Real Time Robot Controller Abstraction Layer

François PERTIN
Software architecture team leader
Jean-Michel BONNET-DES-TUVES
Software team leader

Stäubli Robotics Division
74210 Faverges France
+33 (0)4 50 65 60 60
www.staubli.com
f.pertin@staubli.com

Abstract:
This article is intended to present Stäubli's CS8 robot controller, and more particularly the Real Time Robot Controller Abstraction Layer, an original solution to adapt a "standard" industrial robot to a specific application. Readers will understand why this solution represents a choice platform to develop a dedicated control.
The system architecture, its advantages and principles of use are presented in this article which is concluded by an example illustrating how easy it is to implement this solution.

Keywords : trajectory generator, open robot controller, application programming interface

1. Introduction

Using robots in engineering schools, universities and applied research centers is a plus, both for students and teachers. Robots are key elements to tackle subjects such as industrial automation, or important peripherals to implement some high-tech processes (in the nuclear, biology, space fields, …).
In many of these situations, a "standard" industrial robot cannot be used, or is misused; indeed, either it does not offer THE required functionality, or a specific control mode is required, or an already existing control software must be used.
Integrating an industrial robot mechanics into a dedicated system then proves a very expensive and uncertain operation. It requires a good knowledge of the robot (not always easy to acquire from the manufacturer), of its control signals, of the electronics used, of safety standards, …
It also requires to solve subsidiary but crucial issues, such as the setting of servoing parameters, the calibration of mechanics, the robot safety. In addition, maintenance tools associated with these functions have to be developed.
As a company of experience, traditionally oriented toward this world of research and innovation, Stäubli has developed an original and high-performance solution to meet this need: the " Real Time Robot Controller Abstraction Layer ".
RTRCAL is a software package that manages the lowest-level functionalities required to control a robot through a reduced C/C++ interface.
The robot is considered by this interface as a set of independent and intrinsically safe joints control, safety and calibration.
It leaves complete freedom to the Client application which has to manage all aspects that build up a robot from a set of joints: its geometry, its kinematics, its dynamics, …
In return, RTRCAL Client should be served by the Client application in "real-time" since commands have to be sent to the joints at the frequency required by the servoing.
Moreover, the interface allows the use of the other CS8 controller's peripherals (input/outputs, …).

2. A solution for a versatile controller

The system is made up of one robot of Stäubli's robot range and of the CS8x 8th-generation controller.

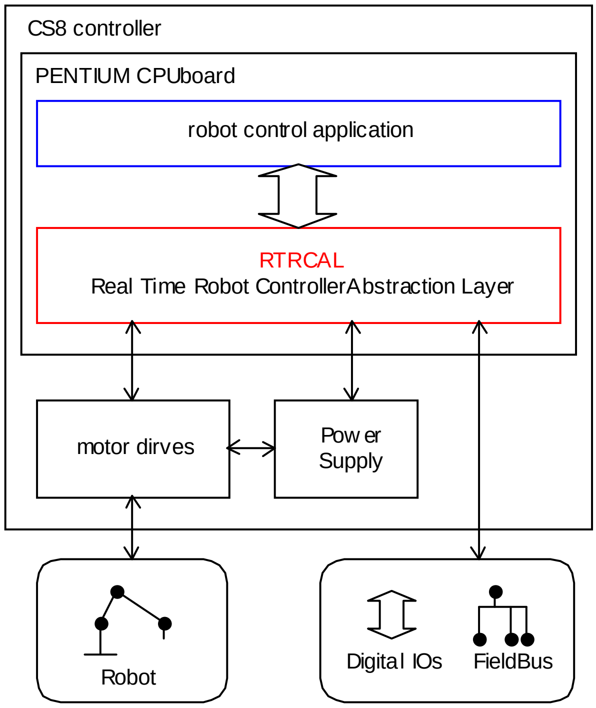

Figure 1: CS8 general architecture

RTRCAL is a software package that commands the controller's hardware resources under the control of an application.
The system's versatility is based on the fact that it can equally be used as a "standard" industrial controller (IHM, VAL3 programming language, etc) or for a dedicated application.
The choice between an environment and another is made at the controller boot ("dual boot").
The ease with which one can switch from one mode to another allows for instance to use the "standard" configuration for all maintenance operations without having to recreate them in the dedicated application.

The CS8 controller integrates a PENTIUM® arithmetic unit and the VxWorks® real-time operating system. The C or C++ programming will then be made with Windriver®'s Tornado® environment of development.
To make the implementation easier and to perform simulations, RTRCAL is also available in the form of a Windows® DLL. In this configuration, the hardware resources (robot and digital inputs/outputs) are simulated.
So, without any additional development, an application developed with RTRCAL can be equally simulated with Windows® or with the real controller.

3. An easy implementation

One of RTRCAL's most attractive aspects lies in the fact that the complete robot joints control is done through an extremely reduced interface. It is ensured by a state machine providing a secured synchronization of unitary control components (joints, brakes, power, inputs and outputs).
The synchronization between the robot joints control and the application is done through the automatic call of a "handler" attached to the joints servoings' real-time clock.
Typically, at each clock pip, the "handler" sends a new command (cmd) to be applied to the different robot joints. This command is calculated in particular from feedback information (fbk) such as the position, the velocity or the motor currents, or from other information on the control state (stat).

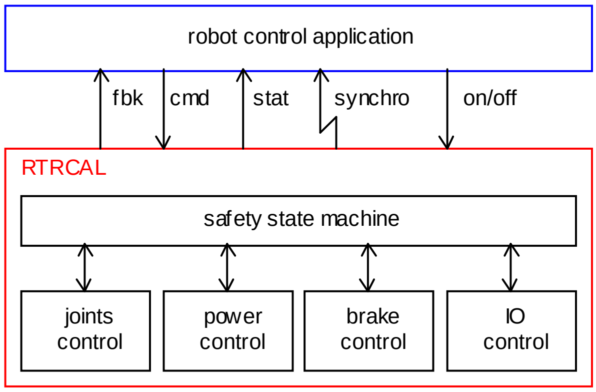

Figure 2: RTRCAL implementation

4. A reliable solution

Error information detected by RTRCAL is immediately available to the application, which should then stop the robot depending on its own constraints, in a suitable temporal window. If it is not the case, RTRCAL generates an emergency stop.
In all cases, the conditions that caused the error are archived on the controller's disk (they can be recovered on a PC using an Ethernet® link) and sent to the serial link.
This information can be used for an online or deferred diagnosis.

5. For an adapted robot control

Generally, the system receives commands (cmd) and provides information (fbk) concerning the different robot joints. Typically, these control data are associated with the robot's trajectory generator included in the application.

Figure 3 summarizes the control structure and the use made of these data.

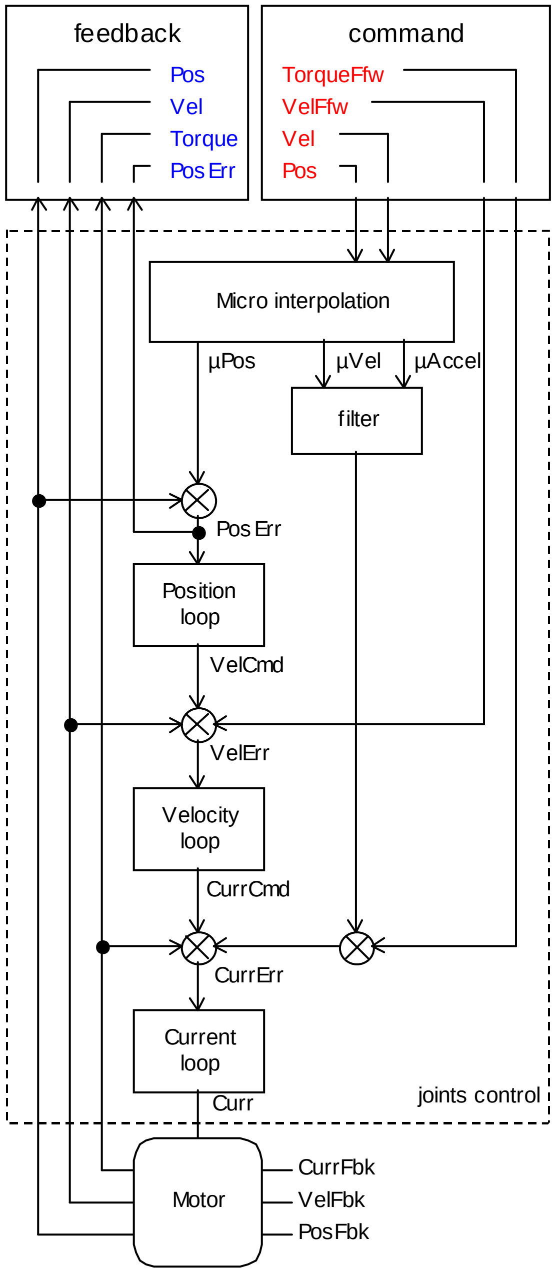

Figure 3: joints control structure

The feedback information (fbk) is made up of the position (pos), the velocity (vel), the torque (Torque) and the position error (posErr). The command (cmd) is represented by the position (pos), the velocity (vel), a velocity anticipation (velFfw) and a torque anticipation (TorqueFfw).

It should be noted that the application can use part of this structure only: for instance, if we use the current loop only, we can adapt the system for a stress control.

In addition, the system checks the consistency of the commands calculated by the application before sending them. Inconsistent cases (too large variation between two commands, contouring error, …) generate an error.

Finally, although the loops' parameters are perfectly suited to the use with robots (which represents a significant advantage), Stäubli can provide tools to change gains upon specific request.

6. Management of the robot absolute position

The robot is delivered calibrated with a configured CS8 controller; it is then ready for use. Nevertheless, if the user wishes it, he can specify the robot absolute position by developing a dedicated calibration algorithm to find the robot absolute position and insert the result in RTRCAL.

7. Other accessible resources

RTRCAL allows to access the different resources available on the standard industrial controller. In particular, the digital inputs and outputs, the modBusTCP and the FlashDisk.

The APPLICOM® solution is available to use the DeviceNet, modBus, profibusDP and CanOPen field buses.

All inputs/outputs are accessible via a unified interface whatever their physical implementation.

Direct access to the FlashDisk is also possible using the VxWorks® primitives. The two /sys et /usr predefined partitions can be used for the backup of user files.

8. Conclusion

With the RTRCAL layer, experienced programmers have a versatile robot controller that can be used in a "standard industrial" or "Research & Development" configuration.

This approach guarantees the permanence of the development already done, since Stäubli ensures an ascending compatibility of its interface with all robots of its range and with the future CS8 controller's evolutions.

9. Example of use

Define interrupt service routine

```
void isr (void) { star tSynchronous task }
```

Construct RTRCAL interface

```
LLI_RobotId  robotId = LLI_construct(&isr);
```

Synchronous task

```
FOREVER
{
    Wait for ISR signal
    // get feedback
    LLI_get (robotId , &fbk);
    // get status
    LLI_state (robotId , &stat);
    // Compute command (trajectory generator)
        myTrajectoryGenerator (stat, fbk, &cmd);
    // send computed command
    LLI_set (robotId, &cmd);
}
```

Define trajectory generator

```
myTrajectoryGenerator (stat, fbk, &cmd)
{
        switch(stat)
        {
        case LLI_DISABLED :
                // following mode,
                // copy feedback in command
                cmd.m_pos = fbk.m_pos ;
                cmd.m_vel = 0 ;
        break ;
        case LLI_ENABLED :
                cmd.m_pos = myValue;
                cmd.m_vel = myValue;
        break ;
        default :
                // position is the same
                // and velocity is 0
                cmd.m_vel = 0 ;
        break ;
        }
}
```

Enable the robot

```
LLI_enable(robotId);
```

Disable the robot

```
LLI_disable(robotId);
```

Get input or input handler

```
LLI_getDoutId (robotId, "OUT0" , &doutId)
LLI_getDinId  (robotId, "IN0" ,   &dinId)
```

Write an output

```
LLI_writeDout(dout, DOUT_ONE);
```

Read an input

```
LLI_readDio(dinId, &status);
```

10. References

http://www.windriver.com
http://www.staubli.com

11. Equipment

1 Robot
1 CS8 controller
1 CDROM including:

- documentation,
- the executable of the CS8 program under Windows
- the RTRCAL libraries for Windows and VxWorks
- an example of functional use under Windows and VxWorks, ready for use.

11 RTRCAL users

CYBERNETIX
306 rue albert einstein
technopole chateau combert
bp94-13382 Marseille 13
Contact: Pascal Baclet, Simon Geffroy

EDF-der
bp 496,15 quai wattier
78401 chatou cedex
Contact : Jacques Guillet, Mr Georges Mauvillet

UNIVERSIDAD DE JAEN
Avda. Madrid, 35
23071 JAEN
tel. 953 00 24 43
fax. 953 00 24 00
Departamento Electronica
Contact: Juan Gómez

INSTITUT DE ROBOTICA
Parc Tecn. Barcelona,Edif. U
c/Llorens i Artigas, 4-6 2ªpl
08028 BARCELONA
tel. 93 401 57 84
fax. 93 401 57 50
Contact: Federico Thomas

INSTITUT D'ORGANITZACIO I CONTROL DE SISTEMES INDUSTRIALS (IOC)
Diagonal,647,Planta 11
08028 BARCELONA
tel. 93/4016653
fax. 93/401 66 05
Contact: Luís Basañez