Proceeding of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS 2001, Canadian Space Agency, St-Hubert, Quebec, Canada, June 18-22, 2001.

EUROPA Robot Controller

A. Rusconi (Tecnospazio, Milan, Italy) e-mail: arusconi@tecnospazio.it

R. Finotello (Tecnomare, Venice, Italy) e-mail: finotello.r@tecnomare.it

T. Grasso (Tecnomare, Venice, Italy) e-mail: grasso.t@tecnomare.it

R. Mugnuolo (Agenzia Spaziale Italiana, Matera, Italy) e-mail: raffaele.mugnuolo@asi.it

A. Olivieri (Agenzia Spaziale Italiana, Matera, Italy) e-mail angelo.olivieri@asi.it

Keywords space robot controller, force control, redundancy, science payload, manipulation, safety barriers

Abstract

This paper describes the main characteristics of the EUROPA robot controller hardware and software.
EUROPA (External Use of Robotics for Payloads Automation) is a robotic experiment for externally exposed payloads, which will be installed on an EXPRESS Pallet Adapter on the S3 truss of the International Space Station.

1. Introduction

For more than 10 years, extensive research and development in the area of space robotics have been performed in the European community. Amongst the different application scenarios, the dextrous manipulation of Space Station external payloads using a medium sized (1.5 - 2 m long) robot has been studied in depth. All the subsystem technology has already been created. In particular, in the frame of the national Italian programme named SPIDER (Space Inspection Device for Extravehicular Repair) a 7 axis robot system and an end effector have been developed.
To demonstrate the potential for in orbit use of robotics for external applications, ASI assumed the initiative to find a new flight opportunity in the framework of ASI/NASA co-operation for the International Space Station development. After ASI proposal, NASA accepted, scheduled and manifested EUROPA experiment on the EXPRESS Pallet platform on the International Space Station (ISS).
EUROPA is intended to perform a realistic end-to-end robotic technology demonstration to show the advantages and the feasibility of a versatile robotically tended exposed payload infrastructure.
The EUROPA design allows performance of the following tasks:

- installation/removal of small payload containers on exposure attachment ports;
- handling of payload units (experiment samples or sample cassettes) for the purpose of scientific/ technological investigations;
- close-up visual inspection of payload units by means of a camera.

All of the above tasks can be high-level pre-programmed and checked on ground and then performed automatically on orbit, with ground monitoring and possibility to intervene and correct the situation in case anomalies are detected.
The EUROPA robot controller will be an externally exposed unit (as part of the EUROPA flight segment items mounted on the EXPRESS Pallet), fully dedicated to controlling the robot arm, to provide the necessary computing power to include sophisticated control algorithms and reach a high degree of autonomy for the on-board operations.
The EUROPA robot controller provides the following features:

- execution of all the robotic programs written using the Control Development Methodology (CDM, see [1]) with definition of tasks and actions;
- control of a robot arm with seven joints and an end effector (a gripper with two parallel jaws), with real-time management of the arm kinematics redundancy;
- motion control capabilities in free space, using internal sensors (resolvers);
- contact motion control capabilities (using force/torque and tactile sensors) with the possibility to define operation completion either by force/torque threshold or by relative position between the end effector and the grapple fixture interface;
- implementation of a computer based control system fail safe approach, such that the overall EUROPA system will be two-fail safe for specific hazards related to the robot manipulator, which is a moving object (arm outside of defined workspace and collision effects);
- achievement of a good degree of failure tolerance, in order to guarantee safe stowage of the robot arm in front of a single failure in the nominal part of the controller, by providing redundant hardware (emergency control using redundant power lines, redundant CPU board and using cross-strapping features of the motor drivers). This will minimise

astronaut Extra Vehicular Activity (EVA) intervention;
- modular software architecture based on hierarchical NASREM concept, structured in Task Level and Action Level (for task and action decomposition) and Primitive Level and Servo Level (for trajectory definition and servo control).

This paper is organised as follows.
Section 2 contains a general description of the EUROPA system.
Section 3 is dedicated to the EUROPA robot controller. First a description of its main features and capabilities is given, then the safety and contingency aspects are covered, finally the hardware and software architecture is described.
Section 4 contains the conclusion.

2. EUROPA System Configuration

An overall view of the EUROPA system is given in fig. 1.

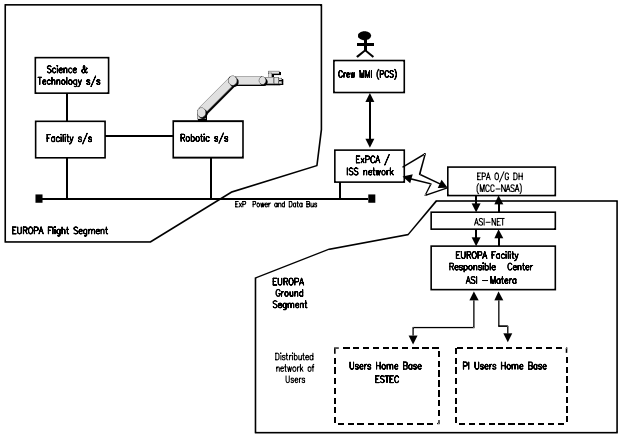

Figure 1: EUROPA System Overview

The EUROPA system is composed of a flight segment and a ground segment.

2.1 EUROPA Flight Segment

The flight segment of EUROPA (see [4]) is the part aimed at the execution of the robotic capability demonstration and the payload handling.
It will be accommodated on the EXPRESS Pallet Adapter (ExPA) (fig.2).
The EUROPA flight segment is divided into four subsystems:
- Robotic subsystem
- Facility subsystem
- Science and technology subsystem
- Crew MMI subsystem.

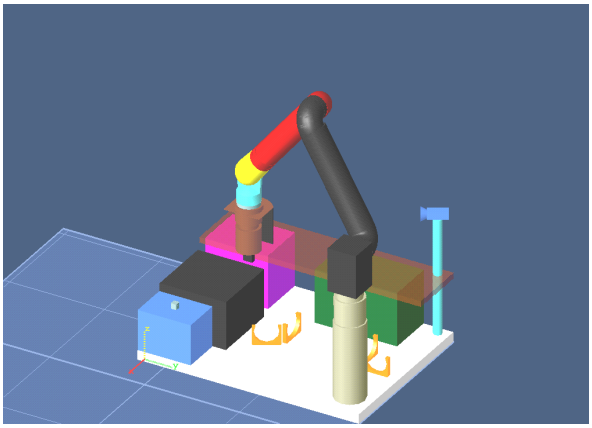

Figure 2: EUROPA Flight Segment on the ExPA

Robotic s/s
The robotic subsystem is composed of the following assemblies:
- hold down assembly – automatic mechanism to latch the arm during launch and re-entry and landing phases;
- arm assembly - to perform, when integrated with its controller, all the required manipulation activities;
- force/torque sensor (F/T) assembly - to allow force and torque control during the execution of operations;
- end effector (EE) assembly - to grasp the items to be handled, and to provide tactile sensing capabilities;
- robot calibration platform (RCP) assembly - to acquire linear distance and video camera image data, to be used for calibration of the robot and the workcell;
- arm astronaut aids - to help the astronauts during EVA operations to recover the EUROPA flight segment when specific failures occur;
- controller assembly - to control the operating sequence of the arm and end effector assemblies in a pre-programmed automatic way.

Facility s/s
The facility subsystem is composed of the following assemblies:
- data handling and power unit (DHPU) - to allow EUROPA mission data handling and storage, and to distribute power from ExPA power buses to EUROPA items;
- thermal control assembly - to provide the thermal control elements for EUROPA items;
- facility subsystem astronaut aids (e.g. foots restraint) - to help the astronauts during EVA operations to recover the EUROPA flight segment when specific failures occur;
- harness - to electrically connect all external EUROPA flight segment items;

- video monitoring unit (VMU) - to allow lighting and panoramic view of the workspace.

Science and technology s/s
The science and technology subsystem is composed of the following assemblies:

- The science payload assembly proposed for EUROPA from ESA side, an experiment which aims at measuring diffusion and Soret coefficients of liquid mixtures. The payload will benefit the basic manipulation capabilities (sample transfer) from EUROPA manipulator.
- The taskboard assembly which provides the required in-orbit infrastructure to evaluate/measure the performance capabilities of the arm. The taskboard is composed of a force and torque checkout unit and a laser diode unit.

Crew MMI s/s
The crew MMI is a command and monitoring station which displays the EUROPA manipulator status, and enables the crew to perform local commanding of the EUROPA flight segment.

2.2 EUROPA Ground Segment

The ground segment of EUROPA is the part aimed at:

- ground calibration;
- program preparation and verification;
- flight robot monitoring and command;
- on-ground data handling.

Among other items (see [5]), the EUROPA ground segment includes the Ground Reference Model (GRM), which is a ground replica of the flight segment and will be used during preparation and verification activities.

2.3 EUROPA Operations

Preparation
All EUROPA on-orbit operations will be pre-programmed and verified on ground using an off-line programming system and the GRM.
The resulting activity plan will be a set of hierarchical activities consisting of tasks and actions, which will be uploaded on the flight robot controller.

Execution
The execution of the activity plan will be possible either under ground control (access to the TM/TC channels is assumed to be available) or under on-board MMI control (crew PC), either in autonomous or interactive mode.
In case of anomalies, the system will provide the possibility to intervene and correct ("interactive autonomy"), through immediate stop of the current execution at any time.
After the stop, the user can resolve the anomaly and resume the experiment or perform the arm stowage (nominal or non-nominal).

3. EUROPA Robot Controller

3.1 Controller Features

EUROPA robot controller is based on state-of-the-art robotic technologies. Concepts such as active position-force control, resolution of redundancy of degrees of freedom and resolved motion are fully exploited.
The interaction of the EUROPA Operator with the flight controller is based on the preparation and the verification of the robot operations using the ground reference model for emulation.
Robot motion and its interaction with the environment is specified using a high level programming language. It allows the operator to easily perform all the operations by editing and verifying simple robotic programs and uploading them to the controller for execution on-board.
In the following paragraphs details of the main controller capabilities will be given.

Command interpreter and generation
Robotic mission preparation and verification is performed on ground using an exact replica of the workcell (the GRM) and other tools for a correct motion planning. At the end of this phase the operator is required to code his work in the form of a high level program file, to be sent to the space robot controller when required. The controller interpreter shall be compliant with the I/F to the EUROPA MMI Ground Station based the SPARCO (SPace Robot Controller, see [2]) libraries concept; in particular all the robotic programs will be written using CDM definition of tasks and actions.
The main operations the robot shall perform are:

- Cartesian and joint motion, in case of arm deployment and fine motion toward the science and technology subsystem
- approach to objects of the surrounding environment, according to specific strategies related to the objects themselves
- object grasping, as in the case of transfer of scientific sample container
- interaction with the environment, as in the case of drawer opening/closing, operations on switches.

These operations shall be conveniently translated into tasks-actions and control flow instructions in such a way that the operator is not required to have particular programming skills to prepare the robotic program file.
The robot controller shall be able to interpret these instructions and execute the relevant commands.

According to CDM, the commands can be divided into tasks, actions and control flow. The task represents the most complex activity a subject is able to perform and it is composed by a suitable sequence of action commands and control flow. An Action represents an ability of the controlled robot at the highest level, it is directly implemented inside the controller. The Operator customizes the characteristics of the commanded action by modifying its parameters. A task is implemented in a program file composed of action commands and control flow commands and customized by suitable input parameters.
Some examples of task commands are:

- *OPEN/CLOSE* a subject , for instance a drawer or a door
- *CONNECT/DISCONNECT*, a subject, for instance a connector
- *DEPLOY/STOW* the robot arm
- *INSTALL/REMOVE* a subject on a specified target
- *INSPECT* a given target

Some examples of action commands are:

- *ACTIVATE* device WITH strategy UNTIL event
- *APPROACH* target TO pose
- *ATTACH* subject WITH strategy
- *DISPLACE TO* pose
- *INSERT* subject INTO target WITH strategy
- *FOLLOW* path ON target WITH strategy
- *LATCH* subject ON target WITH strategy
- *MOVE TO* pose ALONG path
- *PUSH ALONG* path
- *MOVE_IN_CONTACT* ON target ALONG path WITH strategy
- *MOVE_TO_CONTACT* WITH strategy

An example of how a task is built using actions is a possible implementation of OPEN:

```
OPEN:=  [drawer]
        DISPLACE TO pose1
        MEASURE subject.interface
        EVALUATE
        APPROACH
        ATTACH
        PUSH ALONG subject.open_path
        DETACH
        RETRACT TO pose1
        DISPLACE TO standby
```

An important concept is that all the useful data for a given operation is stored in a database compliant with SPARCO library. In other words, database is “object-oriented”, in the sense that only the specific objects of the environments know how to be operated. For example approach path, grasp force, force controlled directions, mechanical interface position are items stored on the object itself. This means that the interpreter shall query the database each time an action is requested in order to correctly build lower level commands to the controller.
Taking into account the type of motion and the relevant constraints, the following set of manipulation primitives will be implemented in the controller:

- transfer motion (gross motion in free space),
- no contact move (fine motion relative to the environment),
- move to/from contact with the environment,
- move in contact with the environment (move while a specified contact with the environment is maintained)
- grasp/release of an object, firm or compliant (if the object to be grasped sets motion constraints to gripper when closed between the jaws)

Each actions is mapped to a motion primitive, whose conceptual definition is based on the:

- type of motion (in free space or compliant to the environment);
- motion constraints on the end effector deriving from the desired characteristics of the contact with environment or the trajectory characteristics along not constrained directions;
- end effector target pose or, if the target pose is not a-priori known, the primitive termination conditions.

Using the hybrid position/force control, with respect to the task frame, a generic position/force primitive is produced by associating a suitable "trajectory generator" to each of the six Cartesian degrees of freedom of the end effector (three translations and three rotations).

Redundancy resolution

The SPIDER arm, used for EUROPA operations, has 7 rotational degrees of freedom. A brief sketch of the arm is given in Fig. 3.

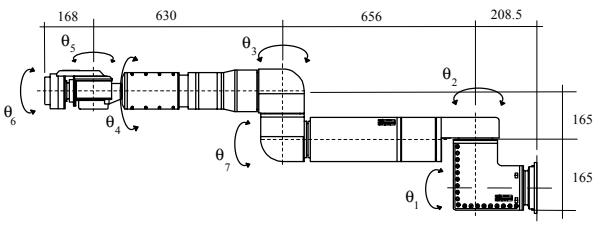

Figure 3: Arm Architecture

Since the typical tasks involve the control of the location of the robot tool, it derives that the manipulator is redundant for those tasks. The redundancy degree is 1 and, inside the robot workspace, each tool location corresponds to ∞^1 joint configurations belonging to the jacobian null space in the current configuration (the

motion along the jacobian null space is also called *self motion*). Therefore it is necessary to choose among these configurations each time a tool setpoint is generated, while assuring joint trajectory continuity. The most common methods to solve redundancy integrate motion equations taking into account additional constraints or criteria to be optimized, and starting from the current robot configuration as initial condition. The typical equation that solves the inverse kinematics problem is

$$\Delta q = J^{+}(\Delta x + Ne) + k(\mathrm{I} + J^{+}J)\nabla g \qquad \textbf{(eq. 1)}$$

where q is the current joint configuration, x is the desired Cartesian increment, e is the Cartesian error, J is the jacobian, $g(\cdot)$ is the criterium to be optimized, N and k are suitable gains.

The inverse kinematics algorithm is local. This means that if a specific tool trajectory is started from two close initial configurations, nothing is said about the final configurations, which could be far one another. Further the eq. 1 suffers from drift problems, in the sense that closed Cartesian trajectories in general do not correspond to closed joint space trajectories. The drift on joint trajectory could either be asymptotic to a specific configuration, or diverge until a joint limit is encountered. This fact could be a problem if robot motion were automatic or teleoperated and not fully tested on ground. Suppose for instance to automatically rotate a wheel by some turns; the operator expects that at the end of the motion the robot returns to its original position. On the contrary, if robot trajectories are fully tested on ground, drift could be acceptable since it has been verified. In this case it is necessary to have the same inverse kinematics algorithm on both ground and space, and start from the same initial configuration. In other words, use the same system configurations.

In the case of EUROPA controller, this is true until force control is used (whose description is given in the next paragraph). In fact the inaccuracies of the environment geometric model, force sensor response/noise, force control strategies and friction properties of the contact surfaces could drive the robot arm in not pre-planned configurations, while repositioning on the tool locations verified on ground. This problem yields not only for closed trajectories, but also for open ones (i.e. the final joint configuration does not coincide with the ground pre-planned one), and position trajectories subsequent to force/position motion could start from not pre-planned initial configurations. To cope with this problem it is necessary to fix, in some manner, the degree of redundancy. One possibility is to generate joint trajectory using a drift-free algorithm, and then to pass this trajectory to the higher speed force loops in order to suitably control the joint motion in the jacobian null space. The algorithm to be used could be based on the extended jacobian:

$$\Delta q = \begin{bmatrix} J \\ J_{add} \end{bmatrix}^{-1} \begin{bmatrix} \Delta x \\ \Delta x_{add} \end{bmatrix} = J_{ext}^{-1} \Delta x_{ext} \qquad \textbf{(eq. 2)}$$

where the subscript 'add' means additional and 'ext' means extended. The additional constraint (which forms the additional jacobian) is imposed to complete the set of equations. It can be used to control some robot geometric parameters (in the simplest case to fix one joint angle to a predetermined value), or optimize some criteria if the additional constraint is suitably chosen.

Force control

If the SPIDER manipulator has to actively interact with the environment, then it is necessary to equip it with force control. Various strategies are possible. For EUROPA controller the active force/position control in the operational space has been chosen. It relies on the assumption that it is possible to decompose the Cartesian space into *m* directions controlled in position and *6-m* controlled in force, where *m* goes from 0 to 6. Along the force-controlled directions, only desired forces (moments) are considered, and resulting position is not of interest. On the other hand, along the position-controlled directions only desired positions (orientations) are considered, and resulting force is not of interest. These directions are mutually orthogonal, even if it is possible to generalize the contact model to non-orthogonal directions. In order to act to the correct directions, suitable *selection matrices* are prepared; they 'filter' the signals generated by the force and position controller parts in such a way that they do not influence one another. Force/Position controller parts are typically PID's, and their outputs are Cartesian forces, to be reported to joint space in the form of joint torque values, that will move the robot motors. A principle sketch of the position/force control scheme is given in Fig. 4, where M is the selection matrix (built using the direction flags specifying which directions are position-controlled, and which force-controlled), and task frame is the reference frame used for position and force description. Usually this task frame is the tool frame, but sometimes it is useful to consider other frames to better achieve the current task.

Note also that the robot input is the set of "desired" joint torques. This means that some effort has to be done to accomplish this requirement, also taking into account that the SPIDER arm has no inner torque loops at the joints.

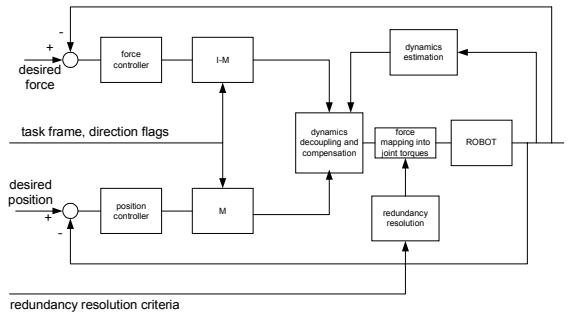

Figure 4: position/force control scheme

As said before, an important point is related to redundancy resolution. This is addressed by imposing specific joint torques, deriving for instance from higher level planning, along the self motion. The projector for the torques on this self motion will be dynamically consistent, and computed using dynamics information (i.e. inertia matrix). Due to the strong computational effort required, the dynamics estimation will be done at low rate, possibly interpolating the relevant data in the middle. In this case it is important to acknowledge that if precise dynamics data is missing, self motion contribution could affect Cartesian motion of the tool.

3.2 Safety and Contingency Aspects

Besides standard safety hazards (structural failures, contamination, etc.) which are applicable for an externally exposed payload, there are some specific hazards which are applicable to a robotic system.

Collision effects and workspace limitation

When the arm moves, there is a danger for collision with surrounding objects and for protrusion outside the assigned workspace. According to ISS requirements, each ExP payload must remain within an envelope of 864 x 1168 x 1245 mm). Protrusions outside the payload envelope could cause collisions with adjacent ExP payloads or field-of-view obstructions. Any exceedances to the ExP payload envelope make a payload non-standard and will have to be approved by the ISS Program. Payload envelope exceedances will be pre-planned operations, and neighbouring ExP payloads will be informed about any permanent or intermittent payload envelope exceedances. Therefore, it is necessary to guarantee that EUROPA will perform all of its operations inside the assigned envelope (standard or non-standard).

From the above considerations, the following hazards can be derived:

- Collision Effects" and "Collision-Caused Payload Parts Release" (to be controlled in both speed and current in order to reduce the impact - the impulse - of a non-nominal collision or excessive force/torque interactions during contact operations),
- "Arm Parts outside of defined Workspace (to be controlled in position)".

In order to control these hazards, architectural requirements are derived at all levels of the EUROPA system, and some specific requirements are derived at controller assembly level. For the above identified hazards, a two-fail-safe computer based control system will be implemented against each one of the hazard causes (position errors, speed errors, current errors).

The following requirements are identified:

- The arm workspace must be limited to prevent collision, by forbidding the arm to go outside envelope limits. The limitation will be double fault tolerant;
- The speed of the arm must be kept below predefined thresholds. The limitation will be double fault tolerant;
- The arm pushing force must be kept below given thresholds. The limitation will be double fault tolerant;
- After a potentially safety-critical fault is detected, the arm must be brought in safe status;
- The brakes must be fail-safe (i.e. when power is not applied, the arm is braked).

The system preliminary architecture has been derived taking also into account the above requirements and, as we will see in the following sections, specific features are considered for the controller (involving both hardware and software):

- Two safety devices (beyond the nominal control) ought to detect current limit overcross, to limit arm pushing force; this check will be mainly performed by hardware (with the current thresholds settable and readable by software);
- Two safety means (beyond the nominal control) ought to detect speed limit overcrossing, to limit speed of the arm;
- The position attitude of all joints with respect to the allowed workspace ought to be controlled (beyond the nominal control), based on measures performed by the output shaft resolvers, fine signal and coarse signal, by two different devices;
- The controller/safety computers (boards) involved in arm motion/safety barriers ought to have independent clocks, independent processors and independent memories. Indeed, the speed and position checks will be performed by independent computer boards with resident software (resolver acquisition boards provided with CPU).

Arm stowage before re-entry

There is another hazard associated to the fact that in front of a single failure that causes the interruption of

the experiment, it is in general not possible to guarantee that the robotic arm can return automatically in stowed position. The hazard is:

- Impossible Arm Stowing (for re-entry).

This requires to foresee, in some cases, an astronaut EVA intervention. It is necessary to do an effort in order to minimise the probability and the duration of such an intervention. The following means can be envisaged:

- Automatic re-stowing of the arm after the failure;
- Quick re-stowing of the unbraked arm by EVA;
- Re-stowing by de-coupling the manipulator output shafts from the motor shafts by EVA;
- Physical dismounting of the arm by EVA.

At least one of these interventions shall guarantee the safe re-entry of the ExP when scheduled.

From the controller point of view, it will be necessary to provide some degree of failure tolerance, involving both hardware and software measures, to perform automatic stowage in most possible cases, thus avoiding EVA intervention. As we will see in the following section, the controller will be provided with redundant hardware (emergency controller) which intervenes when there is a failure on the nominal part (nominal controller).

Depending on the failure, the automatic re-stowing can be supported in the following ways.

If, after the failure, the nominal controller is still operational, a particular management of the commanded motion can be implemented. The nominal trajectory is planned to deal with possible non-nominal poses that might lead to interruption of the execution:

- a set of "safety key-poses" are defined and managed for each trajectory;
- each "safety key-pose" is associated with a dedicated arm stowage sequence.

If, after the failure, the nominal controller is no longer operational, the emergency controller can be used, in order to guarantee safe stowage of the robot arm.

3.3 Controller Hardware Architecture

The controller hardware architecture is shown in Fig. 5.
The controller is composed of the following three main parts:

- Nominal controller;
- Emergency controller;
- Motor Drivers.

The first one provides the robotic arm control in nominal mode, the second one is activated only in emergency mode to perform the arm stowing. Both controllers share the motor drivers block.

In order to guarantee the single failure tolerance, each motor driver output can be connected to two motors by a dedicated switching network (cross strapping) in order to allow the sequential operation of a single joint at a time even in failure condition.

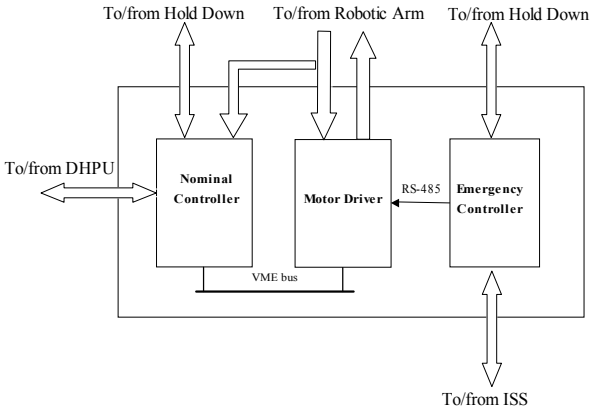

Figure 5: Controller Hardware Architecture

Nominal Controller

The nominal controller provides the following features:

- Nominal motion control, to achieve coordinated motion of the end effector in free space and in contact with the environment, by using internal sensors such as resolvers and external sensors on the end effector, such as force/torque and tactile sensors;
- Capabilities to acquire the hold down sensors and to actuate the hold down mechanism, to latch/unlatch the arm in its stowage configuration;

The nominal controller consists of the following boards, which are interfaced via the VME standard bus:

- Robot Control board, a CPU Module based on PowerPC (supporting serial lines, memory and VME controller interface)
- Output resolver board (fine), a resolver acquisition board, which includes a microprocessor for safety (for position and speed check based on fine resolvers on output shaft)
- Sensor I/F board, to interface the force/torque sensor
- General I/O board, for analog/digital I/O
- Motherboard, to provide electrical connection for the VME boards that constitute the controller.

The nominal controller includes also the nominal power supply, which receives power from the DHPU via a 28V regulated power bus.
Two RS422 serial lines (nominal and redundant) from/to the DHPU provide the data link for telecommands and telemetry.

Emergency Controller
In case there is one failure on the nominal chain (nominal controller and/or data handling and power unit) that cannot be recovered, the motor drivers block shall be isolated from the VME bus and the arm motion control (stowage function) will be performed by the emergency controller. The possibility of performing the automatic stowage for a wide range of failures on the nominal chain will minimise astronaut EVA intervention.

The emergency controller provides the following features:

- It is directly connected with the ExPA + 28 Vdc power line to distribute power to motor drivers and hold down
- It is directly connected to the ExPA 1553 line for communication
- It transmits commands to the driver via RS-485 serial line allowing direct drive control of the arm joints (motion of a single joint at a time with small steps)
- It is able to command the use of cross-strapping features of the drivers, if needed
- It controls the hold down mechanism, to latch/unlatch the arm in its stowage configuration.

The emergency controller consists of the following boards:

- Emergency Control board, a CPU Module based on the Standard Payload Computer (SPLC, see [3]), supporting serial lines and memory, including a MIL-1553B interface mezzanine and a digital I/O mezzanine;

The emergency controller includes the emergency power supply, which receives power from the ISS via a 28V power bus.
A MIL-1553B serial line from/to the ISS provides the data link for the arm control in emergency mode.

Motor Drivers
This part provides the driving power for the robotic arm/end effector motors and brakes, by providing four boards (each one dedicated to two joints).
This part is shared between the nominal and the emergency controller (it can be connected to both controllers)
This part provides the following features:

- receive motor current setpoints from the controller
- perform motor resolver acquisition
- provide motor commutation function, by generating two-phase current reference vectors
- perform current loop and provide power drive to the motors.

In nominal mode, commands will be received from the nominal controller via VME bus.
In emergency mode:

- commands will be received from the emergency controller via serial line;
- it is foreseen to drive one joint motor at a time;
- in case of failure on one driver board, a given joint can be moved by another board, by means of cross-strapping circuitry.

The motor drivers block is composed of the following boards:

- Driver boards, four boards providing the power to the motor and resolver acquisition for motor commutation;
- Cross Strap board, a front plane providing the cross strapping of the motor drivers;
- Output resolver board (coarse), a resolver acquisition board, which includes a microprocessor for safety (for position and speed check based on coarse resolvers on output shaft).

3.4 Controller Software Architecture

EUROPA controller software architecture will be based on hierarchical NASREM concept, structured in Task Level and Action Level (for task and action decomposition) and Primitive Level and Servo Level (for trajectory definition and servo control). This hierarchical decomposition of the system functions is supported by a horizontal decomposition describing the typology of the functions for a given level. There will be *Task Decomposition*, in which robot motion is decomposed according to the hierarchical level, *Sensory Processing* where all the sensed data are suitably processed, and *World Modeling* that provides suitable world reconstruction and modeling algorithms. The NASREM architecture is reported in Fig. 6.
Not shown in the figure for simplicity there are the *Global Memory*, that is the entire database of the system, and the *Operator Interface*. These two modules can communicate with each module of the figure.
This architecture allows easy introduction of not foreseen devices such as tele-operation devices to command speed setpoint according to a given reference frame, and exteroceptive sensors, (e.g. vision system, proximity sensor), to perform visual trajectory control or constrained motion.

Typical functions in the Servo Level may be:

- Actuator/Gripper position/force loops (H1)
- Active Force/Position loop (H1)
- Fine setpoint interpolation (H1)
- Actuator command generation (H1)
- Forward kinematics, jacobians (M1)

- World-Device database interpreter, and data consistency check (M1)
- Actuator commands and sensor data prediction (M1)
- Sensor filtering/integration (G1)
- Virtual sensors (G1)
- Sensors calibration and offsetting (G1).

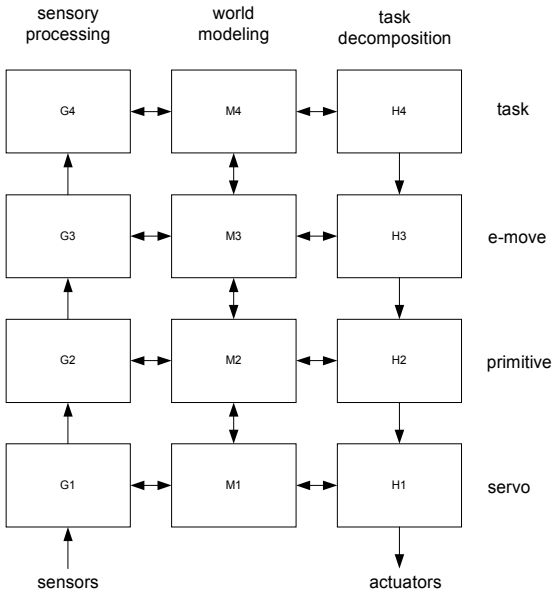

Figure 6: NASREM architecture

Typical functions in the Primitive level may be:
- Inverse kinematics (H2)
- Redundancy resolution (H2)
- Inverse dynamics at joint and cartesian level (H2)
- Gross via-points interpolation (H2)
- Actuator setpoint generation (H2)
- World-Device database interpreter, and data consistency check (M2)
- Object reference frame computation (M2)
- Redundancy resolution strategies management (M2)
- Forward kinematics and dynamics, jacobians (M2)
- Estimation of arm joint position and speed (G2)
- Fusion between arm sensors and proximity sensors (G2).

Typical functions in the E-Move level may be:
- E-Move command interpreter (H3)
- Via-Points generation (H3)
- Choice of control strategies between via-points (H3)
- Grasp point evaluation (H3)
- Computation of robot-world distances (M3)
- Setup for object reference frame computation (M3)
- World-Device database interpreter, and data consistency check (M3)
- Arm calibration (M3)
- Estimation of "e-move accomplished" (M3)

Typical functions in the Task level may be:
- Task command interpreter (H4)
- E-Move list generation (H4)
- Estimation of "task accomplished" (M4)
- Arm base frame calibration (M4).

4. Conclusion

The EUROPA robot controller will provide a hardware and software platform, fully dedicated to on-board control of the EUROPA robot arm, based on state-of-the-art robotic technologies. It will allow to implement sophisticated control algorithms and reach a high degree of autonomy for the on-board operations.
In addition, it will make use of a computer based control system fail safe approach, which provides all the necessary features to ensure the safety of the robotic operations while it still preserves their flexibility and does not prevent manipulator dexterity (workspace limits are not fixed by hard stops or mechanical limit switches, but can be changed depending on the application, allowing to potentially operate on other ISS payloads).

References

[1] P. Putz and K. D. Mau, *A&R Control Development Methodology Definition Report*, Dornier, doc. nr. CT2/CDR/DO Issue 2, 1992.

[2] P. Putz, "SPARCO: an Advanced Space Robot Controller", *Preparing for the Future* Vol. 5 No 4, Dec. 1995.

[3] C. Taylor, H. König and U. Schloßstein, "Standard Payload Computer for the International Space Station", *ESA Bulletin 93*, February 1998.

[4] A. Rusconi, R. Mugnuolo, F. Bracciaferri, A. Olivieri, F. Didot, "EUROPA (External Use of Robotics for Payload Automation)", in *Proceedings of 6th ESA Workshop on Advanced Space Technologies for Robotics and Automation - ASTRA 2000* (ESTEC, Noordwijk, the Netherlands), December 5-7 2000.

[5] R. Leone, S. Losito, R. Mugnuolo, F. Pasquali, F. Didot, M. Favaretto, R. Finotello, A. Terribile, D. Galardini, "The EUROPA Ground Segment", in *Proceedings of 6th ESA Workshop on Advanced Space Technologies for Robotics and Automation - ASTRA 2000* (ESTEC, Noordwijk, the Netherlands), December 5-7 2000.