**International Conference on Control, Engineering & Information Technology (CEIT'14)**

**Proceedings - Copyright IPCO-2014**

**ISSN 2356-5608**

# Control architecture for a multi-robots system : Application to a V+ running controllers

**Wassim Mansour[1], Khaled Jelassi[1]**

[1]Centre Industriel Intégré de Production, Electrical Systems Lab, ENIT
El Manar University
Tunis, Tunisia
{wassim__mansour@hotmail.com; jelassi_2000@yahoo.com}

**Abstract.** This paper deals with the problem of multiple robot control systems, as a main component of the control of flexible manufacturing systems problem. First step was the establishing of the inter-process communication between both server (a V+ operating system running controller) and client sides (a windows OS running computer). The inter-process communication allowed a full control of robots using a unique client application. The model is brought to application through a four robots manufacturing cell. The inter-process communication is made via Dynamic Data Exchange (DDE). A client application is developed using C# language presenting a user interface (UI). This UI offers several functionalities including a teaching monitor and a sequences' editor.

**Keywords:** Industrial robots; V+ language; Dynamic Data Exchange; multi-agent system; ADEPT Robots.

## 1 INTRODUCTION

Robotics applications are becoming more and more essential in the industrial context. It provides enormous benefits and prominent competitive advantage for industrial companies. Due to its ease of production times control and its precision in tasks' execution, the use of industrial robots' arms in the field of manufacturing is taking place. However, the control of such devices and the providing of a better efficiency in tasks' execution is a wide field of investigation. Although several authors have been interested in different data processing and its role in inducing robot's behavior, such as vision systems [1, 2, 11] and torque/force sensors[12, 13], few of them have given attention to the interaction between software components and its role in optimizing the control of robotic cells.

As it is known, communication between a personal computer running a win32 platform and a robot's controller running another operating system needs communication protocol and condition-event logic to allow the user to have a total control over the cell from a simple user interface.

In this work, we aim to set up a software tool providing integrated environment for controlling 4 robotic arms and editing and running tasks' sequences for each of them. To achieve this goal, inter-process communication between different operating systems (win32 platform and V+ operating system) is developed to allow reliable dataflow in request/send data logic.

## 2 DDE CONTROL PROTOCOL – V+ KERNEL IMPLEMENTATION

The Communication between a robot's controller and a personal computer has taken several shapes through different similar works. Means and resources available at the time of implementation of such communication plays a key role in the techniques and methods used. Several authors used Remote Procedure Calls (RPC) [3, 4, and 5] as long as the robot's manufacturer provides

Proceedings - Copyright IPCO-2014

ISSN 2356-5608

RPC servers and software facilities to allow such inter-process communication. Some other authors use an internal controller API based on COM technique to communicate with an internal robot communication runtime provided by the manufacturer [6]. The API uses sockets and the local TCP/IP stack towards controllers.

This work is based on the model of control of industrial robot's arm made by Mansour [7]. The inter-process communication in this model is based on Dynamic Data Exchange (DDE). Although superseded be newer technologies, such as OPC, DDE protocol remained widely used in exchanging data between different programs running on the same environment [3, 8] and on different environments [9]. Only few authors have used such protocol to control robots despite its proven reliability [10]. Communicating with V+ running controllers using DDE protocol remains a good technique given the availability of free software components: DDE servers on both PC and V+ environments provided by the manufacturer (Adept Technologies). Fig. 1 shows the DDE mechanism on Adept controllers.

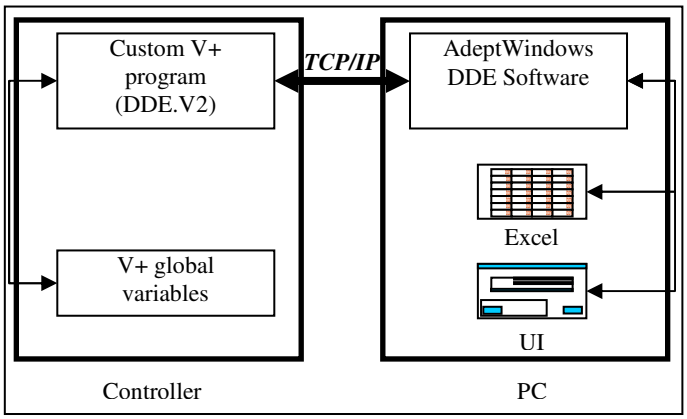


**Fig. 1.** DDE mechanism for Adept controllers

The control of the robot's arm is made through data exchange as shown in Fig. 2 [7]. DDE communication is achieved through two key elements (see Fig. 1):

- A server application that acts as a link between the TCP/IP protocol and the DDE service (running as a task in the Windows environment): The application AdeptWindows DDE.
- A V+ algorithm that transcripts data received via the Ethernet port into values, as well as sending variables (runs as a task on the controller's operating system): V+ custom program (DDE.V2).

Proceedings - Copyright IPCO-2014

ISSN 2356-5608

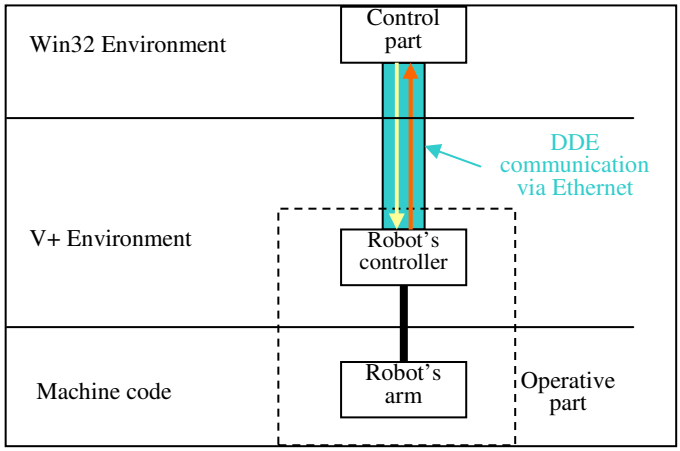


**Fig. 2.** Robot's command architecture

Exchanged data is an essential element for collecting data over robot's status (position, speed …) as well as for sending instructions.

Collecting data is made through simple DDE requests of variables carrying requested data. Adequate data is assigned to convenient variables using two ways:

- Initial assignment while DDE variables are declared in the DDE server running on the controller. Even though, these variables are updated permanently considering a polling time defined by the user. For example, in the following V+ code, a variable named "monitor_speed" is assigned to the V+ function SPEED(1) which returns the actual speed of the robot. Such variable is updated every 1 second.

```
$name[12] = "monitor_speed"
$read[12] = "SPEED(1)"
$write[12] = "SPEED #1"
poll[12] = 1
```

- Assignment through condition-event program loops. In this case an auxiliary variable is used to trigger the assignment of the new value.

Sending instructions is made through V+ programs running on the controller. These programs include infinite loops checking continuously several condition-event loops (if – then loops). Each loop is related to a given instruction as shown in Fig. 3 [7].

Proceedings - Copyright IPCO-2014

ISSN 2356-5608

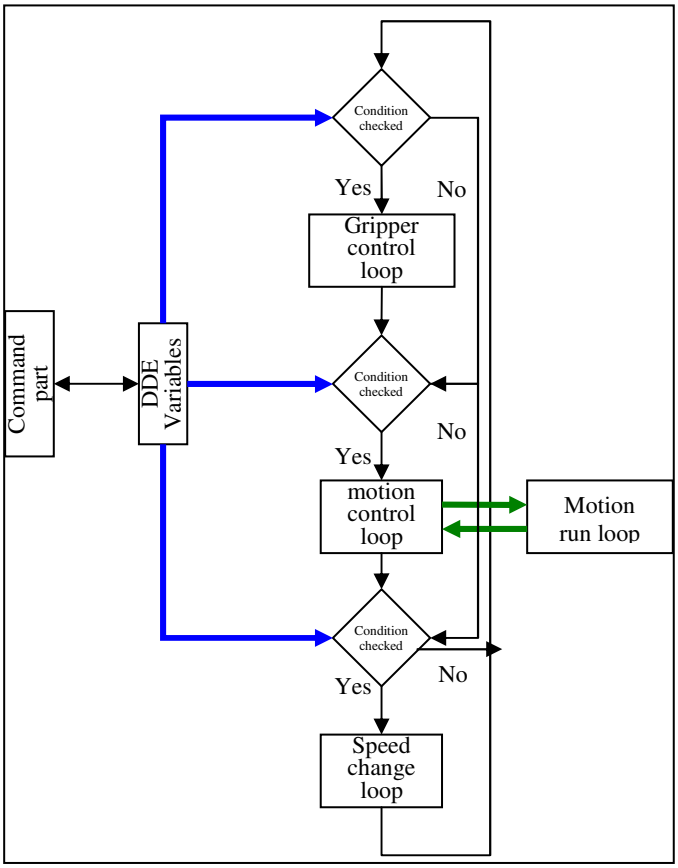


**Fig. 3.** Flowchart of the operative part (running on the V+ system)

Programs of the operative part run simultaneously on the controller's processor and interact together along with the DDE server to achieve the control of robot's motions and statuses by the final user.

Four programs are developed using V+ language and ran simultaneously on the controller's processor so that all required functions are accessible to the user:

- P1 : consists of an infinite loop checking continuously if-then conditions triggering each one a defined function :
  - Opening the gripper.
  - Closing the gripper.
  - Changing monitor's speed.
  - Receiving motion's data (coordinates) and triggering the motion loop (P2).

  Below is an example of such if-then loops. The following loop is used to change monitor's speed.

**Proceedings - Copyright IPCO-2014**

**ISSN 2356-5608**

```
IF var > 0 THEN
    attach
      SPEED var monitor
    detach
      TYPE "Speed has been set to", var
      var = 0
END
```

- P2: Consists also of an infinite loop waiting for a condition transferred from P1. In fact, in case of the if-then motion loop of P1 is triggered, coordinates of the destination are transferred and converted into a "point" type (V+ considers a special type for locations). Then, after checking the feasibility of the motion, a V+ "signal" (a special binary variable) triggers the starting of P2. P2 executes the movement and gives a feedback about the motion. Below, is the loop that receives coordinates and triggers the motion.

```
IF xx > 0 THEN
 nloc[2]=1
 for i=0 to 5
  if (xx==i*2+1) or (xx==i*2+2) then
   mcm[i]=xx*2-(i+1)*4+1
  else
   mcm[i]=0
  end
  Type " ",mcm[i]
 end
 SET point = TRANS(loc[0]+mcm[0]*yy, loc[1]+mcm[1]*yy, loc[2]+mcm[2]*yy, loc[3]+mcm[3]*yy,
loc[4]+mcm[4]*yy, loc[5]+mcm[5]*yy)
  IF INRANGE(point) == 0 THEN
          nloc[1]=0
        SIGNAL 2001
        WAIT NOT SIG(2001)
          ELSE
                nloc[1]=1
                xx=0
          END
      for i=0 to 5
           mcm[i]=0
  END
END
```

Proceedings - Copyright IPCO-2014

ISSN 2356-5608

The following V+ algorithm represents the main part of P2.

```
WHILE 1 DO
        WAIT SIG(2001)
        attach
        move point
        detach
        xx=0
        SIGNAL -2001
END
```

- P3: Consists of several nested loops allowing the reading of edited motions by the user (pseudo-code) and their execution as long as sending continuously feedback about robot's status (speed, position, gripper status, skipped motions, motions' incertitude, …). A brief description is given in Fig. 5.
- P4: Is the DDE server algorithm. It consists of several sub-functions interacting together in order to achieve the transcription of data received from the TCP/IP protocol and sending data in a recognizable form (DDE protocol).

## 3 CONTROL TOOL DEVELOPMENT (.NET)

The control model presented above is extended to the control of a whole flexible manufacturing cell. Fig. 4 shows the layout of the considered manufacturing cell.

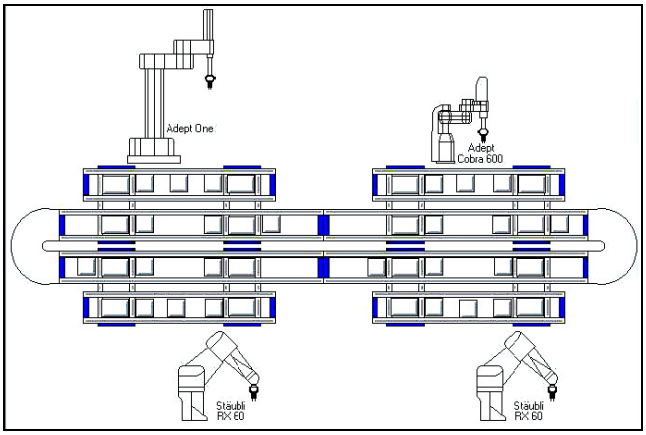


**Fig. 4.** Layout of the implementation cell

The considered cell includes four industrial robots' arms which are the following:

- Adept Cobra 600.
- Adept One.
- 2 Stäubli RX 60

Proceedings - Copyright IPCO-2014

ISSN 2356-5608

Both Adept robots are SCARA robots having 4 axes. Stäubli robots have 6. The cell comprises other elements such as conveyor belts, cameras …

The developed software was designed to allow the user the following features for each of the four robots' arms:

- Moving the robot's arm to desired locations using an MCP-like (Manual Control Pendant) utility.
- Saving position to database.
- Editing and running sequences of actions including:
    - Moving to already saved positions
    - Opening/closing the gripper
    - Waiting
    - Changing speed

The software allows the user to choose the number of times a sequence should be ran. This feature has been implemented using two separate algorithms one on the server side (pc) the other on the client side (V+ system). The two algorithms communicate using DDE variables in the same logic of condition-event loops. Due to the difference of processors' speed between the computer and the robot's controller an adequate synchronization is incorporated. Fig. 5 shows a schematic representation of the communication between these two algorithms.

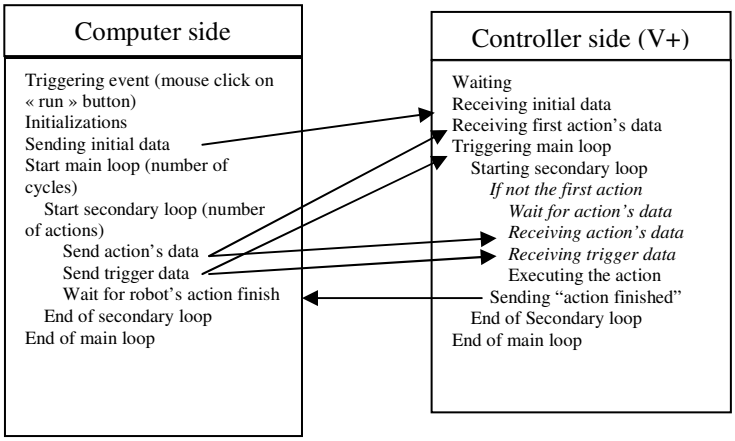


**Fig. 5.** Communication between "control" and "operative" algorithms

Considered data in Fig. 5 are as follows:

- Initial data : Number of cycles, number of actions per cycle
- Trigger data : Boolean variable (for each trigger event)
- Action's data : Type of action and related data for each :
    - Move : destination's ordinates
    - Open/Close (Boolean)
    - Wait : delay
    - Change speed : new speed
- "action finished" data : Boolean variable

Proceedings - Copyright IPCO-2014

ISSN 2356-5608

The whole system is implemented through developed software in C# language using Visual Studio 2010. The UI (Fig. 6) integrates all the features discussed above.

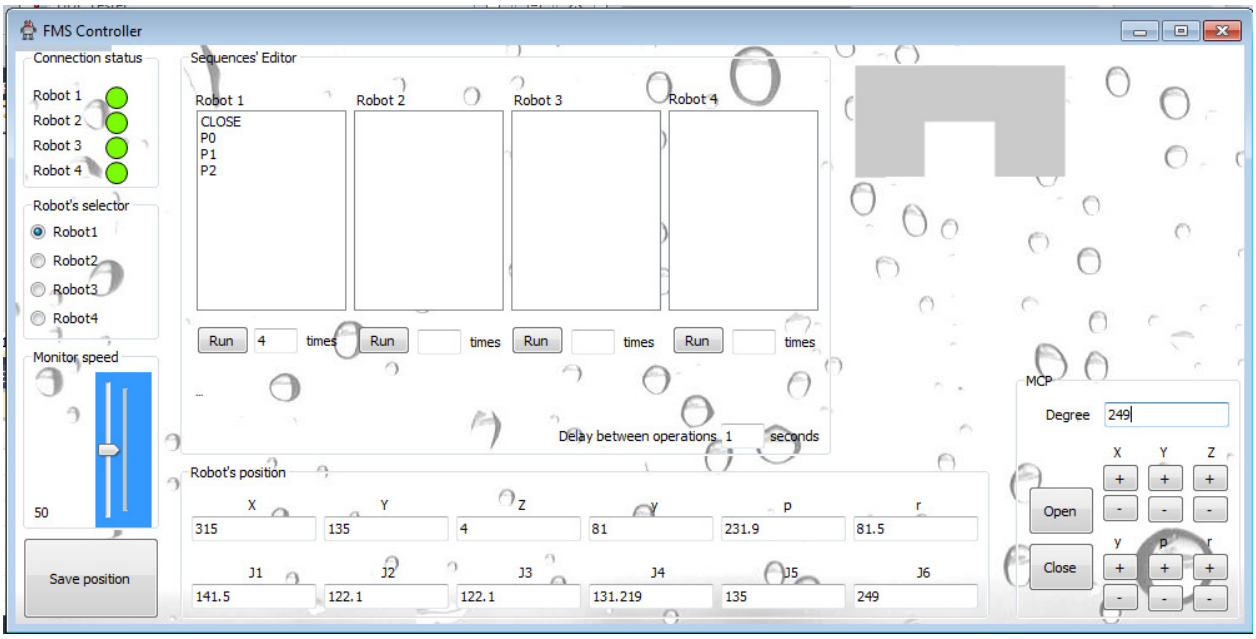


**Fig. 6.** User interface

An Oracle database has been integrated to the software so the user can save edited sequences for further use. Those sequences generally correspond to a given product treatment. More precisely, the sequences are manufacturing operations that could derive from an ERP.

## 4 CONCLUSION

In this work, we've been interested in the implementation of a software tool allowing the edition of manufacturing sequences subsequently to teaching the robot. To accomplish this aim, an adequate combination of algorithmic structures had to be employed in order to guarantee efficient data transmission and synchronization between both operative and control parts which run different operating systems. Inter-process communication was made using Dynamic Data Exchange and it proved its reliability in our context. The built system has a satisfying robustness (no failures in data exchange).

Further researches may be directed towards linking a production scheduling algorithm to the existing system so that managing already saved sequences would be such that it optimizes a given criterion. Another research axis may be the integration of a real time kernel so that synchronization between tasks assigned to different robots will be guaranteed.

**ISSN 2356-5608**

## References

[1] Yanfei Liu, Adam Hoover, Ian Walker, Ben Judy, Mathew Joseph and Charly Hermanson : A New Generic Model for Vision Based Tracking in Robotics Systems, Proceedings of the 2003 IEEURSJ InU. Conference on Intelligent Robots and Systems Las Vegas, Nevada ' October 2003 248-253.

[2] Yanfei Liu, Adam W. Hoover, and Ian D. Walker : A Timing Model for Vision-Based Control of Industrial Robot Manipulators, IEEE TRANSACTIONS ON ROBOTICS, VOL. 20, NO. 5, OCTOBER 2004, 891-898.

[3] J. Norberto Pires, J.M.G. Sá da Costa : Object-oriented and distributed approach for programming robotic manufacturing cells, Robotics and Computer Integrated Manufacturing 16 (2000) 29-42.

[4] Peter Corke, Wen Hu, and Matthew Dunbabin : An RPC-based Service Framework for Robot and Sensor Network Integration, Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd, 1-6

[5] G. Glez. de Rivera, R. Ribalda, J. Colás, and J. Garrido : A Generic Software Platform for Controlling Collaborative Robotic System using XML-RPC, International Conference on Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME, 1336-1341

[6] Csokmai Lehel Szabolcs: COMPARISON BETWEEN TWO METHODS OF ROBOT CONTROL IN THE PRODUCTION OF PRISMATIC PARTS, Nonconventional Technologies Review, Romania, June, 2012, 27-31.

[7] Wassim Mansour, Khaled Jelassi, Imen Chaieb Memmi: Control of an Industrial Robot using Ethernet - Application to a V+ Running Controller, Electronic Devices Volume 2 Number 2 September 2013, 53-58.

[8] Carstoiu, D. ; Brodschi, A. ; Eftimiu, C. : Netware dynamic data exchange, Electrotechnical Conference, 1994. Proceedings., 7th Mediterranean, 284 - 287 vol.1.

[9] Yuanmu Deng, Keerthi Shet, Haihong Li, Periannan Kuppusamy, Jay L. Zweier Z : Real-time calculation and visualization of spectra in field-cycled dynamic nuclear polarization spectroscopy, Computer Methods and Programs in Biomedicine Volume 82, Issue 1, April 2006, Pages 67–72

[10] L.Ken Keys, Robert Hirschfeld, P.B. Desai, A. Sastry : Intelligent manufacturing systems (Smart Mechatronics & CIM) at Louisiana State University, Mechatronics Volume 5, Issue 7, October 1995, Pages 743–752

[11] Th. Borangiu, F.D. Anton, Anamaria Dogar : Visual Robot Guidance in Conveyor Tracking with Belt Variables, International Conference on Automation, Quality and Testing, Robotics, May 2010. 1:1-6

[12] Anton Satria Prabuwono, M.A. Burhanuddin, Samsi Md. Said : Autonomous Contour Tracking Using Staircase Method for Industrial Robot, 2008 10th Intl. Conf. on Control, Automation, Robotics and Vision Hanoi, Vietnam, 17–20 December 2008, 2272-2276

[13] Eric Barnett, Jorge Angeles, Damiano Pasini, Pieter Sijpkes Surface Mapping Feedback for Robot-Assisted Rapid Prototyping, 2011 IEEE International Conference on Robotics and Automation Shanghai International Conference Center May 9-13, 2011, Shanghai, China 3739-3744