

# MCS-8 EIGHT AXIS EMBEDDED MOTION CONTROL SYSTEM

G. Jansa, R. Gajsek, M. Kobal, Cosylab, Ljubljana, Slovenia

## Abstract

MCS-8 is an 8 axis motion control system for stepper motors, servo drives and a variety of special drives. Central to the controller is the Delta Tau Turbo PMAC 2 programmable multi axis controller. It incorporates a full PLC specifically designed for positioning and control functions.

We have developed a layer on top of the generic and complex Turbo PMAC 2, making the MCS-8 very easy to use as a stand alone box in a control system. Software running on embedded controller can range from EPICS, SPEC to local SCADA system with an Ethernet and RS232 interfaces.

MCS-8 is capable of controlling even the most complicated motion such as Stewart-Gough platform (Hexapod). Hexapod offers six degrees of freedom positioning system with sub-micrometer precision and repeatability. PMAC controller is used to calculate kinematics and EPICS for end user interface. Important features are user selectable point of rotation in space and point to point scanning of all six axes.

## INTRODUCTION

In particle accelerators there are many applications that require controlling of motors. These applications range from controlling injection devices in storage ring, such as undulators and wigglers, to components in beamlines which can be as simple as slits or complicated devices such as Double crystal monochromators (DCM) or mirrors. Especially DCMs usually requires controlling of different types of motors such as servo motors with dual feedback for velocity and position, stepper motors, or even nano or pico motors for fine positioning of crystals. MCS-8 is built in a modular way so that it supports any combinations of motors.



Figure 1: MCS-8

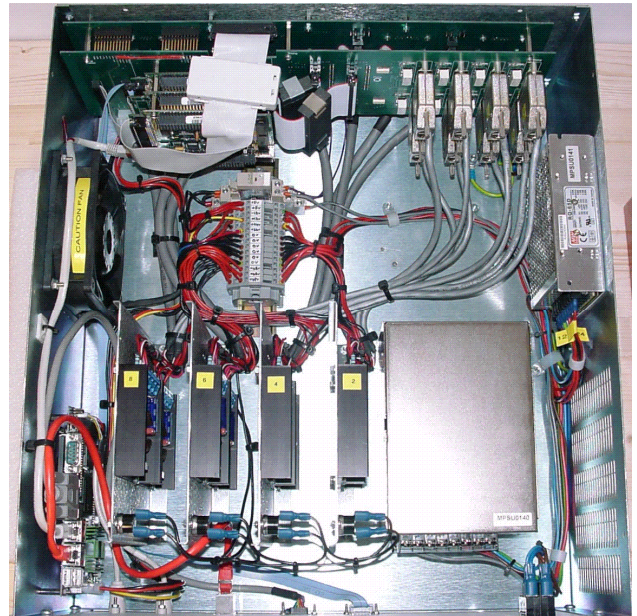


Figure 1: MCS-8 inside

Main hardware components of MCS-8 are Turbo PMAC 2, embedded computer (microIOC) and power drives. MCS-8 software is divided to low level software residing on Turbo PMAC 2. Basic motion control includes programs for single axis point to point moves, home reference search and various housekeeping tasks, programs synchronize motion of multiple motors can be achieved. Robotic applications control can be performed using kinematics calculations. In recent projects we used EPICS as an interface to PMAC for controlling most devices used in synchrotron light beamlines such as slits, mirrors, DCM, hexapod or individual motors. We use EPICS GUI written in EDM or MEDM [1]. Panels hide the complexity of the software. Very popular way of interfacing MCS-8 is by using SPEC [2]. SPEC is a package for instrument control and data acquisitions widely used for X-ray diffraction at synchrotrons. MCS-8 comes without embedded computer if used with SPEC, since SPEC is usually installed on a remote linux workstation.

This article describes the components of the MCS-8: Turbo PMAC 2, microIOC and various power drives. Second part presents an example of usage of MCS-8 to control hexapod acting as optical element in ADRESS beamline in Swiss Light Source (SLS).

## MCS-8 COPONENTS

### Turbo PMAC 2

Turbo PMAC 2 is the heart of MCS-8. It uses the increased speed and memory of the newest generation of digital signal processing (DSP) ICs. It has capability to control up to 32 axes in 16 independent coordinate systems. Turbo PMAC 2 board itself has at most 8 axis interface channels. To actually control more than 8 physical axis users must use either special expansion board or remote interface circuitry on the MACRO ring [3]. The DSP of the base version of turbo PMAC 2 runs at 80 MHz. Serial interface is used to directly interact with PMAC (can be used for manual configuration) using Delta Tau software running on Windows. Ethernet is used for internal communication between PMAC and microIOC.

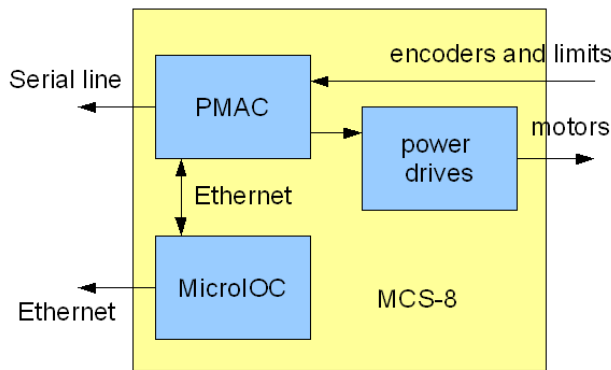


Figure 1: MCS-8 main components

Software on the Turbo PMAC 2 is divided into: setup, motion programs, homing programs, PLC programs and servo and phase algorithms. Setup on Turbo PMAC 2 is performed by means of setting system variables called I variables [3]. There are more than 8000 variables in total and 100 for each motor. I variables are used for settings like maximum velocity, soft limits, PID parameters. Motion programs are programs that are usually used to move the motors. They provide an easy way to specify sequences of coordinated motion of multiple axes and the execution of any calculations that are synchronous with the programmed motions.

Homing programs are needed if the encoder on the motor is relative and the absolute position is lost after a power failure or a hard reset. Basic homing program drives motor to one of the limit switch and then the home reference search sequence is initiated. Other types of homing programs for motors that don't have home reference are also possible (they can use one of the limits switch as a home indicator).

PLC programs are intended for actions and calculations that are asynchronous to the programmed motions. PLC programs repeatedly scan in the fashion of regular

programmable logic controllers. They are used for initialization, monitoring of various registers, to perform manifold actions at certain conditions, such as switching off power drives after motion for motors that would otherwise overheat and they can also be used to implement actions which PMAC would perform if communication to the remote controller would fail (e.g. stop all motors).

Servo algorithm implements PID loop for each motor. In this algorithm PMAC calculates required signal for the outputs based on the difference between actual and demanded position. Standard servo loop is shown in figure 2.

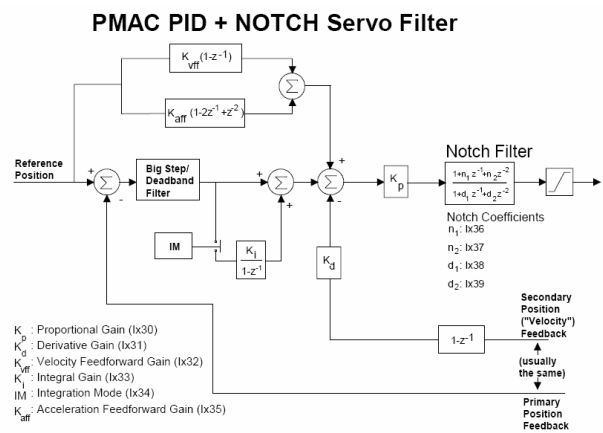


Figure 2: Standard servo loop

Advanced users can write their own servo algorithms. Since these tasks are executed at very high priority they can also be used as a very fast input/output operations or calculations if they are applied on motors that are not used in the system.

### MicroIOC

MicroIOC [4] is embedded single board computer (SBC) usually running software for controlling devices. It has standard outputs for mouse, keyboard, VGA monitor, 2 USB plugs and also two Ethernet connections. All these except one internet connection can be accessed from MCS-8 front panel. One of the Ethernet connections is used for internal communication between PMAC and EPICS software. It does not have a hard disk; all software is stored on a compact flash disk (CF) which is loaded at boot time. Most widely used operating system is linux (Debian distribution) but also Windows or RTEMS are possible.

### Power drives

The PMAC board provides control for a wide variety of actual drives. Some of the supported are:

- Mounted internally in MCS-8
  - Microstepper Driver – this drive is a PWM Chopper, providing bipolar drive for 2-Phase stepping motors.

- Servo Motor Driver - This is an intelligent PWM servo drive designed to drive brushed and brushless servomotors.
- Other drive options mounted externally to MCS-8:
  - DC brushless motors
  - Intelligent “Picomotor” drives
  - Nanomotors
  - Piezoelectric drives

## HEXAPOD APPLICATION

Parallel kinematics manipulators (PKMs) have been rediscovered in the last decade as processor's power finally satisfies computing force required for their control. Their great payload capacity, stiffness and accuracy characteristic as result of their parallel structure make them superior to serial manipulators in many fields.

One of the most accepted PKMs is Stewart-Gough platform based manipulator, also known as hexapod platform. Hexapod consists of two platforms, one fixed on the floor or ceiling and one mobile, connected together via six extensible struts with spherical or other types of joints. That construction gives mobile platform 6-DOF (degrees of freedom). Hexapod movement and control is achieved only through strut lengths changes.

### Kinematics

In hexapod motion control it is essential to have capabilities of calculation inverse and forward kinematics. PMAC provides two special buffers, one for forward and one for inverse kinematics. If using kinematics in motion control, no motion program needs to change because PMAC calls kinematics calculations internally. The purpose of forward kinematics is to calculate tool tip position from joint (motor) positions. Inverse kinematics is mathematical inversion of forward kinematics; it calculates joint (motor) positions based from the tool tip positions [5].

In hexapod application user can demand new position of the hexapod only in tool-tip coordinates. PMAC first calculates forward kinematics before a move is initiated. This is done since motors could be individually moved and the starting tool tip position would then not match the real world. Calculated values are used as an input to inverse kinematics calculations. PMAC automatically calls the inverse kinematics for each programmed move typically every 10 milliseconds.

### EPICS software

EPICS software runs on a microIOC. Since all major calculations are done inside PMAC EPICS software acts

mainly as interface to the user panels. It is used to control and monitor tool tip moves, monitor individual motor status, setup hexapod coordinates systems and scanning functionality. Figure 3 shows an example of EPICS GUI panel. Control in tool tip coordinates is in the right top corner. Lower right part is used to monitor individual motors status (limit switches, failures and velocity).

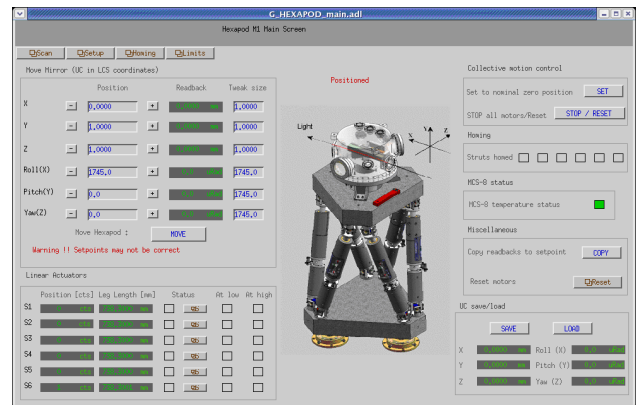


Figure 3: Hexapod user panel

Scan functionality offers user to scan hexapod position in one of the 6 tool tip axis (X, Y, Z, Y, Yaw, Pitch and Roll). User must select start position, end position, number of points in scan and dwell time in each point. For each point the software is able to obtain values from 6 different sensors. At the end of scan graph with acquired data is displayed and peak statistics is calculated.

## REFERENCES

- [1] [www.aps.anl.gov/epics/extensions/medm/index.php](http://www.aps.anl.gov/epics/extensions/medm/index.php)  
<http://ics-web1.sns.ornl.gov/edm/>
- [2] <http://www.certif.com/spec.html>
- [3] “Turbo PMAC – User manual”, Delta Tau, October 2004
- [4] R. Sabjan et al., “microIOC and EPICS: The Complete Control System Node”, PCaPAC 2005
- [5] Lung-Wen Tsai, “Robot Analysis”