PROFIBUS PC BASED MOTION CONTROL WITH APPLICATION TO A NEW 5 AXES PARALLEL KINEMATICS

Patric Pham, Dr Mohamed Bouri, Markus Thurneysen, Prof. Reymond Clavel

Laboratoire de Systèmes Robotiques (LSRO)
Swiss Federal Institute of Technology
1015 Lausanne, Switzerland

ABSTRACT

Generally the classical control architecture used to carry out a robot position control is based on a classical industrial motion planner that needs both encoder and analog cabling to insure the regulation loop. This work proposes a PROFIBUS PC based control architecture that has a double originality. The first originality concerns the hardware architecture and the second concerns the software flexibility that allows easiness on implementation and modification for controlling parallel robots.
This work will also focus on a totally new parallel kinematics possessing 5 degrees of freedom called Alpha5. It has the exceptional ability to achieve rotations of $\pm$ 90° in amplitude on both, the A and B axis. The present kinematics shows an important complexity in its mechanics and is therefore a more difficult structure to control. The present work has proven the feasibility of this PROFIBUS PC based control even for such a complex mechanical structure. Figure 1 shows the prototype of the Alpha5 robot.

Keywords: Field bus, PROFIBUS, motion control, PC based control, real time PC, parallel robotics, Alpha5, Delta

1. INTRODUCTION

In today's factories thousands of actuators and sensors are working, serving to guarantee a steady flow of production. These devices mostly need a transfer of information with a central unit that looks after the control and the information processing. This information transfer is made by using a physical support, mostly a copper wire or optical fiber. By the increasing number of devices in the factory the complexity of the whole connectivity grows enormous. Preventing this complexity is the point where the fieldbus outstands.
The fieldbus is a totally digital system for exchanging information between devices. Its manner of interconnecting the devices in the field considerably reduces the complexity of wiring and therefore offers a great transparency at the installation and use. All the stations of the fieldbus-system are connected in series, using the same physical support to connect one station to another. The figure 10 shows both architectures.

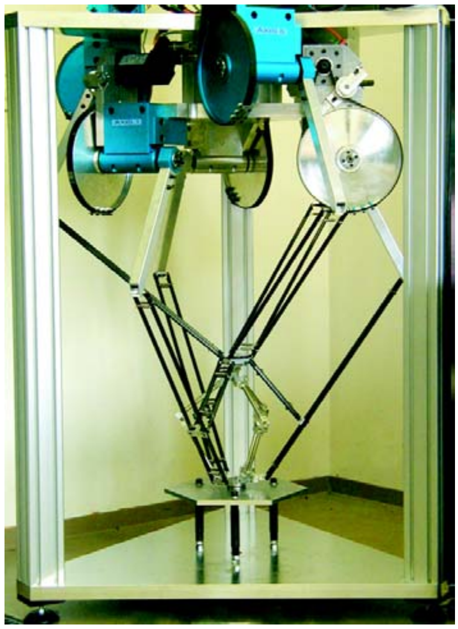
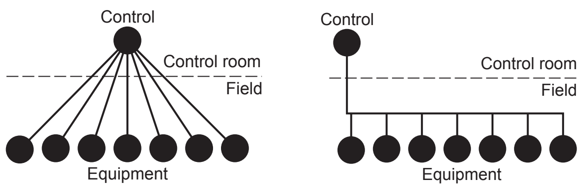


Figure 1: *The Alpha5 Prototype*



(a) *Classical Centralized Architecture*

(b) *Fieldbus Architecture*

Figure 2: *Both communication systems on field level*

Another possible directive of the fieldbus is to decentralize the regulation loop to the individual devices. Devices which are ensuring their own regulation loop are called "intelligent". Both velocity and position control loops may be available. The individual devices get their set-point through the bus and regulate it independently without necessarily closing the regulation loop through the bus. This aspect can considerably diminish the computing power needed on the central control unit. In most cases this set-point is the final needed position. The intermediate positions for each axis are computed by taking into account configured local acceleration and velocity profiles. This is not usable for robots for which the axes positions must be synchronized through a geometric model depending on the robot kinematics. To take profit from the easiness of the fieldbus cabling, this paper proposes to close the loop via the fieldbus. Two control loops have been used. The first loop is the velocity control loop provided by all the fieldbus devices. The second one is the position control loop implemented on the PC and is cascaded with the velocity control loop. The fieldbus that has been chosen for our application, namely PROFIBUS, will be shortly described in section 2.3.

2. FIELDBUSES FOR MOTION CONTROL

2.1. Review on a Classical Analog Robot Control

A classical robot control architecture is shown in figure 3. It consists of a central intelligent unit (namely a PC, or any DSP based processor board) with an axis interface board and amplifiers. These amplifiers are generally more complicated and also embed a velocity control loop. They are called servo drives. The role of the axis interface board is reading the encoder impulses and setting the analog values of the amplifiers. The regulation loop is closed via the intelligent unit. This regulation loop can only refresh one value at a time: the desired velocity or torque by setting the analog input of the servo drives (amplifiers). No additional information can be sent because of the analog communication (through +/-10V range). Hence no other feed-forward information can be taken into account.

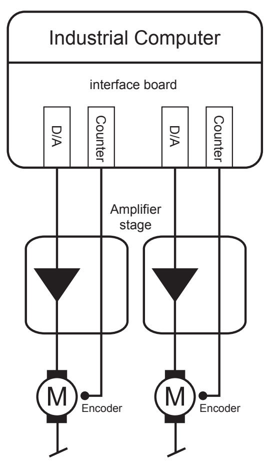


Figure 3: *A 2 Axes Classical Robot Control*

2.2. Fieldbus Usefulness for Motion Control

Contrarily to the classical analog motion controllers, the field buses receive the set-points in digital format through the bus. This digital communication gives the possibility to provide more than just the set-point. For example a drive that gets a velocity set-point as well as a feed-forward torque to improve the regulation performances.
The digital communication gives also the possibility to integrate error-detection algorithms to guarantee error-free transmission of all the set-points. This aspect brings a certain independence from the ambient noise, which is an ill in analog communication.
Another advantage of using a field bus is the important reduction of cabling material costs and maintenance costs. All the needed cables, between motor and drive, are provided by the servo drive manufacturer.

2.3. PROFIBUS Presentation

For our motion control application we have chosen the PROFIBUS[1] fieldbus. This choice has not been carried out due to any outstanding technical efficiency of this fieldbus. We only have been solicited by a manufacturer to perform a feasibility study of the use of its servo drives for parallel kinematics. The control architecture we propose can either be applied to CANOPEN or any other field bus.

PROFIBUS is a data bus which has been designed for use at the field level. This fieldbus is intended to interconnect plenty of devices of different types (e.g. sensors, drives, valves etc.). It permits the dialogue between equipment of different manufacturers without using highly specialized interfaces. Its universality and openness are assured by European and International standards. It is suitable for data transmission which require reflex actions and extremely short reaction times as well as transmission of high quantities of data. The transmission rate goes from 9.6 kbit/s up to 12 Mbit/s, depending on the chosen bus hardware.
There exist two different types of protocols called *communication profiles*:

The PROFIBUS-DP protocol. The DP protocol has been used in our case. It's the most widespread protocol in industry because of its rapidness, its performances and its cheap wiring. Its purpose is to be used between devices and control unit.

The PROFIBUS-FMS protocol. The FMS protocol is used for highly demanding communication tasks. This protocol privileges the richness of functionalities before the short response times. This protocol is more and more losing parts in the industry today. It gives the possibility to communicate between the single devices.

[1]Process Field Bus

3. PRESENTATION OF THE PROFIBUS PC BASED ROBOT CONTROL

A PROFIBUS PC based robot control architecture has been developed. Its architecture is shown in figure 4. It consists of a central computer unit (in our case it is a PC) in which is installed the PROFIBUS Master board, and beside them the drives which are connected in series. The drives consist of different parts which are the PROFIBUS-interface, the control-"intelligence" (position and velocity loops) and an amplifier stage.
The communication between PC and drives is assured by exchanging digital data-structures (so-called "telegrams") through the bus. The PC is generally sending the set points to the devices whereas the devices are sending measurement made on the axes back to the PC. Figure 5 shows the control hardware of our prototype.

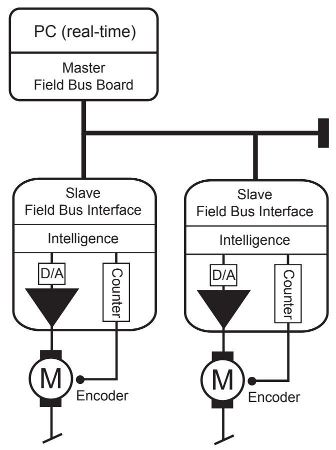


Figure 4: *2 Axes PROFIBUS PC based Robot Control*

Figure 5: *The Control Hardware. The figure shows the 5 drives of our prototype. They contain the PROFIBUS-interface, the control-"intelligence" and the amplifiers.*

3.1. The Control Software

This section presents the motion control architecture developed to implement the PROFIBUS layer. This controller is implemented on a *Windows* based PC with a real time extension. The controller synoptic is shown in figure 6. It is strongly flexible and is built up with different modules, some of them working as real-time processes.

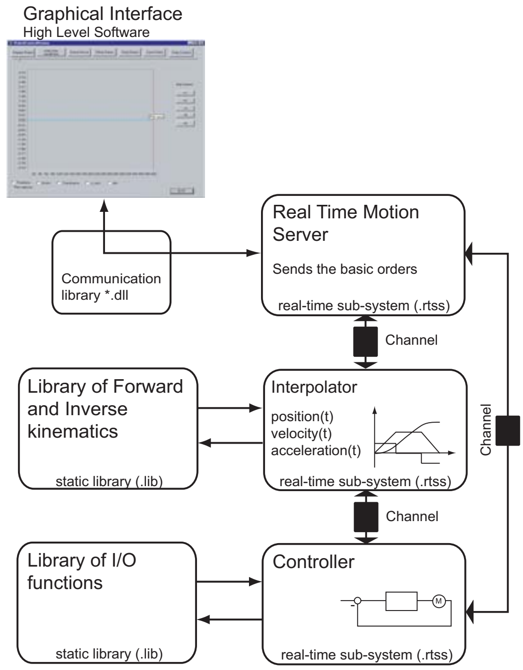


Figure 6: *The Control Software*

The modules composing the controller are described as follows:

The Graphical User Interface (GUI). The graphical user interface is the man-machine-interface used to communicate with the real time motion server by using the communication library (dll). It allows functions like *deploying the robot information*, *managing the data to display*, *start* or *stop the control*, ... etc.

Communication Library (.dll). This interface, provided as a *dll*, permits to communicate with the real time controller and to let it invisible for the user. It makes available all high-level instructions necessary to set up the motion and the control parameters.

Real Time Motion Server. The real time motion server is the principal door of the real time controller. It receives orders and parameters from the GUI via the *dll* and it dispatches them to the concerned processes (interpolator or controller).

Interpolator. It generates the trajectories with a specified geometry and a given dynamics. It generates the intermediate set-points at each cycle time.

Controller. The controller attends to the regulation of the set-points on each axis.

Library of Forward and Inverse Kinematics. This library contains the functions for converting the articular coordinates (the motor angles) into the operational coordinates (x, y, z, θ, ϕ) and vice versa. This library totally depends on the kinematics of the machine.

Library of Input/Output Functions. This library contains the functions that are directly interacting with the hardware layer of the control. The most important functions are for sending the servo drive's set-points and reading the positions of the axes. They are strongly associated with the functions made available by the hardware drivers. The library also contains the routines for the correct formatting of the servo drive's set-points (see section 3.2) that will be sent through the bus.

The cycle-time of the controller amounts 2ms. Up to 1ms is required by the routine for communication via the bus. These 1ms are, when using the 12 Mbit/s hardware, varying very slightly in function of the number of axes. This communication routine is provided by the hardware driver of the PROFIBUS Master-board and its execution-time can therefore not be modified.

This software concept permits an absolute easiness and flexibility for controlling any parallel or serial kinematics. Through this extremely flexible architecture the necessary work to adapt the software is kept to a minimum. The only module to be readapted for a new kinematics is the library of forward and inverse kinematics. When changing the hardware layer (e.g. CANOPEN, IEEE1394 or others), the library of I/O functions must be modified. All these mentioned advantages combined with the ability to work on a normal PC makes this control software a powerful tool for controlling parallel kinematics.

3.2. Bus Communication

The bus communication is based upon the telegram. It is a data structure with a specific format containing the information for all axes. It is used for sending as well as receiving information to/from the PROFIBUS servo drives. The figure 7 shows the specific format of the whole telegram containing the data for all 5 axes.

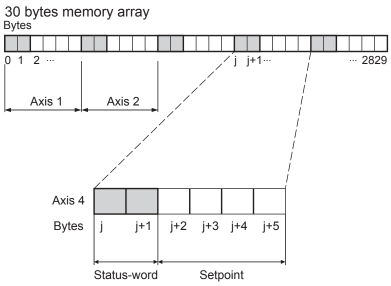


Figure 7: *The bus protocol for 5 axes and 1 set-point, in our case the velocity.*

Each axis has its own part of the memory-array. The driver addresses these single parts of the telegram to the corresponding devices. The whole telegram is sent to all axes which then themself, by comparing the addresses, are picking out their own part of the data. Each part corresponding to an axis is itself divided into more or less sub-parts. The first 2 bytes are the so-called status-word which contains either the information about the actual state of the device or the command of changing state, depending on the purpose of the telegram (sending or receiving). The following part of the telegram can contain a various number of set-points, which highly depends on the chosen control mode (position, velocity or torque). The figure shows a telegram for sending only one set-point. If more than one set-point is considered, the telegram just gets enlarged by some bytes that will contain the additional information. For example, if a feed forward torque algorithm is to be implemented, the user can choose a control mode where additionally to a velocity set-point a feed-forward torque is sent in the telegram. This capacity of sending various set-points is an important advantage of digital communication that can considerably improve the control performances.

Once all the information (the status-words and all the set-points) is correctly formatted and inserted in the data-structure the whole data-structure is sent through the bus using a simple routine. This routine is available via the hardware drivers.

3.3. Control Principle

The controller has to minimize the position error between the realized and desired motions. When working in position control mode, the PROFIBUS servo drives give only the possibility to set up the final desired position with its given dynamics. This mode may not be used in all robot motion planning because some robots have axes which have to be synchronized through the geometric model. To overpass this difficulties we have implemented the following control principle shown in figure 8.

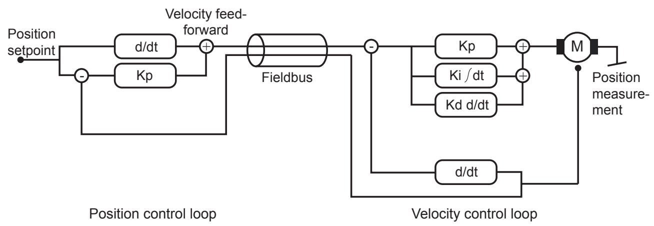


Figure 8: *The Regulation Loops*

Two cascaded control loops have been used: the velocity-control loop and the position-control loop. The position profile provided on the PC through the interpolator and the inverse kinematics is used for the position control loop and is differentiated to generate velocity feed-forward information. The generated velocity is then sent to the drives which are working in velocity-control-mode. The velocity-control loop is closed in the drive itself. It is working at $500\mu s$ cycle-time. This whole velocity control loop is invisible for the user. The only visible parts are the control parameters that can be changed via a special GUI (provided by the servo drive furnisher).

The actual position of the motors is read by the resolvers and sent back through the bus to the control software in degrees unit. The second loop, position-control loop, is closed through the whole fieldbus and the PC.

4. APPLICATION TO PARALLEL KINEMATICS

The aim of this section is to present our solution for controlling this complex 5-axes parallel kinematics. Additionally we will present some aspects that are intensely associated with the use of a parallel kinematics.

4.1. Kinematics

The Alpha5 kinematics were developed at the *Laboratoire de Systèmes Robotique (LSRO)* at the EPFL in Lausanne ([3],[1]). It is a totally parallel kinematics possessing 5 degrees of freedom with the ability to reach the $\pm$ $90°$ in amplitude on both, the A and B axis. For nowadays parallel kinematics this is a considerably high amplitude of rotation on both axes. This ability has been until now one of the big advantages of serial machines compared to parallel machines.
This section will only focus on aspects of the kinematics that are strongly associated with the control of the robot.
The whole mechanical structure can be divided into three sub-kinematics:

The Delta The Delta [4] is a well-known parallel kinematics which has 3 translational degrees of freedom. It connects the intermediate platform to the base.

The Alpha The Alpha is the outer structure which connects the base with the robot output and with the intermediate platform.

The Orion The Orion is a passive parallel structure which connects the intermediate platform to the lower platform. Its only purpose is to oppress the rotation of the robot-output around a vertical axis.

As it can be seen in figure 9 the kinematics is mainly based on 6 arms that are movable in rotation. These arms are distributed angularly all $60°$. Since the kinematics possesses only 5 degrees of freedom, only 5 of the arms have to be actuated.

4.2. Forward and Inverse Kinematics

The only mathematic model used for the control was the inverse kinematics. It is actually the easier model to compute for parallel kinematics.
The forward kinematics, known to be a complex problem in parallel robotics, was not necessary to determine thanks to an special initialization method (see section 4.3).

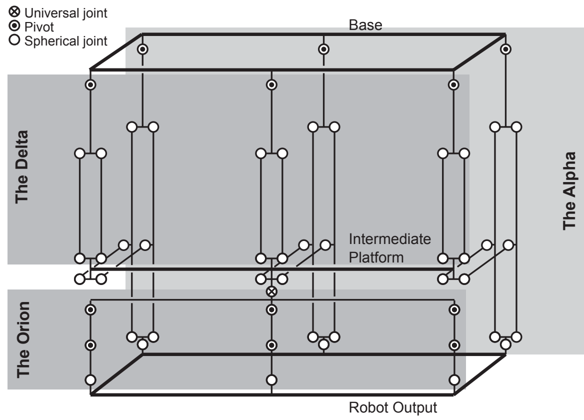
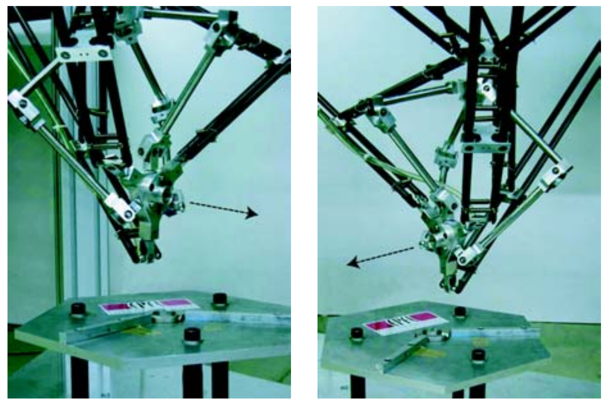


Figure 9: *The Alpha5 kinematics. This representation manner shows the kinematics in a pseudo 3-dimensionnal way. The Alpha structure is represented in the background contrary to the Delta- and Orion structure which are represented in the foreground of the figure. In reality all the arms are distributed angularly all 60°, the Delta- and Alpha-arms always alternating.*



(a) *The Robot Output Rotated by +90°* (b) *The Robot Output Rotated by -90°*

Figure 10: *The Robot Output in both extreme Orientations*

4.3. Initialization

The initialization of the encoders is done while the robot is in a parking-position. This means that the robot output is precisely positioned in a mechanical retainer from which we know the exact position and orientation. Knowing this position and orientation, we can compute the initial motor angles (or encoder values) through the inverse kinematics. The parking mechanics are shown in figure 11.
By initializing the robot this way the need for a forward kinematics model is useless. With a conventional initialization method, using mechanical references on the motor-outputs, the direct kinematics is needed to deter-

mine the initial operational coordinates from the referenced initial motor angles. The only negative aspect of such an initialization method ”by parking” is that at the moment of activation of the robot its position must be perfectly determined.

Figure 11: *The Parking-System used for Initialization of the Robot. This systems consists of three spheres placed on the end-effector and three v-shaped grooves on the workbench. When applying a light vertical force (own gravitational force) on the end-effector when the spheres are roughly engaged into the grooves the whole system will center itself and guarantee a perfect positioning for initializing the robot.*

4.4. Performance measurements

The following figures (12,13) show the control performances for the first rough dimensioning of the controller parameters. The figure 12 shows a certain position profile of one axis done during a task of the robot. The figure 13 shows the corresponding velocity profile. The maximal error in this time interval is 0.7 deg at $t = 3.5s$. At this moment we can measure an acceleration of about 500 deg/s^2.
Measurements for the static case demonstrate that the error does not exceed 0.03 deg.

5. CONCLUSIONS

The architecture of the control as well as the control software proved to be a fast and efficient manner for controlling this parallel robot. Thanks to the modularity of the control software it has been possible to rapidly adapt it for this new parallel kinematics and to the new fieldbus hardware. The only new routines that have been created for the digital communication have had as purpose to correctly format the set-points in order to match the given specification of the field bus. An impressive demonstration of the robot’s capabilities has finally been accomplished, where the robot paints the logo of the Swiss Federal Institute of Technology on a ping-pong ball using all 5 axes.

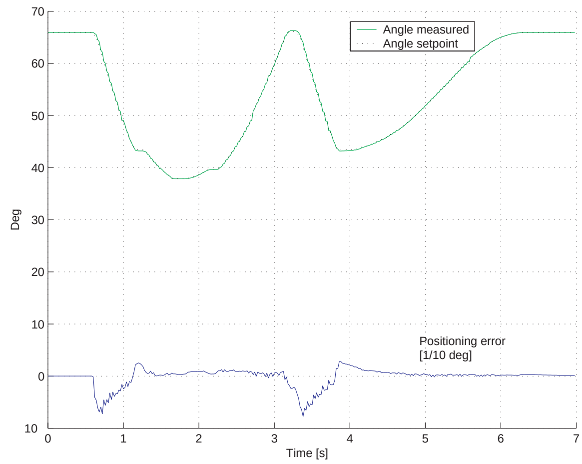


Figure 12: *Positioning Graph. Due to de chosen scale, the differences between the mesured and the setpoint angle can barely be seen on the upper curves. The differences are therefore amplified in the lower curve.*

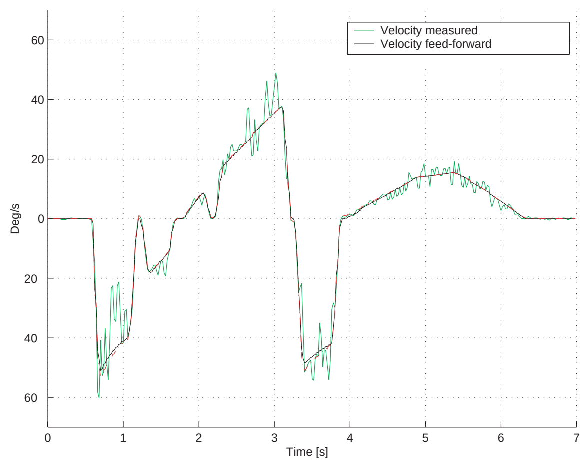


Figure 13: *Velocity Graph*

6. REFERENCES

[1] Pham, P., Brühwiler, B.: *Alpha5: Un robot parallèle à 5 ddl*, Internal Report, February 2002, EPFL Lausanne, Laboratoire de systèmes robotiques (LSRO)

[2] Pham, P.: *Alpha5: Commande d’Axes par Bus de Terrain PROFIBUS*, Diploma Project, February 2003, EPFL Lausanne, Laboratoire de systèmes robotiques (LSRO)

[3] Thurneysen, M.: *Développement de structures parallèles 4 et 5 ddl*, CTI intermediate report, 1998, EPFL Lausanne, Laboratoire de systèmes robotiques (LSRO)

[4] US Patent 4,976,582. Inventor: R.Clavel. Delivery date: Dec. 11, 1990. Priority Date: Dec. 16, 1985.