First Steps Towards an Open Control Architecture for a PUMA 560

Nuno Moreira, Paulo Alvito, Pedro Lima

Instituto de Sistemas e Robótica

Departamento de Engenharia Electrotécnica e de Computadores

Instituto Superior Técnico, Torre Norte

Av. Rovisco Pais, nº 1, 1096 Lisboa Codex

e-mail: pal@isr.ist.utl.pt

Abstract

In this paper, progress efforts towards an open control architecture for a PUMA 560 manipulator are described. The original UNIMATE architecture was modified to allow direct control, trajectory planning, task planning and external sensors handling by a network of dedicated personal computers. Meanwhile the new platform was tested by the implementation of one-PD-per-joint and computed torque controllers.

The first steps towards the new architecture and the results obtained with the controllers are presented in the paper.

Keywords : Open Control Architecture, Robotic Manipulator, Computed Torque Controller, PD Controller.

1. Introduction

The PUMA 560 is a six-joint industrial robot manipulator, whose original UNIMATE MARK III Controller includes a DEC LSI-11/02 processor, six digital and analog servo boards and two amplifier boards. Each of these amplifier boards drives three joints of the manipulator. The digital servos have a Rockwell 6503 microprocessor to interpolate the trajectories provided by the LSI-11/02 processor. The analog boards, connected to the digital boards through a Digital to Analog Converter (DAC), contain an analog servo designed as a current feedback controller for each joint motor.

Interaction with the user is possible through the VAL-II operating system. The LSI-11/02 interprets VAL-II statements and generates joint interpolated coordinates, sending them to the six 6503 microprocessors.

This architecture offers many difficulties when used for high level control tasks, since the operating system VAL-II is based on a closed architecture. Resources to accomplish sensing, such as vision and force, are not supported. Also, the joint controllers have fixed gains and it is impossible to change them or to change the trajectory generation algorithm. Hence, the implementation of new control, trajectory generation or task planning algorithms is not possible.

The methodology followed to avoid the above mentioned limitations, was to replace the operating system VAL-II and most control hardware of the manipulator. The new installed hardware allows direct access to the joint positions and torques. Simultaneously new software was developed to replace the VAL-II basic functions.

This groundwork will allow future development of control algorithms based on external sensing (hybrid position/force control and vision based control). It will also allow trajectory and task planning supported by an open control architecture based on primitive tasks (`move`, `plan trajectory`, `follow trajectory`, etc.).

The paper is organised as follows:

Section 2 describes the original control architecture of the PUMA 560 system, followed by the report of the changes introduced in the new architecture (section 3). The methodology for controllers implementation and test is described in section 4. Section 5 presents and discusses the results obtained with the controllers.

2. Original Control Architecture

The PUMA 560 original control architecture is depicted in Fig. 1.

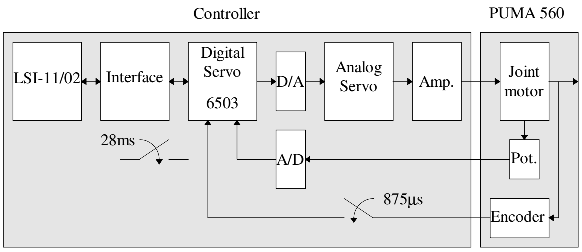

Figure 1 - Original Control Architecture

The functions of each module of Fig. 1. are described below.

2.1 LSI-11/02 Processor

Interprets the VAL-II commands and sends position setpoints to the six 6503 microprocessors at a rate of 35.7 samples per second.

2.2 Digital Servos

There are six digital servos, each of them containing a 6503 microprocessor. The 6503 obtains the actual joint positions and velocities from the encoders, at a rate of 1.2 Ksamples per second. This higher rate requires an interpolation of the set points established by the LSI-11/02, to attain a rate of 1.2 Kset points per second.

From the position and velocity errors, the 6503s calculate the torque values that are transmitted to the analog servos through the DACs.

2.3 Analog Servos

The analog servos must track the torque set points provided by the digital servos, by controlling the current of each joint motor.

2.4 Amplifiers

The power amplifiers supply the necessary power required by the joint motors.

2.5 Incremental Encoders

The encoders provide three output signals - channels A, B and Index Pulse.

Channels A and B are in quadrature, making possible the detection of movement direction. The index pulse is used to detect a complete encoder revolution. This signal is combined with the potentiometer reading to calibrate the encoder readings.

2.6 Potentiometers

The joint angular position provided by the encoders is only a relative, not absolute measure. To establish the initial encoder reading at arm power startup, the encoder readings must be calibrated. This is accomplished by using the absolute angular position provided by the potentiometers. The potentiometers values are read by the Analog to Digital Converter (ADC).

The potentiometers values are not used until the next arm power startup.

3. Open Control Architecture

3.1 Hardware Design

To provide a greater system flexibility and expansion capacity, the new installed hardware supplies direct access to the joint positions and bypasses the old joint controllers. This new system is based on the Trident Robotics cards TRC004 and TRC006.

The new system composed by the TRC004 and TRC006 cards plus a Personal Computer (PC), replaces the old system composed by the LSI-11/02 processor, the VAL-II operating system and the joint controllers.

The remaining components from the original architecture are the incremental encoders, potentiometers, power amplifiers and analog servos.

The boards containing the CPU, RAM, EPROM, serial controller, interface cards, and the joint servos were all removed from the control card cage. The TRC004 card replaces their functions, handling the analog circuits kept in the controller cage.

The TRC006 card provides an interface between the TRC004 and the ISA bus of the PC.

The block diagram of the new architecture is represented in Fig. 2.

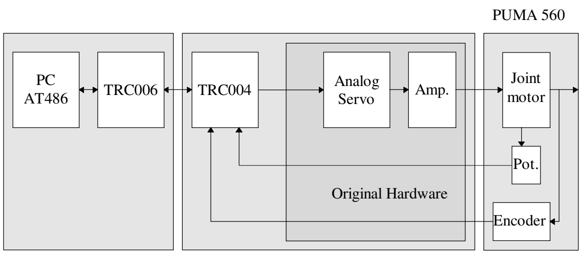

Figure 2 - New Architecture

In the new architecture, the set point update rate and the torque update rate have identical values. Due to the fact that the external computer has to perform all the calculations associated with the trajectory planning and the control algorithm every sampling period, there is an upper bound on the sampling rate used. The maximum sampling rate achieved using a 486 DX2 processor was approximately 400 Hz, but in the tests presented here the sampling rate used was 200 Hz.

3.2 Software Design

The initial step in software design was the development of routines to replace the basic functionality provided by VAL-II: power on joint calibration, joint position overflow test, and maximum joint velocity test.

In the next step, inverse kinematics and inverse Jacobian matrix calculation routines were implemented to provide the user means to define positions and velocities along a trajectory, in Cartesian coordinates. The trajectory knot points are currently interpolated by a cubic polynomial.

In this first approach, the main goal was to test the new control architecture, so only partial effort was spent to achieve modularity. In the near future, primitive tasks will be implemented and used as building blocks for task planning and execution, supported by a network of PCs.

4. Implementation and Test of Joint-Space Controllers

To test the functionality of the new hardware architecture, two different joint-space controllers were implemented: a PD controller and a computed-torque controller. As explained before, a simple user interface allows the user to input a collection of knot points which define a trajectory in cartesian space, including desired position, orientation and velocity of the end effector. The knot points are then converted into joint space, where they are interpolated by a cubic polynomial per joint.

The PD controller, consisting of a set of six decoupled PDs (one per joint), is depicted in Fig. 3.

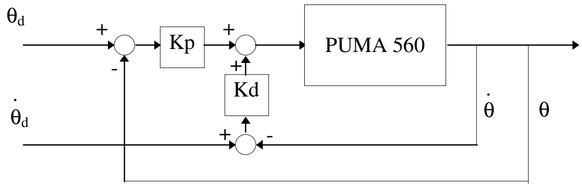

Figure 3 - Single Joint PD Block Diagram

The gains Kp and Kd were empirically adjusted, and are listed in Table 1.

Joint	1	2	3	4	5	6
K_p	700.0	1100.0	400.0	40.0	30.0	40.0
K_d	20.0	20.0	20.0	5.0	5.0	5.0

Table 1 - PD Gains

Even though it can be proved that, with full compensation of gravity and stiction terms, the PD controller is asymptotically stable [5], its dynamic performance is only acceptable at low speeds [2]. This problem lead to the introduction of non-linear controllers which include the manipulator dynamics in their control laws. If the model is accurate enough, non-linear dynamics can be cancelled and the tracking error asymptotically driven to zero. The computed-torque controller is a good representative of this class of controllers.

The dynamic equations of the PUMA 560 manipulator are [1]:

$$\tau = M(\theta)\cdot\ddot{\theta} + B(\theta)\cdot(\dot{\theta}\dot{\theta}) + C(\theta)\cdot(\dot{\theta}^2) + g(\theta) + S(\dot{\theta}) + F\cdot\dot{\theta}$$

where,

$M(\theta)$ - Inertial matrix [6×6]
$B(\theta)$ - Coriolis matrix [6×6(6-1)/2]
$C(\theta)$ - Centrifugal matrix [6×6]
$g(\theta)$ - Gravitic vector [6×1]
$S(\dot{\theta})$ - Stiction vector [6×1]
F- Friction vector [6×1]
θ - Joint positions vector [6×1]
$\dot{\theta}$ - Joint angular velocities vector [6×1]
$\dot{\theta}\dot{\theta}$ - Cross velocity products [6(6-1)/2×1]

Figure 4 shows the block diagram of a closed loop computed-torque controller for the PUMA.

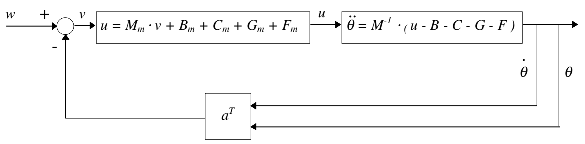

Figure 4 - Computed-torque controller

In the diagram,

$$a^T = \left[\begin{array}{cccccc:cccccc} a_1 & 0 & 0 & 0 & 0 & 0 & a_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_1 & 0 & 0 & 0 & 0 & 0 & a_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_1 & 0 & 0 & 0 & 0 & 0 & a_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1 & 0 & 0 & 0 & 0 & 0 & a_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_1 & 0 & 0 & 0 & 0 & 0 & a_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_1 & 0 & 0 & 0 & 0 & 0 & a_2 \end{array}\right]$$

is a matrix of error and derivative of error gains (a_1 and a_2, respectively),

and $$w = \ddot{\theta}_d + a^T \cdot \begin{bmatrix} \theta_d \\ \dot{\theta}_d \end{bmatrix}$$

From Fig. 4,

$$v = w - a^T \cdot \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \ddot{\theta}_d + a^T \cdot \begin{bmatrix} \theta_d - \theta \\ \dot{\theta}_d - \dot{\theta} \end{bmatrix} \quad (1)$$

And the following expression for $\ddot{\theta}$ is obtained, by replacing u in the dynamic equations:

$$\ddot{\theta} = M^{-1} \cdot (M_m \cdot v + B_m + C_m + G_m + F_m - B - C - G - F) \quad (2)$$

If the model equations match exactly the actual PUMA dynamics (i.e., $M_m=M$, $B_m=B$, $C_m=C$, $G_m=G$, $F_m=F$), the following differential equation is obtained by cancellation of terms and replacement of v in (2) by its expression in (1):.

$$\ddot{\theta} = \ddot{\theta}_d + a^T \cdot \begin{bmatrix} \theta_d - \theta \\ \dot{\theta}_d - \dot{\theta} \end{bmatrix} \quad (3)$$

Equation (3) corresponds to the following second order differential equation for the position error:

$$\ddot{e} + a_2 \cdot \dot{e} + a_1 \cdot e = 0$$

One can choose the values for a_1 and a_2 so that the closed loop system will track a reference trajectory like a critically damped system.

Choosing a value of 10 $rad \cdot s^{-1}$ to the natural frequency of the system and a damping ratio of 0.5, the following values are obtained for a_1 and a_2.

$$a_1 = 400 \text{ rad}^2/s^2, \; a_2 = 20 \text{ rad/s}.$$

In the tests performed with the real system, it is computationally expensive to compute all the dynamic components of the PUMA 560 model for every sampling period. Furthermore, Leahy et al [2] showed that the effect of neglecting the Coriolis and centrifugal terms is not appreciable, and the elimination of the non-diagonal terms of the inertial matrix leads to improved performance of some link controllers. Hence, only the gravitational terms and the diagonal elements of the Inertial matrix were computed. In the tests, the parameter values of the dynamic model (e.g., masses, moments of inertia) in [1] were used.

5. Experimental Results

Several experiments were made to test the two controllers. Representative results are shown here.

The knot points of the trajectory used in all tests are listed in Table 2.

knot point	Ready	2	3	4
Time (s)	0	4	7	11
P_x (m)	0	0.45	-0.45	0
P_y (m)	0	0.45	0.2	0
P_z (m)	0.878	0.3	0.5	0.8
O (°)	0	45	-30	0
A (°)	90	45	30	0
T (°)	0	45	30	0

Table 2 - Test trajectory knot points

The angles O, A and T are the ones described in Unimation PUMA 560 Manual [4]. The resulting desired trajectories for the six joints are shown in Fig. 5.

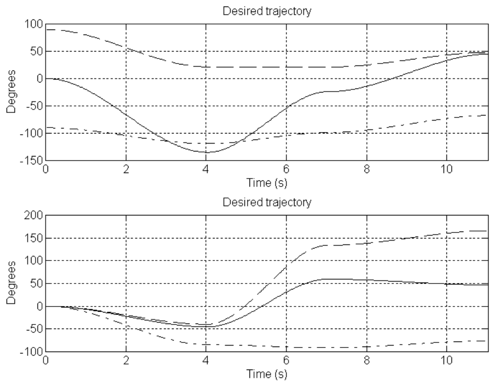

Figure 5 - Desired Trajectory
Joints 1,4 —
Joints 2,5 —·
Joints 3,6 ——

The tracking errors and the applied torques when the PD controller is used are plotted in Figures 6 and 7, respectively.

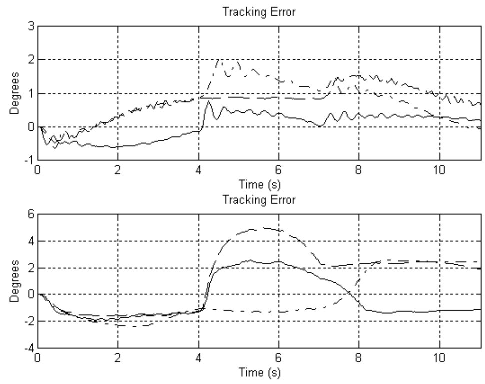

Figure 6 - Tracking Error for PD controller
Joints 1,4 —
Joints 2,5 —·
Joints 3,6 ——

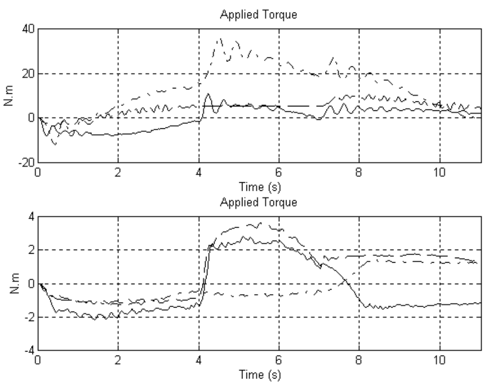

Figure 7 - Applied torque for PD controller
Joints 1,4 —
Joints 2,5 —·
Joints 3,6 ——

The corresponding results for the computed-torque controller are shown in Figures 8 and 9 respectively.

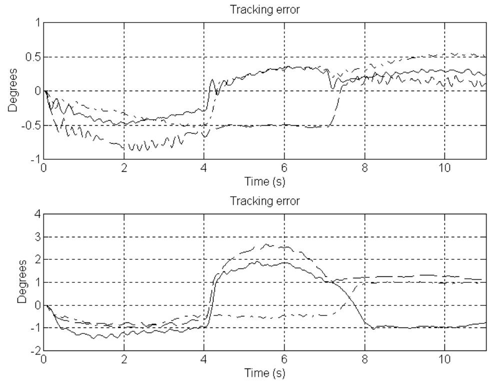

Figure 8 - Tracking error for
Computed Torque controller
Joints 1,4 —
Joints 2,5 —·
Joints 3,6 ——

The results show that, in steady state, both controllers present the same error magnitude. The error is not zero due to the non-cancellation of stiction terms, and the unmodeled joint motor dynamics and backlash.

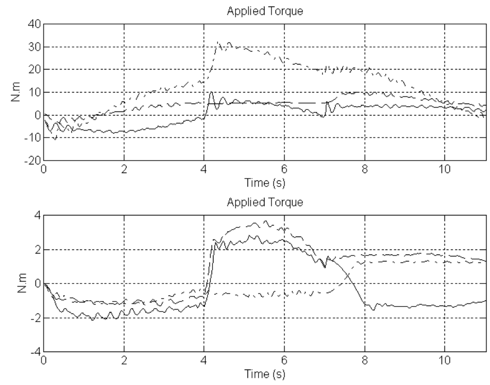

Figure 9 - Applied Torque for
Computed Torque controller
Joints 1,4 —
Joints 2,5 —·
Joints 3,6 ——

The beneficial effects of using the computed torque technique are best observed in dynamic tracking, where the observed error is significantly smaller, for similar torque amplitudes.

6. Conclusions and Future Work

The first steps towards an open control architecture for a PUMA 560 were described.

The added effort of implementing all the basic functions previously implemented by VAL-II, was compensated by the perspectives of future development that the new system offers.

With the new platform it was already possible to test different controllers and trajectory planners for the manipulator. The next step will consist on the specification and implementation of a set of primitive tasks which will be the building blocks of future developments. These will include implementation of hybrid position/force controllers, vision based controllers, telemanipulation, and efforts towards full autonomy. The task primitives will return performance evaluations to be used for overall system performance and task planning as proposed by Lima and Saridis [3].

References

[1] Brian Armstrong, Oussama Khatib, Joel Burdick, "The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm", IEEE, 1986.

[2] M. B. Leahy, K. P. Valavanis Jr, G. N. Saridis, "Evaluation of Dynamic Models for PUMA Robot Control", IEEE Transactions on Robotics and Automation, Vol. 5, Nº 2, April 1989.

[3] P. Lima, G. Saridis, "Design of Intelligent Control Systems Based on Hierarchical Stochastic Automata", World Scientific Publ. Co., 1996.

[4] PUMA MARKIII ROBOT - 500 Series - Equipment Manual.

[5] Mark W. Spong, M. Vidyaságar, "Robot Dynamics and Control", John Wiley & Sons, 1989.