Robotic arm co-operation for assembling a reticular structure

M. De Bartolomei*, F. Grassini* and S. Losito**

*Tecnospazio – Milano, Italy

**Agenzia Spaziale Italiana – Matera, Italy

ABSTRACT

This paper describes a set of experiments performed in the frame of a work currently in progress on the ROSED laboratory testbed (RObotic SErvicing Demonstrator, to be installed by Tecnospazio at ASI "Centro di Geodesia Spaziale" plant in Matera) aiming to demonstrate the capability of two robotic arms to co-operate in order to automatically build reticular structures.

Keywords: co-operation, SPARCO, automatic assembling, truss

1. INTRODUCTION

The idea behind this work is to demonstrate the capability of a robotic system to autonomously perform complex assembling operation in environments in which human intervention is expensive or dangerous such in Extra Vehicular Activities. This is done using the ROSED Test-Bed (see Figure 1) whose kernel is a couple of conventional industrial robotic systems. Each arm is equipped with a gripper and Force and Torque (F/T) sensor, and its control SW has been integrated with the SPARCO SW (SPAce Robot COntrol, a project endowed by ESA in order to develop the basic robot control system for future space research programs and experiments). SPARCO implements the impedance control, which is suitable to allow the robots to accomplish tasks in which there is contact with the environment.

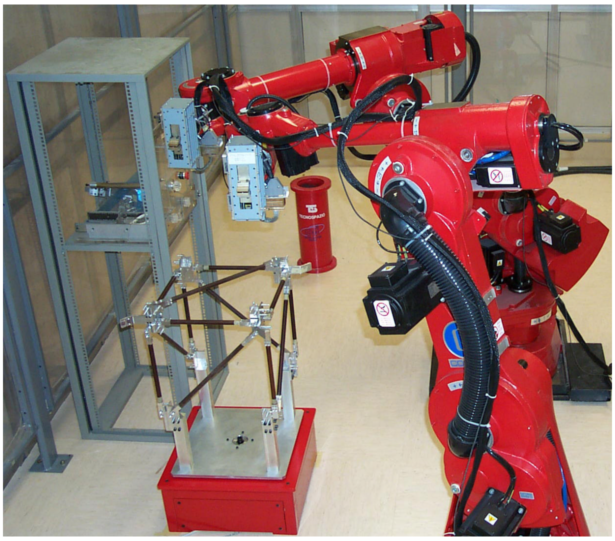

Figure 1: ROSED Test-Bed

Target of the experiments is to make the two above robots able to autonomously assemble a cube as basic item of a truss. Each face of the cube to be assembled is a square with a diagonal link. The edges of the square and the diagonal link are made of cylindrical rods: they are linked to each other through nodes which are placed at the vertices of the square.

The paper is organised as follows:
Section 2 gives an overview of the robotic system (HW and SW) and how the truss mechanical structure was designed to be properly assembled by the robots.
Section 3 describes the analysis made to identify the strategy to be followed by the robots during the assembling, also taking into account the need of co-operation.
Section 4 details the process of SW development to implement the above strategy.
Section 5 gives some brief conclusions and hints to areas which require further development.

2. EXPERIMENT LAYOUT

2.1 Robotic system description

The manipulator arms involved in the test-bed are commercial COMAU SMART S2 and S4, each driven by a C3G Plus Standard controller. The control unit is based on several processing boards connected among them via a VME bus: the robot CPU (RBC) is in charge of user's interface (I/F) handling, program translation and interpretation; inverse kinematics, trajectory generation and interpolation are demanded to the Servo CPU (SCC1).
The programming language adopted for this controller is PDL2, a Pascal-like language that provides powerful motion control instructions, multitasking, primitives for digital I/O handling and serial line I/F.
The extension for SPARCO is given by an additional SCC board (SCC2) which computes Cartesian set-point correction according to impedance algorithm control scheme. The arms are provided with commercial ATI F/T sensor, whose data is made available to SCC2.
C3G provides RS232 I/Fs: they are exploited to allow communication to End Effector (EE), to provide standard user I/F running on a PC and to communicate via Kermit protocol with another PC.
To aid truss assembling, the truss base is mounted on a rotating platform.

2.2 The SPARCO approach

Aim of SPARCO is to provide a robotic system with a set of functionality to deal with operations in which contact between EE and environment is involved and forces are generated due to uncertainty of the environment. The adopted control concept is the impedance control, aiming to assign a relation between such forces and displacement of EE w.r.t. the nominal desired position. During in-contact tasks, the control goal is to make the EE behave like a mass-spring-damper system, whose parameters can be arbitrarily specified. This is achieved calculating a correction in the Cartesian space prior to inverse kinematics and through an outer control loop independent from the inner one in charge of position control at joint level.

According to the CDM (Control Development Methodology, the ESA standard for Automation & Robotics development), the definition of this control concept followed the identification of a basic set of robotic operations called TASKS such as OPEN, CLOSE, INSTALL, REMOVE, ACTUATE FORCE. Tasks are in turn decomposed into a sequence of ACTIONS, being an Action uniquely mapped to a well precise control concept (pure position control such as DISPLACE TO or F/T control such as INSERT). The modularity of this approach allows the user to extend, according to the needs of the new application, the task and action library.

First, an *Activity Analysis* is performed to identify the new needed Tasks/Actions in terms of *attributes* (i.e. formalisation of user requirements such as initial/termination conditions).
The *Preparation phase* is the next step and is split into two sub-phases:

- code preparation (in PDL2 language) of Action and Task;
- SPARCO database preparation: it stores the whole information relevant to the work-cell layout.

In the SPARCO database the World Model is provided as a geometrical description of the environment, in particular of the subjects to be manipulated. This information is mainly given by means of *frames*. At manipulator level programming, frames represent a logical solution for describing interrelations among the objects, in which the robot can be considered only as a means to reach a desired target position.
World model is defined hierarchically so that frames logically grouped can be expressed w.r.t. a frame defined in the upper level: subjects are logically grouped in facilities, so that a parameterisation of tasks and actions is possible.

Table 1 summarises all the relevant frames: frames are *fixed* if their numerical value is constant regardless of the associated subject position since expressed relative to *movable* frames. With each element (i.e. facility) of the workcell a Facility F frame is associated. F frame is defined w.r.t. to the robot base frame (${}^{B}T_{F}$). Relative to the F frame, the location of associated objects is described by means of Objects O frames (${}^{F}T_{O}$). On_pose and Off_pose frames represent the initial and the final position of the work piece frame P, being the P frame the location (${}^{O}T_{P}$) of a well specified point affixed to the object to be manipulated. The meaning of initial and final position is strictly related to the status of a subject (another variable stored in the database), which indicates also which task can be executed (for example, if the status is ‘full’, only the REMOVE_FROM Task can be executed).
Approach and Grip frames are fixed since expressed relative to the P frame.

The SPARCO database is designed so that tasks which undo each other effect can be thought as the reversal sequence of actions of the other, so same set of data is used for both tasks. The database is updated at the end of each Task or Action in order to reflect the new status of the environment. The OPEN (a drawer, for example) Task is split into the action sequence reported in Table 2 (note that SLIDE is considered terminated only when the force due to drawer stroke-end is sensed). Simply switching On_pose and Off_pose, the database is ready to be used by the CLOSE (of the same subject) Task, which has the same structure.
Another couple of dual tasks is REMOVE_FROM / INSTALL_IN (see Table 2): the action sequence is slightly different due to the fact that the object shall respectively remain attached or detached at the end of the task. However, the concept related to the definition of frames associated to initial and destination position is still the same.

2.3 Truss mechanical description

Each element of the TRUSS was designed with more than one goal:

- to enable its manipulation from the robot
- to be light weight
- to have a high modularity

The result has been that the entire TRUSS is based on three items:

- Node
- Lateral Rod
- Diagonal Rod

Nodes and rods interface between them via two pegs mounted on the node which have to slide into holes placed at the end of the rod (see Figure 2).

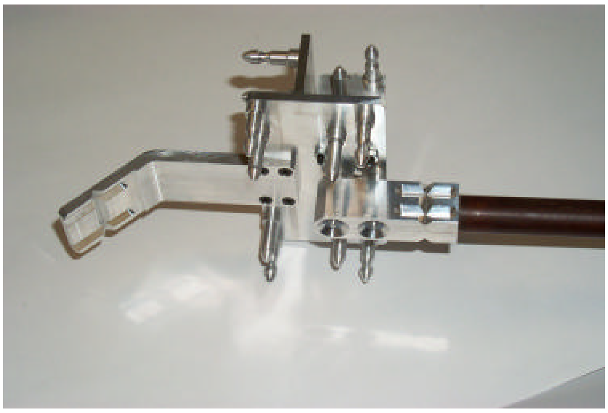

Figure 2: Node and rod

To ensure rigid connection, a locking mechanism triggers at the end of insertion. To enable their handling by the robots, rods and nodes have been designed with a suitable mechanical interface which fits with the EE. They can be transported, inserted, held and pushed by the two robots (see Figure 3). Particular effort has been put to have the structure symmetric: when the cube is mounted, all the lateral faces appear equal, and this suggests the design of an iterative algorithm.

In order to be assembled, nodes and rods have to be taken from the relative repositories by the two arms. Stable supports for the cube to be built are four columns, at top of which the four lower nodes are mounted. The columns are attached to a base which can be rotated of steps of 90 degrees through a motor connected and driven via I/O signal by S2 arm controller.

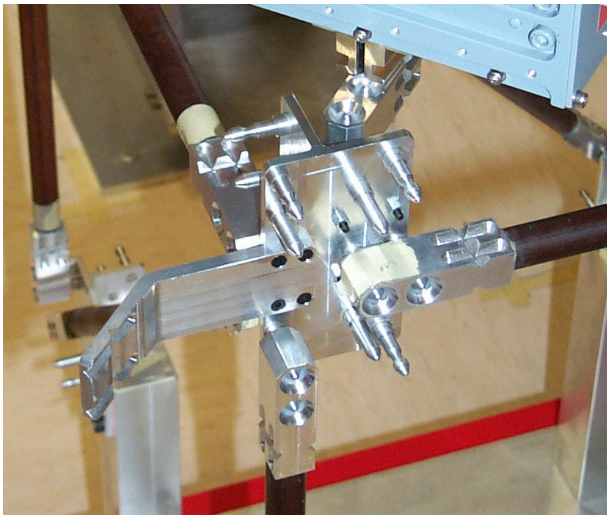

Figure 3: rod pushed to be connected

3. TRUSS ASSEMBLY ANALYSIS

3.1 Task analysis

The cube assembling operations are supported by the revolving base which allows the two robot to operate always at the same way to build the lateral faces of the cube and to deal with working space constraints. Therefore the Activity Analysis has identified a set of logical operations (called compound tasks) in which the overall robotic system (2 robots and rotating base) are able to modify the test-bed layout from an initial to a final condition. Columns and vertical faces are numbered anti-clockwise. Vertical edge 1 (2) is, regardless of base rotation, the closest to S2 (S4) arm. Face n is between edge n and either edge n+1 (if n<4) or edge 1 (if n=4).

Table 3 summarises the Compound tasks to be implemented: they are parametric w.r.t. either the edge or the face to be assembled. Table 4 demonstrates that a sequence involving assembling of the same edge and face leads to the assembling of the cube. VEn stands for Vertical Edge n, VFn stands for Vertical Face n, TF stands for Top Face. For each row, 'X' mark in a column means that the related item is assembled at the end of the correspondent compound task.
Critical operation in the above-defined compound tasks is the connection of a rod to a node: high force has to be exerted to lock the connection. This is not always possible since the force applied by the robot results in the structure displacement, due to its flexibility, instead of exceeding friction of the locking mechanism.
In order to deal with this, one of the two robots transports and inserts either the rod or the node while the other holds the structure to avoid its displacement.
Therefore, the definition of a new task (HOLD) is needed: the idea behind the HOLD task is that the robotic system must mechanically couple with the truss structure and avoid any displacement due to forces exerted by the other robot. To achieve the high stiffness required, the F/T control must be stopped (only force magnitude is monitored), so no compliant motion occurs at all.

In the Vertical_Edge_Assemble, S2 and S4 robots take respectively a node and a rod from the repositories; S4 installs the rod in the lower node on column 1, then goes to grasp (to hold it) the other end of the rod so the S2 can insert the node into it.

In the Vertical_Face_Assemble, a rod must be linked to two nodes, one for each end (for example the diagonal link): in this case, first a rod is transported and inserted into a node that is held by the other robot. Then, the other end of the rod has to be connected to the remaining node by pushing the rod (which is not completely inserted and locked) while the other robot holds the node. The same operation is repeated for the other end to be sure that the locking mechanism has triggered.
If, during the HOLD task, the node were grasped from its standard grasping interface, the node+robot subsystem would not be stiff enough because of node flexibility due to its lightweight structure. Moreover, the interaction forces generated by the partially installed rod could cause a big displacement of the node (and, as a consequence, of its grasping interface) from its expected position, leading to a not correct grasping.
In such cases the solution is to have the "holding robot" get in contact and lean itself to the node in such a way to constrain its motion in the direction along which forces generated by the other robot action are expected. The already existing PUSH action was exploited to get contact with the node, obtaining also the required position error compensation and a good contact stability between jaws and node.

Therefore, two different strategies for the HOLD task have been identified:
- HOLD "by grasping", used to hold the rod while the node is inserted into it;
- HOLD "by pushing", used to hold the node while the rod is inserted or connected.

The new CONNECT task is based on the already existent ACTUATE_BY_PUSH task, whose purpose is to generate a force able to exceed friction: as a consequence, the rod moves until the locking position is reached. Moreover, like for the holding robot, pushing instead of grasping allows a better position error recovery. The pre-condition for this task is that the rod had been previously at least partially installed in the node by a INSTALL_IN task at the end of which high accuracy of the rod position is not required because of the PUSH error recovery property.
Obviously the torsion of the structure depends on the applied force magnitude. During the insertion of the vertical rod lower end the structure below the lower node is stiff enough not to require HOLD but not enough during connection of the diagonal since in this case exerted forces are bigger.

3.2 Data preparation

As far as the repositories are concerned, no further Task/Action definition is needed, since the only required operation is REMOVE either nodes or rods so they are ready to be installed. Each of the two repositories has associated a facility frame. Set of data relevant to the position of the single objects are relative to facility frame, therefore changing of repository position implies only identification of such frame and database updating accordingly. Improvement w.r.t. the SPARCO baseline was to define even the PATH nodes w.r.t. facility frame. Nodes are taken only by S2, so this facility is not stored in S4 database.

Data relevant to the cube geometry are mainly associated to vertical face 1 and node 3 of face 3. The facility frames associated to nodes were identified w.r.t. the robot base. Subjects of the facility are mainly associated to each couple "pegs"+"rod end" (for each connected rod).

As depicted by Figure 4 and Figure 5:

- On_pose is chosen associating it to the rod fully installed;
- Off_pose is chosen associating it to the rod fully removed;
- Object frame is chosen to exploit symmetry among object frames and facility frame;
- Approach and Grip poses were defined according to rod geometry;

According to the SPARCO environment modelling philosophy, the work-cell is seen and is fully described as a set of subjects with an associated status. The database was created to reflect the status in which the cube is fully assembled and ready to be unassembled only for the sake of simplicity. Data are automatically rearranged to be coherent with any given different status (for example the initial one, in which nothing has been installed or the truss is partially built) and, after each task and action execution (including base rotation), updated to be aligned to the new layout. Therefore, frames associated to a node are not affixed to it: they describe fixed positions in the robot working space and can be associated to the node geometry only when the cube is assembled.

According to SPARCO conventions, frames associated to a subject must be defined following rules that differ depending on the task to be executed on it: for example, Z direction is positive while installing and pushing. On the couple "pegs"+"rod end" several different tasks can be performed:

1. INSTALL the rod in the pegs (Z direction is negative);
2. INSTALL the node into the rod (therefore Z direction is the opposite w.r.t the previous INTALL and the grip pose is not on the rod but on the node);
3. PUSH the rod into the node (still Z direction different);

Therefore, for some couples, more than one subject has been associated. Moreover, for nodes 1, 3, 4 of face 1, another subject, relative to the position for the HOLD "by pushing", was determined.

4. TRUSS ASSEMBLY IMPLEMENTATION

4.1 Robot co-operation: synchronisation strategy

The definition of the Truss assembling strategy must obviously take into account how the robots should co-operate: this leads to analyse and solve issues relevant to synchronisation and safety (to prevent collision between the two robots).

To exploit as much as possible the impedance control and the SPARCO modular approach it was decided that:

- Free space had to be logically divided to avoid robot collision so that a SMART S2 operation area, a SMART S4 operation area and a COMMON operation area have been defined. Whenever a robot is entering the common operation area there should be the guarantee that the other robot is in its own operation area.
- Neither of the two robot is the master of the assembly operations: each robot proceeds with its own task sequence and a co-operation is established when needed, according to the best solution for the particular case.

These two leading ideas resulted in a synchronisation method based on *rendez-vous*. Each robot independently executes its own assembling sequence whose items are compound tasks, tasks, actions and rendez-vous. The idea behind rendez-vous is that activities following it must be started simultaneously by the two robots: as a consequence, when a robot arrives to a rendez-vous, it must stop itself, waiting for that also the other robot catches up the correspondent rendez-vous. Only when both robots are at the rendez-vous they can proceed with the next operation. All rendez-vous are handled at compound task level (between two tasks) because at task level (between two actions) there is no knowledge of the system (both robots) status and avoiding collision would be almost impossible.

There are two exceptions to this rule: the HOLD and the CONNECT tasks. The HOLD task for its intrinsic nature requires synchronisation during its execution: the robot A in charge of holding reaches a position in which it is able to hold the truss for the other robot B which is waiting for the structure to be held. When robot A has reached its hold position there is a rendez-vous to unblock robot B. To leave its hold position, robot A must obtain authorisation from robot B that must have completed its task. Therefore the hold task can be divided from a logical point of view in 3 main steps each separated by a rendez-vous:

1. go to hold
2. keep holding
3. return form hold

The CONNECT task has been extended so that it can be executed, depending on the situation, in two modalities: atomic and in co-operation. In the second case, a rendez-vous is placed before the PUSH action, just before approaching to the subject to be pushed. Although this rendez-vous is executed inside the task, from a logical point of view is controlled at compound task level. It can be used to have the robots executing the PUSH action at the same time during the connection of a rod that does not require to be held or to speed up task execution.

Rendez-vous can be logically thought as transitions enabled by the presence of two tokens provided by the two robots. Each robot at rendez-vous provides its own token and waits for appearing of the one of the other robot. Then, it deletes it and waits for that its own token disappears (deleted by the other one which in this way acknowledges the rendez-vous enabling the transition). Therefore, the behaviour is totally symmetric, the synchronisation is negotiated between the two robots. Each rendez-vous has its own (sequential) numeric identifier: tokens associated to it have the same number. Transitions are enabled only if the two tokens have the same identifier (each robot checks if the token provided by the other one has same id of its own). This mechanism was implemented in a simple and reliable way: the two robot controllers share an Hard Disk located in a remote PC via serial line (Kermit protocol). The token is a file to be put or removed: the information needed (token identifier and source) is coded in the file name (for example, token number 45 provided by S2 robot is named sync045.s2).

4.2 Mounting the truss: overall sequence description

Table 5 describes in detail the compound tasks described in Table 3: Base_Rotate of steps 2, 5 and 8 that are included in Vertical_Edge_Assemble (phase 3). Rendez-vous are represented through black lines just below a pseudo-task called "rendez_vous". The HOLD task has been spliced in 5 sub-tasks (go to hold, rendez_vous, keep holding, rendez_vous and return from hold); the CONNECT in co-operation in 3 sub-tasks (Prepare to connect, rendez-vous, connect). Background colours have been used to mark distinct compound tasks (vertical bold lines group sub-tasks in the associated tasks HOLD and CONNECT).

The sequence is relevant to the assembling of a single cube: rearranging the facility frames, the same sequence is able to add on top as many cubes as robot working space allows.

Application SW has been designed so that the truss assembling can be resumed or started from any point to deal with failures or necessity to interrupt the sequence or execute only some steps of it. The information to be provided to S2 and S4 robot controller is:

1. The Compound Task sequence step ID (as in Table 4);
2. The phase ID (as in Table 5);
3. The token ID for synchronisation at the next rendez-vous;

Item 3 is the same for both robots while other can differ, provided first encountered rendez-vous (bold line in Table 5) is the same.

4.3 Truss assembling remote monitoring system

With the same mechanism used to provide tokens, robot controllers make available information relevant to the assembling execution (task being executed, messages relevant to the status of the robots). This information is collected by an application program running on the PC, which displays it through a Graphical Interface. This program acts as a server since it can also run as client on a different computer in order to provide the same messages to another user. A client application must register itself by contacting the server: at this point the server will keep sending to the client the same information it receives from the robot controllers, which displays it, until client requires to be disconnected. Communication occurs through LAN, so truss experiment can be remotely monitored from more than one computer.

5. CONCLUSIONS

The ROSED Test-Bed proved to be suitably thought and designed to successfully complete the TRUSS cube assembling but also highlighted the critical issues in this kind of activities. Impedance control, combined with ad-hoc mechanical design, proved its effectiveness to deal with uncertainties during in-contact operation. The modular structure of the control system allowed a breakdown of the TRUSS assembly in independent activities involving different competencies (impedance control tuning, mounting sequence strategy planning and many others). However, mechanical design is not sufficient to enable the impedance control to deal with positioning uncertainties cumulating during assembling of bigger structures then a cube. This suggests the need of exploiting additional sensor capabilities (e.g. vision, distance sensors) to bring the environment uncertainty within a range suitable to impedance control: these new capabilities can be easily added since allowed by the modular design of ROSED that is based on the SPARCO approach.

6. REFERENCES

[1] Putz P., Mau K. "A&R Control Development Methodology Definition Report", Doc. no. CT2/CDR/DO, Issue 2.0, Sept. 1992

[2] Surdilovic D., Specification of impedance control algorithms, ESA report SPAR-RQ1300-IPK-002, Issue 2.2, 1994

[3] Grassini F., SPARCO Executive Summary, Tecnospazio document SPAR-SD-TS-008, Rev. A, Dec. 95

NAME	MOBILITY	NOTATION	DESCRIPTION
Robot Base *B*	Fixed	--	It is the frame associated to robot base
End-point *E*	Movable	${}^{B}T_{E}(\boldsymbol{q})$	It is the frame associated to the robot flange
Tool-centre point *T*	Movable	${}^{E}T_{T}$	It is associated to the robot gripper
Facility *F*	Fixed	${}^{B}T_{F}$	It is associated to an item of the work-cell (e.g. rack)
Object *O*	Fixed	${}^{F}T_{O}$	It is associated to an item of the facility (e.g. drawer)
Work-piece *P*	Movable	${}^{O}T_{P}$	It is associated to an item of the object (e.g. drawer bottom)
Approach_pose *Appr*	Fixed	${}^{P}T_{Approach\_pose}$	Location from which the arm approaches the object to grip it
Grip_pose *Grip*	Fixed	${}^{P}T_{Grip\_pose}$	Location in which the TCP is at grasping

Table 1: SPARCO frames

OPEN/CLOSE tasks	REMOVE_FROM	INSTALL_IN
DISPLACE TO stand_by_pos (stored in the DB)	DISPLACE TO	DISPLACE TO
MOVE_ALONG path_of_nodes (stored in the DB)	MOVE_ALONG	MOVE_ALONG
MOVE_LINEAR_APPROACH (to Approach frame)	MOVE_LINEAR_APPROACH	MOVE_LINEAR_APPROACH
APPROACH (to Grip frame)	APPROACH	APPROACH
ATTACH	ATTACH	--
SLIDE (move P frame to off_pose)	EXTRACT	INSERT
DETACH	--	DETACH
RETRACT (to Approach frame)	RETRACT	RETRACT
MOVE_ALONG path_of_nodes	MOVE_ALONG	MOVE_ALONG
DISPLACE TO stand_by_pos	DISPLACE TO	DISPLACE TO

Table 2: Task breakdown

COMP. TASK	INITIAL CONDITION	FINAL CONDITION
Vertical_Edge_Assemble <n>	One rod and one node in the respective repositories.	Vertical edge assembled on column <n> (lower end of the rod connected to already installed lower node; upper end connected to new node).
Vertical_Face_assemble <n>	Two vertical edges assembled in column <n> and <n+1>; One lateral and one diagonal rod in their repository.	Vertical face assembled (3 edges and diagonal link connected to lower left and upper right vertices).
Top_Face_Assemble	four vertical faces assembled; One diagonal rod in its repository.	Top face assembled (diagonal link connected horizontally).
Base_Rotate	None.	Base is rotated of 90 degrees (former column n is now column n+1).

Table 3: Compound tasks

Step id	COMP. TASK	VE1	VE2	VE3	VE4	VF1	VF2	VF3	VF4	TF
1	Vertical_Edge_Assemble 1	X								
2	Base_Rotate		X							
3	Vertical_Edge_Assemble 1	X	X							
4	Vertical_Face_assemble 1	X	X			X				
5	Base_Rotate		X	X			X			
6	Vertical_Edge_Assemble 1	X	X	X			X			
7	Vertical_Face_assemble 1	X	X	X		X	X			
8	Base_Rotate		X	X	X		X	X		
9	Vertical_Edge_Assemble 1	X	X	X	X		X	X		
10	Vertical_Face_assemble 1	X	X	X	X	X	X	X		
11	Base_Rotate	X	X	X	X		X	X	X	
12	Vertical_Face_assemble 1	X	X	X	X	X	X	X	X	
13	Top_Face_Aseemble	X	X	X	X	X	X	X	X	X

Table 4: Cube assembling sequence

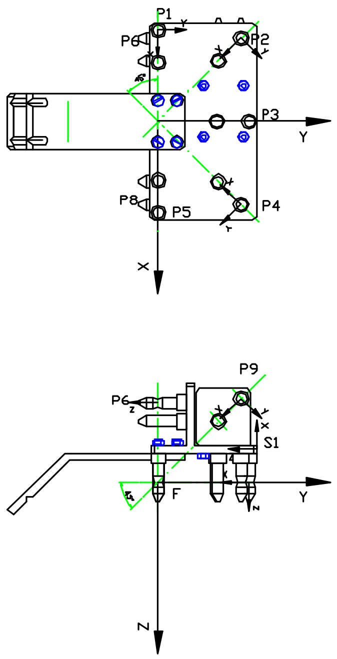

Figure 4: Node frames (1)

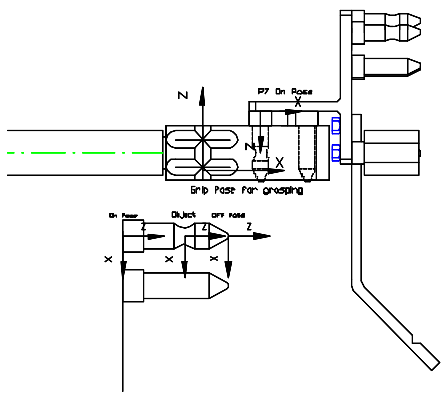

Figure 5: Node frames (2)

Compound Task	S2 phase	SMART S2 task	SMART S4 task	S4 phase
Vertical_Edge_Assemble	1	Remove node from repository	Remove rod from repository	1
	2	Rendez_vous	Rendez_vous	2
	3	rotate_base		
	4	Rendez_vous	Rendez_vous	3
			Install rod in lower left node	4
			Go to hold left rod by grasping upper end	5
	5	Rendez_vous	Rendez_vous	
	6	Install node in left rod upper end	Keep holding	
	7	Rendez_vous	Rendez_vous	
			Return from hold	
Vertical_Face_Assemble	1	Rendez_vous	Rendez_vous	1
	2	Remove rod from repository	Go to hold by pushing upper right node	2
	3	Rendez_vous	Rendez_vous	
	4	Install horizontal rod in upper right node	Keep holding	
	5	Rendez_vous	Rendez_vous	
	6	Go to hold lower left node	Return from hold	
		Rendez_vous	Rendez_vous	3
		Keep holding	Connect upper left node/horizontal rod	4
		Rendez_vous	Rendez_vous	5
		Return from hold	Go to hold by pushing upper right node	6
	7	Rendez_vous	Rendez_vous	
	8	Connect upper right node/horizontal rod	Keep holding	
	9	Remove diagonal rod from repository		
	10	Install diagonal rod in upper right node		
	11	Rendez_vous	Rendez_vous	
	12	Go to hold by pushing lower left node	Return from hold	
		Rendez_vous	Rendez_vous	7
		Keep holding	Connect lower left node/diagonal rod	8
		Rendez_vous	Rendez_vous	9
		Return from hold	Go to hold by pushing upper right node	10
	13	Rendez_vous	Rendez_vous	
	14	Connect upper right node/diagonal rod	Keep holding	
	15	Rendez_vous	Rendez_vous	
			Return from hold	
Base_Rotate	1	Rendez_vous	Rendez_vous	1
	2	rotate_base		
	3	Rendez_vous	Rendez_vous	2
Top _Face_Assemble	1	Rendez_vous	Rendez_vous	1
	2	Remove diagonal rod from repository		
	3	Install diagonal rod in back left node		
	4	Rendez_vous	Rendez_vous	2
	5	Prepare Connect back left node/diagonal rod	Prepare Connect upper right node/diagonal rod	3
		Rendez_vous	Rendez_vous	
		Connect back left node/diagonal rod	Connect upper right node/diagonal rod	

Legenda:

lower left node	Node: 1	Face: 1
lower right node	Node: 2	Face: 1
upper right node	Node: 3	Face: 1
upper left node	Node: 4	Face: 1
back left node	Node: 3	Face: 3

Table 5: Compound Task detail