Implementation of dynamic controllers using real-time middleware for a low-cost parallel robot

J. Cazalilla, M. Valles, M. Diaz-Rodriguez, V. Mata, A. Soriano, A. Valera

Abstract— **As it is well known, robots are complex devices capable of performing very fast and precise movements. Developing robots and its controllers is a very challenge task due to the different technologies that are needed to be dealt (access to peripherals, real-time operating systems, communications …) and the amounts of programming work that is needed. Moreover, this work needs to be repeated in most cases when a new robot is developed again. So, a new way for developing the code implemented for controlling these robots that allows to reuse it in a safe way is needed.**

In this paper, the Orocos (Open Robot Control Software) middleware has been chosen for the controller development. Orocos is a real-time middleware, focused to control systems, especially those related to robotics and automation. Its greatest advantage over the other solutions available is the capability to provide an off-the-shelf hard real-time operation. This is essential in most of the robotics applications, converting this middleware in a very suitable piece of software. Since Orocos is a component-based middleware, several CBSD (component-based software development) techniques have been used to design and implement the control system. Thus, using a modular control structure, a number of advantages can be achieved such as code reusability, execution of the modules in a distributed way, ability to load or instantiate a module several times, easy following of flow execution and fewer programming errors. In addition, these components are configurable and reconfigurable in runtime.

In fact, in a typical control application, there are three parts that can be thought as software components that are reading sensors, computing the control action, and sending it to the actuators. Besides this, in some applications it is also possible to find common parts to different control strategies that can be implemented as separated software components that are loaded, instantiated and configured depending on the control strategy that it is needed in a particular moment.

The robot that has been controlled using Orocos modules is a low cost parallel-robot PRS. The parallel distinction means that the terminal end of the robot member is connected to the base with a number of independent members, with the particularity that they all work in parallel. The main advantages of this type of robots are the high speed and accuracy, and the ability to carry heavy loads.

In order to read the encoders of the robot (one at the base of each joint), the acquisition card Advantech PCL-833 has been used. For sending the control action the acquisition card Advantech PCI-1720 has been used, with range from -10V to +10 V and a precision of 12 bits.

Different dynamic model-based controllers has been developed as modular components using Orocos middleware for controlling the parallel robot. The main feature of these controllers is to use the natural energy of the system (kinetic and potential) in such a way that the control objective is achieved.

One of these controllers is the passivity-based one proposed by Paden and Panja. The controller has a compensation of the dynamic parameters of the robot (gravity, inertia and coriolis), and a PD regulator. By implementing this controller in a modular way, it is possible distributed execution of the modules with higher computational load, reducing total execution time. The sample time used is 100Hz (10 ms), obtaining an error position less than 0.5 mm, and a runtime close to 0.5 milliseconds.

Reusing and dynamically loading some of the modules used at the previous controller and adding others, a force control and a hybrid force-position control, using the ATI Delta force sensor, has been implemented also. The force sensor is made of steel and silicon strain gauges, thus the measurement error is nearly zero. A good performance has been achieved in both experimental results as design and implementation process of the new controllers.

At the same way, using the position modular control foresaid, a communication module between the industrial computer and a WiiMote (by Bluetooth protocol) has been developed that allows maneuver the robot from this peripheral. Since Orocos middleware is implemented in C++, it is relatively easy to get both the buttons state of the WiiMote as the accelerometer measures. Thus, it is possible to control the robot with high accuracy.

Finally, some applications related on the rehabilitation of ankles and knees using the parallel robot have also been implemented in a modular way. They consist in placing the foot over the platform of the parallel robot. Thus, the robot can be programmed to encourage the patient to perform active, passive, and assisted movements. As seen above, because of the high precision of the robot, these movements are made with high accuracy, so it would ensure that the movement is made correctly. Furthermore, using the force sensor, the forces applied to the ankle and knee can be monitored in real-time. Thus, depending on the forces measured the position reference can be modified in order to perform an assisted movement.

J. Cazalilla. Instituto de Automática e Informática Industrial, Universidad Politécnica de Valencia, Spain, CO 46022. Corresponding author; e-mail: jcazalilla@ai2.upv.es

M. Valles. Instituto de Automática e Informática Industrial, Universidad Politécnica de Valencia, Spain, CO 46022. ; e-mail: mvalles@ai2.upv.es

M. Diaz-Rodriguez. Facultad de Ingeniería, Universidad de los Andes, Mérida, CO 5101 Venezuela; e-mail: dmiguel@ula.ve

V. Mata. Centro Investigación de Tecnología de Vehículos, Universidad Politécnica de Valencia, Spain, CO 46033: E-mail: vmata@mcm.upv.es

A. Soriano. Instituto de Automática e Informática Industrial, Universidad Politécnica de Valencia, Spain, CO 46022; e-mail: ansovi@ai2.upv.es

A. Valera. Instituto de Automática e Informática Industrial, Universidad Politécnica de Valencia, Spain, CO 46022; e-mail: giuprog@ai2.upv.es