulated by the off-line program,
3) coordinating the actions of the agents controlling the robots by issuing appropriate commands,
4) controlling the agent switching the power supply and air supply through the bench to the robots.

5. Controller

The structure of each agent of the system follows the general guidelines regarding embodied agents (fig.3) (Zieliński, 2006; Zieliński, 2010). In general each agent has the capability of: commanding its effector, configuring its exteroceptors, delivering information to other agents, as well as updating its own memory. This is done by its transition functions on the basis of the proprioceptive information obtained from the effector, aggregated readings from the exteroceptors, information delivered by other agents and its own memory record. However in this particular case the agents did not employ exteroceptors and the contact with other agents was limited only to interchange of information with the coordinator. Moreover this interchange took place sporadically – only when the coordinator commanded a new behaviour, after the previous one had been completed. This significantly simplified the inner structure of the agents. The structure of the agents directly controlling the hardware (i.e. embodied agents) is expressed by the pseudocode presented in fig.4, where → represents data transmission between the components of the agent or between the agents, j is the number of the agent (0 for the coordinator), e is the effector, T is the transmission buffer, c is the control subsystem of the agent, x and y refer to the input and output buffers of the control subsystem, f_c is the transition function, f_τ is the terminal condition, and m is the reference number of the pair of those two functions. Each such pair defines a behaviour of the agent for the duration of the execution of the inner loop of this code. The coordinator (agent 0) designates those functions for execution invoking different behaviours of the agent as required by the task. Each of the behaviours of each of the embodied agents enumerated in section 4 requires a single pair of those functions. They were defined formally and subsequently implemented according to those formal specifications. The complexity of the specification of the transition function depends on the required behaviour and the representation of the controlled device. In the discussed case the most complex behaviour was the Cartesian motion of the PKM, taking into account both the velocity and acceleration limits and the capabilities of the device executing the generated trajectory, i.e. the EPOS2 motion controllers (Zieliński, 2011). Terminal conditions are just predicates (Boolean valued functions), thus they are simple to specify and easy do implement.

Operation of the coordinator (agent 0) is specified by two behaviours only. The first one is dedicated to commanding the subordinate agents with new actions to execute, while the second is just to halt the system in the case of failure of any of its components. The coordinator is triggered by two kinds of events: an internal timer managed accordingly to the off-line generated plan and notifications from the embodied agents. Data sent from the coordinator to the individual agents consists of an identifier of the commanded action and its parameters (i.e. state of solidification and vacuum together with the orientation for the head, target pose for the PKM, element of the translocation sequence for the mobile base and set of pins to supply with power or compressed air, for the bench). In turn, all the agents reply with notification when the commanded action has been successfully completed or failure report in the case of any error. The coordinator makes a record of these events and detects whether deadlines specified by the plan have been met.

6. Planner

The work piece is a thin metal sheet formed into a 3D surface defined by the CAD data. This sheet is subjected to either milling or drilling. The vicinity of the area being machined must be held firmly during that operation, thus the heads of the robots must be placed not far away from each other and the tool performing the operation. For drilling holes in small areas a static

configuration of head locations suffices, but during drilling holes along a large contour and during a milling process the heads must be relocated fast to follow the CNC tool progress. The required stability of the work piece for a given machining process is achieved when a set of constraints on the relative location of the tool and the supporting heads is satisfied. For each agent, the supporting head can be placed in a continuous space limited by the workspace of the PKM and the current location of the mobile base, while the mobile bases can only be placed in discrete locations on the workbench, thus they belong to a finite set of locations, as determined by the docking and locomotion subsystems. Furthermore, the movement speed must not exceed the motor capabilities. Hence, it would be a very complex task for a human expert to try to create manually an action plan for even two mobile agents. Thus a specific dedicated automatic solution has been developed, implemented by a computer program, called the planner (Szynkiewicz, 2010). It is composed of four main parts (fig.5):
1) CAD/CAM data dependent *work piece analysis* module,
2) work piece- and agent data-independent *path planner* (a Triple-CSP planner, where CSP stands for the Constraints Satisfaction Problem (Szynkiewicz, 2010)),
3) agent-data-related *CSP-constraints* and
4) CAD/CAM and agent-dependent *time planner*.

The *work piece analysis* module is executed only once per work piece type. Its goal is to decompose the CNC-tool trajectory into segments. The path planning problem is converted into a discrete constraints satisfaction problem (Russell, 2002). A classic CSP is defined by means of a state space and a graph of constraints. A state consists of values assigned to a finite set of variables from finite domains. A solution to CSP is every complete state (i.e. all variables have assigned values) which satisfies the graph of constraints. Here, the path planner consists of the *planner control* module that exercises overall control over path and time plan creation. The three hierarchically arranged modules, called *Head-CSP*, *Base-CSP* and *PKM-CSP,* execute stages of the path planner corresponding to the three parts of the agent and implementing the so-called *Triple-CSP* search. They all use an incremental state-space search algorithm (a depth-first strategy with backtracking), to find corresponding head state paths, base state paths, and PKM state paths, respectively (fig. 6). The path planner explores physical and geometrical constraints only. Hence, the plan, if produced, will satisfy all known constraints, although it may not necessarily be an optimal one. For a particular planning problem we need to add domain knowledge (in the form of CSP-variable constraints) and provide some application-dependent code of functions generating a next node in the search tree and the one representing the stop criterion. Some local optimization criteria can be considered and expressed by properly ordering alternative CSP-variable assignments. It may happen that at some point (segment) the on-line execution of the plan must be stopped. This happens when a single plan for the entire work piece does not exist and the machining process must be split into several parts.

For the cooperating mobile agents the following constraints need to be considered:

1) Geometric constraints between agents and the work piece contour: expressing the necessary physical requirements for adequate support for the given work piece and machining process (e.g. minimum and maximum allowed distance from head to the contour, maximum allowed distance between two consecutive head locations).
2) The workspace of the PKM: used to check quickly feasible base-head position pairs (e.g. the minimum and maximum distance between central base axis and head center).
3) Geometric constraints between bases and PKMs: needed to avoid collisions between agents during base position transitions.

4) The inverse kinematics problem solution of the PKM: used when defining feasible PKM states for consecutive head positions.

Finally, a complete path plan (for both cooperating agents) need to be verified by the *time planner* module, as all actions specified in the path plan must be executed in time and in a given order in accordance with the CNC tool scenario. A time plan is performed, that maps action sequence indices onto the time axis, consistently with dynamic models of agent parts.

7. **Results**

A number of experiments have been conducted to assess the quality of operation of both the control system and the automatically generated plans. Individual agents had been found to operate correctly by executing motion commands of an exemplary plan step-by-step, before running complete sequences. Detailed performance metrics, which relate to the mechanical requirements of the systems (e.g. stiffness during support of work pieces or durations of the mobile bases translocations) depend on the mechanical parameters of the actuator devices rather than on the structure of the control system. Thus, focus of the experimental phase was on evaluation of the operation of the selected hardware components and the developed control software, as well as, results of the planning algorithms.

The control system was implemented using the MRROC++ (Zieliński, 1999; Zieliński, 2010) robot programming framework. All the modules were implemented in C++ and the real-time Linux operating system was used as a runtime platform. Modularity of the control software proved to be the key benefit during preliminary testing of the mechanical components and also during final integration of the system. The selected off-the-shelf components proved to operate correctly in spite of a significant electromagnetic noise generated by all the manufacturing devices installed on-site. Executed plans required up to several supporting poses and up to 10 translocations of the mobile bases. Similar trajectories have been used for both drilling and milling, as most of the work pieces required holes to be drilled only along the supporting contour. Typical test runs involved a continuous operation of the system for up to several minutes. Kinematic parameters of the system have been calibrated using the CNC machine as a measurement tool.

The performance of the system is determined by the duration of the repeatable sequence of the activities of the agent. The times presented here are measured on a purposefully excessively safe sequence produced for the initial tests of the system. Locking/unlocking of the robot from the bench (about 1.5s – this is twice as much as needed, as the two legs that had to be lifted were raised consecutively instead of in parallel and the pressure was significantly reduced), rotating the robot by a quantum of 60° (about 2s) and rising/lowering the PKM to/from the support position (about 3s). Both rising and lowering of the PKM are conducted in two phases – fast linear motion in configuration space and a slower linear motion in Cartesian space in the vicinity of the work piece. The length of the Cartesian segment, over which the PKM motion is fairly slow, was 6 cm instead of the necessary 2 cm The delays introduced by the wireless connection are in the order of milliseconds and do not influence the overall performance, since the communication between the agents is limited to coordination of the activities of the agent. The CPU and memory use during the operation of the system is minimal, since most of the calculations are performed off-line by the planner. The theoretically computed time of support for a single head, based on the parameters of the head and the machining velocity (head side length – 0.07 m, tool velocity – 0.017 m/s) is 4.2 s.

Taking into account the capabilities of the motors (maximum accelerations and velocities) and the motion distances the following times have been obtained: PKM Cartesian motion – 0.35 s, PKM configuration space motion – 0.65 s, base rotation including PKM contra rotation – 1.85 s, lowering the legs in parallel with maximum pressure applied 0.25 s. Each sequence is composed of lowering and lifting of the PKM, lifting and lowering the mobile base legs and rotation of the base, thus its duration is 2(0.35+0.65+0.25) +1.85 = 4.25 s. Hence the tool velocity would have to be reduced a bit or the capabilities of the agents increased (e.g. the head size increased).

The planning performance depends on how constrained the reconfigurable fixture parameters are (bench dimensions, agent dimensions and speed, PKM workspace volume) in comparison to the work piece and CNC properties (location w.r.t. the bench, number and type of corners of the CNC path) and material features (support force distribution in the neighborhood of the cutting or drilling tool). If, for instance, the work piece size is too small, its shape very complex or the required CNC tool velocity too high then the planning process will normally take more time and require more backtracking steps compared to a case where the ratio of the segment lengths to the head size is more favorable.

The planner in practice is able to find a feasible plan always, if only the plan exists. No plan can be found if the work piece is too small in relation to the size of the agent. On the base of the experiments, we can formulate the performance for the Head-, Base- and PKM-CSP. The head plan search efficiency directly depends on the simplicity of the work piece machining contour (sequence of edges). The edge lengths must be in appropriate relation to the head size and the angles between consecutive edges should not be too sharp (i.e. only one angle along the contour can be less than 60 degrees), otherwise the head will not be positioned sufficiently near to the contour, violating the support distance limits. A *regular* contour satisfies two properties: (1) edge regularity - for each edge within the contour, at least one head can be placed between the adjoining corners; (2) area regularity - the inner area bordering three consecutive edges is sufficiently large to include two consecutive heads. Edge regularity guarantees that a single edge will be supported by at least one head position, whereas area regularity warrants that for three consecutive short edges the corner angles will be sufficiently large to accommodate the head. The distance limits (for the head-to-contour distance and the head-to-head distance) should exceed $b\sqrt{3}/2$, where *b* is the head side length. If all of the above conditions are fulfilled the head plan will be found. In other cases no guarantee exists.

The Base-CSP search efficiency depends on an adequate relation between the radius of the agent's base (r_B) (and thus the bench pin distances) and the assumed limits (r_{min}, r_{max}) of the distance between the base center and the projection of the head center onto the base plane. These distance limits should correspond to the PKM workspace radius (r_{PKM}). The efficiency of base plan search grows (i.e. the number of expanded search nodes is kept low – no backtracking steps are needed) when the limits are more relaxed (e.g. $r_{min} <= r_B/2$ and $r_{max} >= r_B\sqrt{3}$).

The PKM-CSP search efficiency depends on the work piece curvature and the size of the PKM workspace. The vertical cross-section of the workspace can roughly be approximated by a triangle – if the head is over the border of the base (this corresponds to $r_{min} = r_B$) the height variability is the largest possible (e.g. 7 cm), but by moving the head outside of the base area, the height variability steadily lowers and finally reaches the outermost "corner" of this triangle where only a single position can be attained.

The time schedule of the path plan is obtained by using parametric models of all of the agent actions and the CNC tool operations. Most of the time is spent on the movements of the bases. This knowledge is already included in the Triple-CSP search, where the movements of bases are limited as much as possible and if needed the smallest possible rotations (by 60 deg.) are preferred. When some action of an agent cannot be completed in time (as required by the machining process) a warning is issued and this should lead to changes of the application parameters.

Plan generation for a complex contour containing 20 corners, satisfying the above constraints, takes about 1 minute for a planar contour and up to 15 minutes for a non-planar one. The times were measured for the planner implemented in Matlab and executed on a AMD Phenom II X3 2.6 GHz processor.

8. **Conclusions**

The current version of the software is capable of controlling any number of agents, so the future downscaling of the size of each agent and increasing their number will only necessitate changes of parameter values due to the reduction of the size of each agent, statement of how many agents are there in the system and what is their initial state (e.g. location). The support of larger work pieces requires inclusion of a greater number of robots. Each of the robots will differ from the others by a little, so some of the parameters in their on-board control systems need tuning (i.e. calibration of the kinematics model and the low-level PID motor controllers). The modular implementation of the control software makes it particularly easy to extend the system as most of the parameterisation is done with configuration files. The wireless Ethernet proved to operate correctly during the experiments, however, it should be substituted with an industrial version of the communication protocol.

The control and planner strategy will be extended to handle more than two mobile agents. We intend to add the possibility of proposing changes to the application parameters, when the current parameters (e.g. work piece location w.r.t. bench pins, work piece height, dimensions and workspaces of agents) do not enable the planner to produce a feasible plan.

Acknowledgements

The research within the SwarmItFIX project was funded by the 7th Framework Programme (Collaborative Project 214678). The partners within the project: University of Genova (Italy), Piaggio Aero Industries (Italy), Exechon (Sweden), ZTS VVÚ Košice a.s. (Slovakia), Centro Ricerche FIAT (Italy) and Warsaw University of Technology (Poland). This publication is supported by the statutory funds of Warsaw University of Technology.

References

Avvenente, R. and Khan, A. and Li, X. and Zoppi, M. and Zlatanov, D. and Molfino, R. (2010), "Development and Analysis of a Shape-Conformable Supporting Head for a Self-Reconfigurable Intelligent Swarm Fixture System", *ISR/ROBOTIK'10*, pp.792-799.

Bi, Z. and Zhang, W. (2001), "Flexible fixture design and automation: Review, issues and future directions", *International Journal of Production Research,* Vol. 39 No. 13, pp. 2867–2894.

Kang, Y. and Rong, Y. and Yang, J. and Ma, W. (2002), “Computer-aided fixture design verification”, *Assembly Automation,* Vol. 22 No. 4, pp. 350–359.

Molfino, R. and Zoppi, M. and Zlatanov, D. (2009), “Reconfigurable Swarm Fixtures.”, in *ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robot, ReMAR 2009*, J.S. Dai, M. Zoppi and X. Kong (Ed.), King's College London, UK June 22-24, pp.730-735.

Neumann, K. (1988), “US patent number 4732525”.

Russell, S. and Norvig, P. (2002), *Artificial Intelligence: A Modern Approach 2nd Ed.*, Prentice Hall, Upper Saddle River, N.J.

Sela, M. and Gaudry, O. and Dombre, E. and Benhabib, B. (1997), “A reconfigurable modular fixturing system for thin-walled flexible objects”, *The International Journal of Advanced Manufacturing Technology*, Vol. 13 No. 9, pp. 611–617.

Shirinzadeh, B. and Tie, Y. (1995), “Experimental investigation of the performance of a reconfigurable fixture system”, *The International Journal of Advanced Manufacturing Technology*, Vol. 10 No. 5, pp. 330–341.

Slonneger, K. and Kurtz, B. L. (1995), *Formal Syntax and Semantics of Programming Languages: A Laboratory Based Approach,* Addison-Wesley Publishing Company, Reading

Szynkiewicz, W. and Zielińska, T. and Kasprzak, W. (2010), “Robotized machining of big work pieces: localization of supporting heads*”, Frontiers of Mechanical Engineering in China*, Springer, Vol. 5 No. 4, pp. 357-369.

Zieliński, C. (1999), „The MRROC++ system.” in *First Workshop on Robot Motion and Control (RoMoCo’99) Proceedings*, Kiekrz, Polska, July 28–29, pp.147–152.

Zieliński, C. (2006), “Transition-function based approach to structuring robot control software.”, in *Robot Motion and Control: Recent Developments, Lecture Notes in Control and Information Sciences,* K. Kozłowski (Ed.), Springer Verlag, Vol.335, pp. 265–286.

Zieliński, C. and Winiarski, T. (2010), „Motion Generation in the MRROC++ Robot Programming Framework.” *International Journal of Robotics Research*, Vol.29 No.4, pp. 386–413.

Zielinski, C. and Kornuta, T. (2011), “Generation of Linear Cartesian Trajectories for Robots Using Industrial Motion-Controllers.”, in *16th International Conference on Methods and Models in Automation and Robotics, MMAR’2011,* 22–25 August, Międzyzdroje, Poland. IEEE Catalog Number: CFP11MMA-CDR, ISBN: 978-1-4577-0913-5, pp.62–67.

Zielinski, C. and Kornuta, T. and Trojanek, P. and Winiarski, T. and Walęcki M. (2012), “Specification of a Multi-Agent Robot-Based Reconfigurable Fixture Control System.”, in *8th International Workshop on Robot Motion and Control, RoMoCo’11*, June 15–17, Bukowy Dworek, Poland. K. Kozłowski (Ed.), Lecture Notes in Control and Information Sciences, Springer, Vol.422, pp. 171–182.

Zoppi, M. and Zlatanov, D. and Molfino R. (2010), “Kinematics analysis of the Exechon tripod.”, in *Proceedings of the ASME DETC, 34th Annual Mechanisms and Robotics Conference (MR)*. Montreal, Canada

Corresponding author
Tomasz Winiarski can be contacted at: tmwiniarski@gmail.com

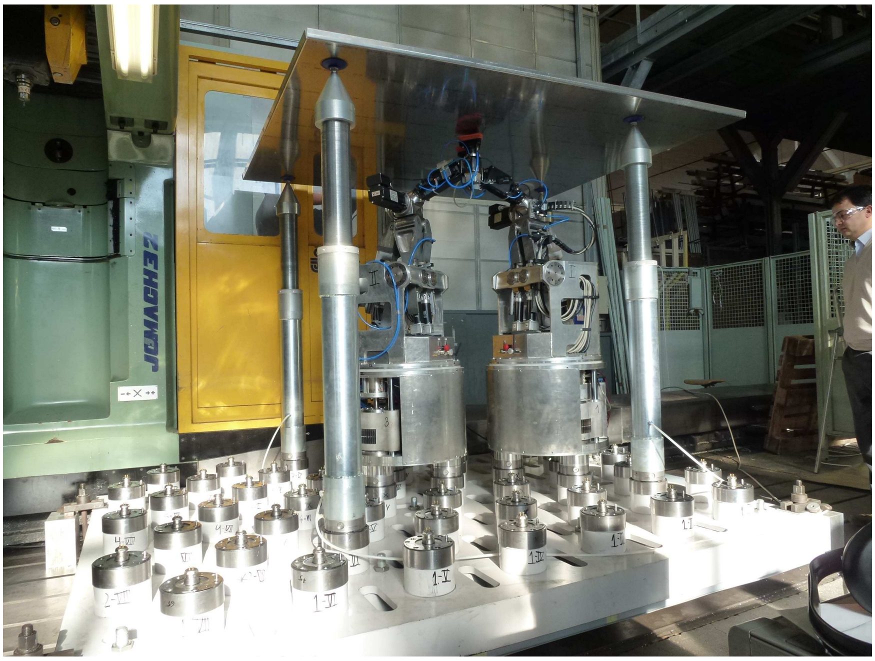


Fig. 1. The multi-robot fixture

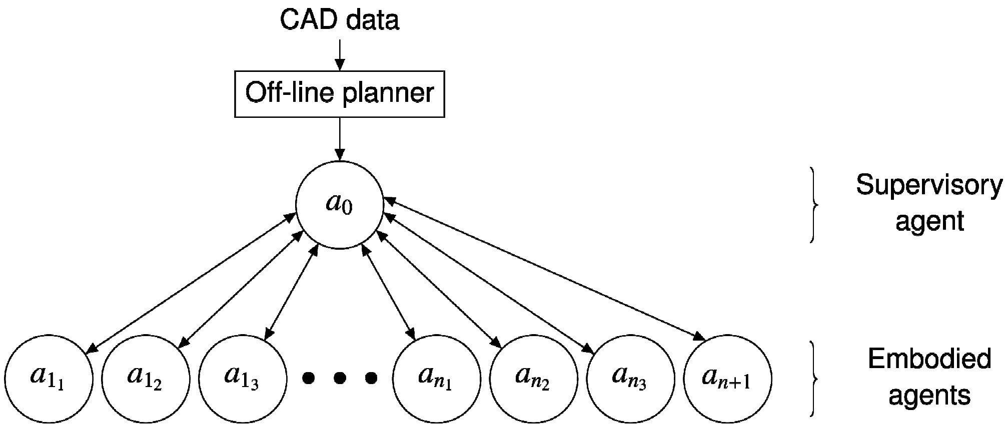


Fig. 2 Structure of the multi-agent control system

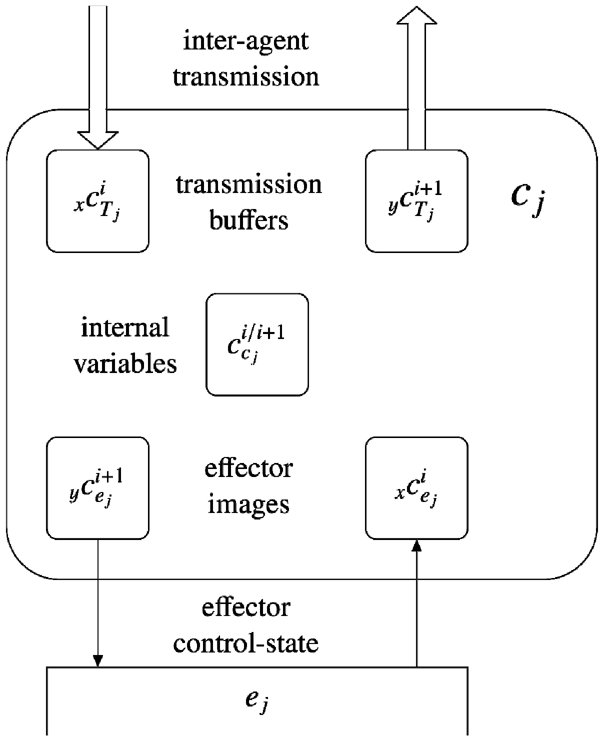


Fig. 3 Structure of embodied agents: a_{j1}, a_{j2}, a_{j3}

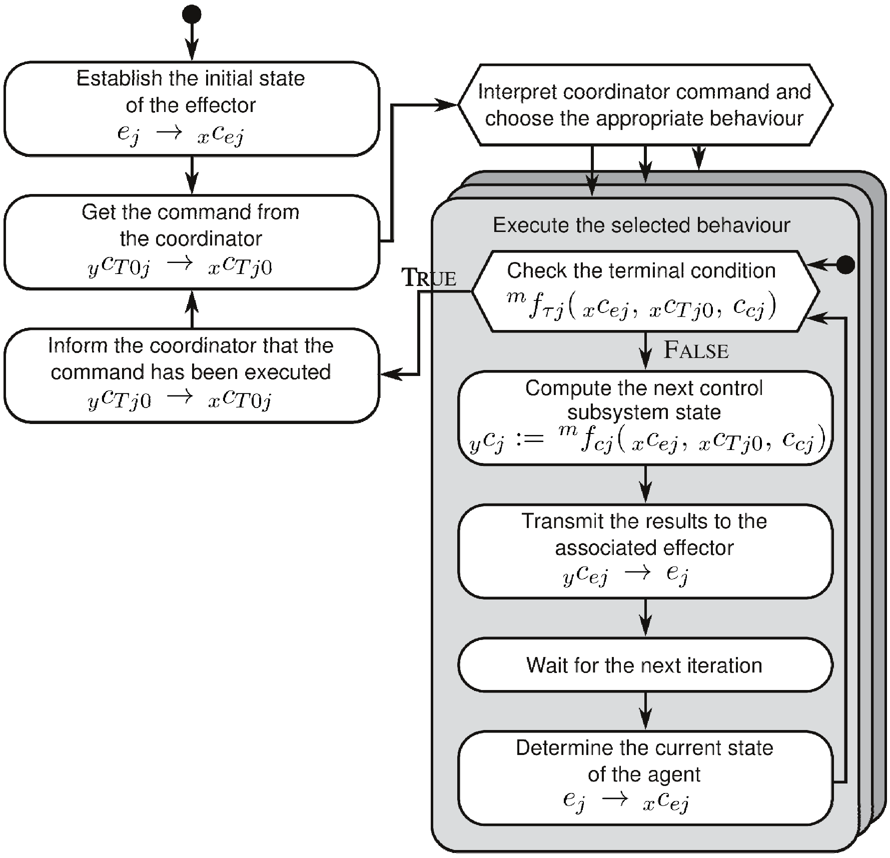


Fig. 4 General state diagram of each of the embodied agents

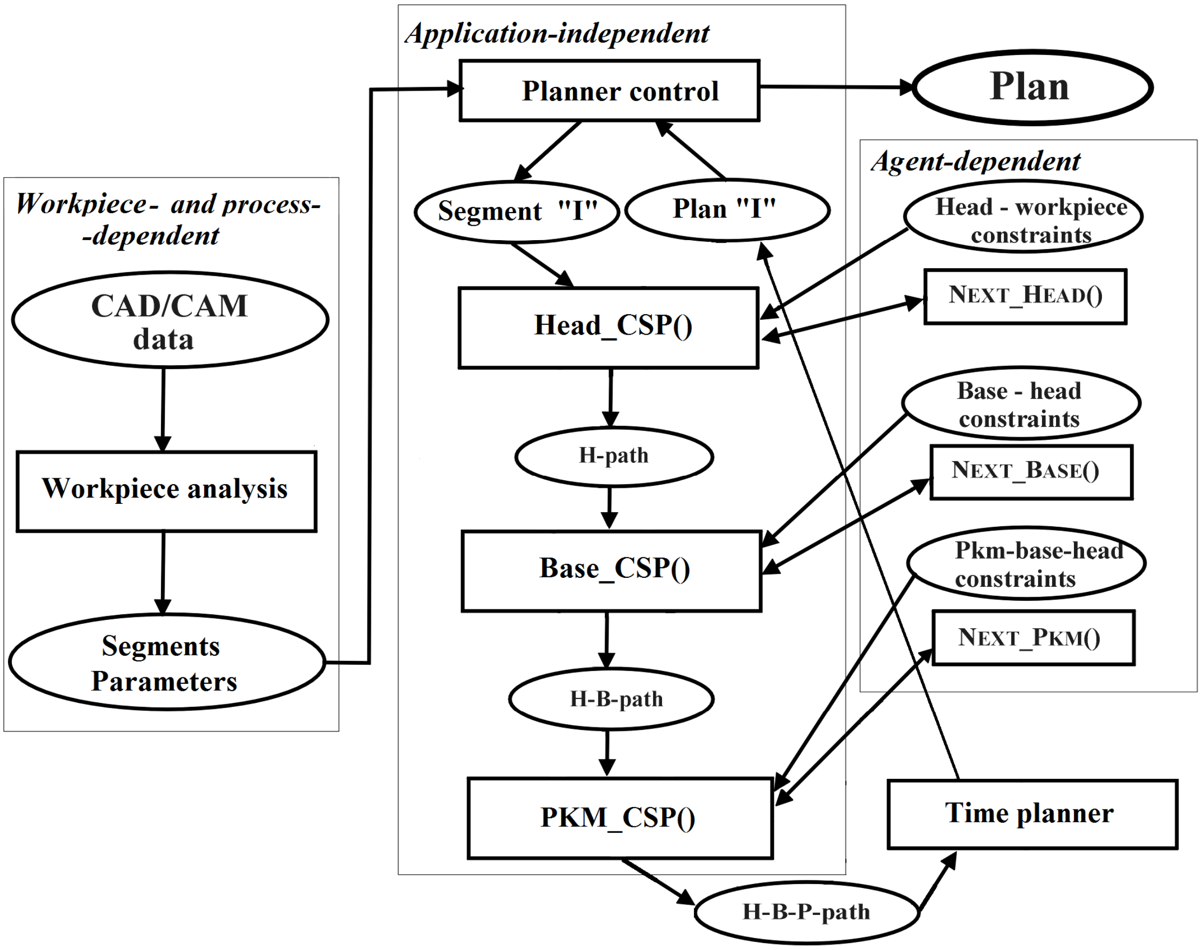
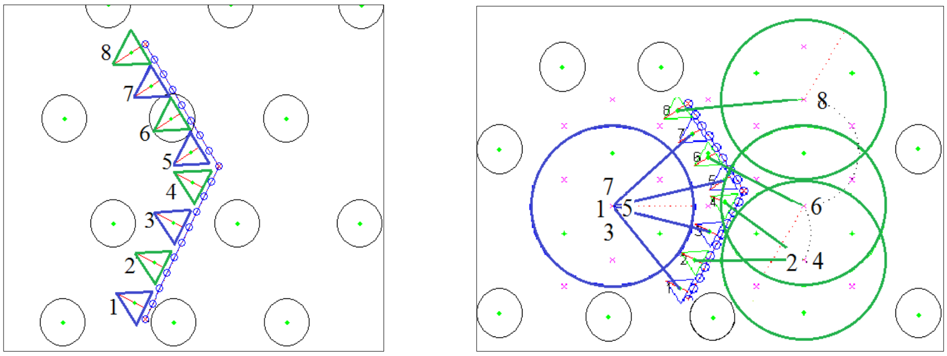


Fig. 5. The structure of the planner program



(a) (b)

Fig. 6. Illustration of a simple path plan (top view onto the bench): (a) a head plan for drilling holes along a contour line, (b) a corresponding base plan.