EMBEDDED CONTROL SYSTEM FOR HEXA PARALLEL ROBOT

MSc. Amadeusz Fedus, West Pomeranian University of Technology
(Prof. Stanisław Bańka, West Pomeranian University of Technology)

Abstract
In this paper was presented implementation and assumptions of embedded controller based on 32-bit ARM7 microcontrollers for Hexa parallel robot. Furthermore, there was presented implementation of angular position digital controller. The most difficult problem was to find the solution of inverse kinematics with respect to the limited processors capacity. Therefore, to achieve better performance similar to commercial solutions the process of control has been divided into four tasks computed on separate processors.

1. Introduction
Embedded systems can be found in various applications: from industrial control systems, to telecommunication, which is the result of their advantages: small dimensions and small power consumption. Other advantage which makes them popular is large number of peripheral devices. The most difficult problem in developing hardware-based control system is a need of designing them from scratch. Moreover, appropriate components of the embedded system (processors etc.) must be selected regarding process requirements. Hence, solutions based on PC (equipped with Input/Output card) are more flexible, easier to develop but their undoubted disadvantage is high cost.

Hexa parallel robot [1] (Fig. 1) was designed as a stiffness actuator with 6-degrees of freedom used in the study of effective force control for industrial robots [2]. With respect to the limited time the MIRPA (Middleware for Robotic and Process Control Applications) system employed software control architecture [3], due to fast prototyping and ensures easy use of the controller of parallel platform, as a superior component (hybrid position/force control [4]).

Besides above advantages of the PC-based controller, the cost of stand alone hardware part in described example has been relatively high (about 5 thousands Euro – control computer, fast prototyping computer, IO cards). Additionally the mobility of such a system has been far from expected. Considering above, at the design phase it has been decided to use two types of control, i.e. software one for research purposes and hardware one for didactical reasons.

The second solution which has been developed for thesis purposes [5] is the topic of this work.

2. Components of control architecture
Hexa robot architecture was implemented on AT91SAM7S microcontrollers with ARM7 core. These processors were chose with respect to their large number of peripheral devices (10-bit AD converters, Timers with PWM mode and serial communication devices). ARM7TDMI is family processor with RISC architecture which means it utilizes a small set of instructions [6]. Processors with ARM7 core were enhanced with 32 and 64-bit instructions for higher performances [7]. Thanks to the simple structure of the core of ARM7 and ARM9, those microcontrollers has been used more often in applications which previously required use of DSP processors.

For high efficiency, process of control was divided into few tasks computed on four microcontrollers which was demonstrated in Fig. 2.

Below algorithms realized by new control architecture and corresponding requirements are listed:
- Servo position controllers of six angles – sampling time (1ms) of each servo controller was selected with the respect of high dynamic of coreless DC motor.
• Inverse Kinematics – for smooth movement of moving platform in Cartesian space angles shouldn’t be computed longer than 10ms.
• Serial communication with Magellan Space Mouse to manually control position and angles of platform
• Visualization on PC

![Diagram of control architecture](image1)

**Fig. 2. Diagram of control architecture**

The division of tasks on separate microcontrollers led to the development of effective communication between microcontrollers, which was a critical point of control system. To meet the requirements concerning sampling time, SPI and I2C communication were used. The Real system mounted on the motherboard is shown in Fig. 3. The Motherboard is provided with connectors which gives ability to use hardware driver or PC-based control system [1]. Its main purpose is to combine control system with manipulator actuators, which demands accurate matching of signals. Every active joint is driven by Hitec servo (HSR5995-TG) which is often used in various robotic applications. To measure angular position using potentiometer and implement more complex control algorithms controller inside the servo was replaced.

Coordinates systems were attached according to the Denavit-Hartenberg convention for closed-loop mechanisms [10]. Global coordinate system {B} was attached with the base of robot and is a reference point for other coordinate systems.

Hexa robot has six degrees of freedom which means the moving platform can obtain any position and orientation in robot’s workspace.

![Hexa Robot Controller mounted on the motherboard](image2)

**Fig. 3. Hexa Robot Controller mounted on the motherboard.**

### 3. Inverse Kinematics

For serial robots the motion planning can be specified in Cartesian or joint space, but in case of parallel robots motion planning can be specified only in Cartesian space. Solution of inverse kinematics can be found if we assume that angles in every joint are limited (complexity of solution of inverse kinematics is inverse in case of parallel robots [9]). Fig. 4. shows how to combine coordinate systems with robot, which is starting point of the algorithm.

![Coordinates systems attached to the Hexa robot](image3)

**Fig. 4. Coordinates systems attached to the Hexa robot**

Distances between {Pi} and {Ji} coordinate systems determine position and orientation of moving platform. Thus, the homogenous transformation matrices $T_{Pi}^{B}$ should be found:

$$T_{Pi}^{B} = T_{B}^{P}T_{Pi}$$

The matrix $T_{Pi}^{B}$ determines representation of moving platform frame with respect to the base frame. Matrix $T_{Pi}^{B}$ describes transformation of set {P} to set {B} and takes under consideration rotation of {P} coordinate system by the angles ($\alpha$, $\beta$, $\gamma$) and displacement vector D. Matrix can be presented in the following form:

$$T_{P}^{B} = \begin{bmatrix} R_x(\alpha)R_y(\beta)R_z(\gamma) & D \\ 0 & 1 \end{bmatrix}$$

391
Matrices $R_x(\alpha), R_y(\beta), R_z(\gamma)$ describe rotation in $x, y, z$ axes and can be written in following form:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where $D$ stands for displacement vector:

$$D = [x_p \ y_p \ z_p]$$

The result of calculations at this stage are origins of ${\Pi_i}$ coordinate systems with respect to the ${\Omega_j}$ coordinate systems. Next step demonstrates how to find vector of joint variables using geometrical method, which was showed in Fig. 5. The problem in present stage of calculations is that there is no information about values of angles in ball-and-socket joints.

$$\begin{align*}
(L_2)^2 &= (L_1^2 + (L_1')^2 - 2L_1L \cos \gamma) \\
(L_1')^2 &= x_p^2 + y_p^2
\end{align*}$$

Using Cosine Rule for angle $\gamma$ in $\Delta_{ABD}$ triangle leads to the following equation:

$$\gamma = \arccos \left( \frac{L_1^2 + (L_1')^2 - (L_2)^2}{2L_1L} \right)$$

Finally substituting 10, 11, 12 in 8 we obtain desired configuration angles in active joints:

$$q = \frac{\pi}{2} - \arccos \left( \frac{L_1^2 + L_1'^2 - L_2^2}{2L_1 \sqrt{x_p^2 + y_p^2}} \right) + \arctg \frac{x_p}{y_p}$$

The main goal of the project was to achieve calculation realization time for inverse kinematics not higher than 10ms, which was going to ensure smooth movement of the platform on space. In spite of high complexity of the algorithm (number of operations of extracting the roots: 12, number of trigonometric functions: 18, number of multiplication/division/addition 224/12/119), microcontroller with 48MHz core clock has determined the configuration angles in time of around 2.1 ms.

4. PD Servo Controllers

Regardless of kinematics structure, manipulator is complex MIMO (Multiple Input Multiple Output) object which has many dynamic and static nonlinearities. In parallel robots all moving parts have small mass concentrated near to base [8], which reduces static nonlinearities (gravity). Despite this, changing dynamic has the same result as inertia perturbation seen from the side of the drives. This particular problem can be solved using gears resulting in lower speed of moving platform, which is rarely seen in applications like pick-and place or assembly. Another disadvantage of using gears in active joints is that more advanced control algorithms must be develop [9].

A construction assumptions are taken directly from its purpose of use and allow to use drives with gears and simple PD control loop. It’s clear that this simple control system may be implemented on 8-bit microcontroller.
ARM processor was chosen regarding future implementation of more complex algorithm (i.e. feedforward) which was another project assumption. Fig. 6. depicts diagram of single joint controller including PWM modulator, PD controller and redundant system monitoring each servo position, protecting servo’s gear from damage. Because strong nonlinearities of object exclude well-known design methods of PD controller, settings where found experimentally. Fig. 7 shows control quality for single joint controller.

5. Manual setting of trajectory

For demonstration reasons, the method for manual setting of trajectory for Hexa platform has been researched. As has been mentioned earlier, the control is done in Cartesian space, the setting of the position had to be working in above space. Those requirements are satisfied by mouse with 6 DOF, which conception has been created in 1970 in german DLR (Deutschen Zentrum für Luft- und Raumfahrt). The system has RS232 interface (now only USB) and sends coded coordinates in frequency of 50Hz. Because of too short time of setting values of coordinates, it has been decided to add function of interpolation, thanks to which the vibrations of construction had been eliminated during the manual control. [11] describes movies demonstrating work of platform with initial software control.

6. Conclusions

The article presents construction of the controller based on 32 bit microcontroller from ARM family. Besides small size of the system and low cost of realization its usefulness has been shown in application of control of the manipulator with 6 DOF. Although its overall functionality does not differ much from commercial systems, it is mainly dedicated for educational purposes and is to be treated as help of understanding the specific issues of control for closed kinematic chains.

References


Author

MSc. Amadeusz Fedus
West Pomeranian University of Technology
Al. Piastów 17
70-310, Szczecin
Tel. (091) 449 54 06
fedus@zut.edu.pl