Coordinating Synchronous (Real-Time) Motion Between EPICS Systems and PMAC Controllers

Curtis S. Wilson
Delta Tau Data Systems, Inc.

ICALEPCS 2013 Conference
San Francisco, California

The Path Until Now

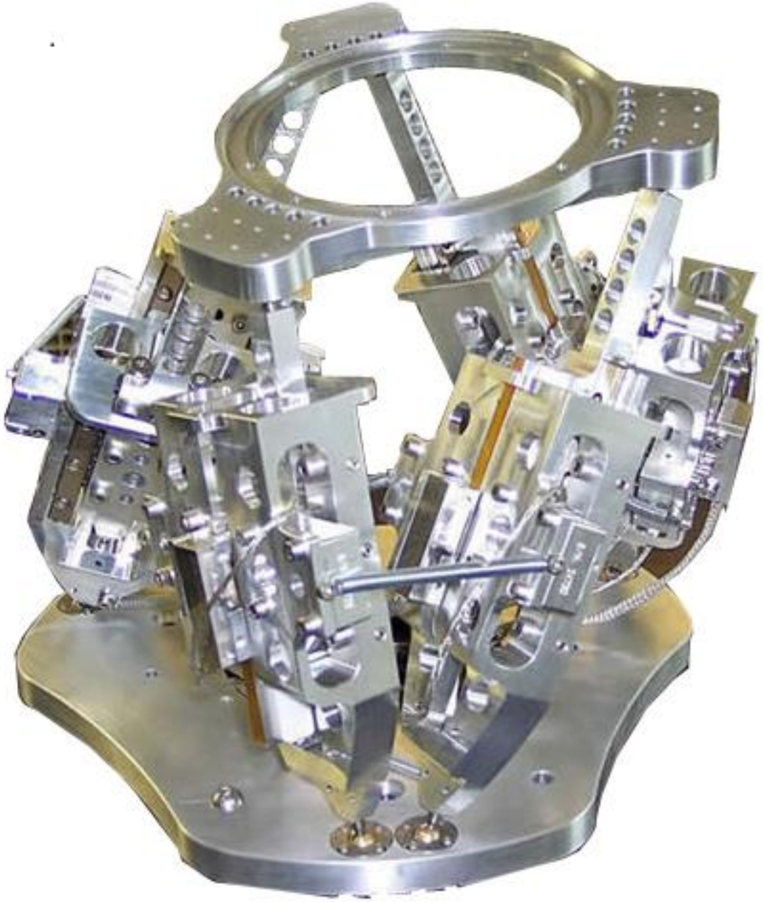
- History of different controllers from multiple vendors
- Relatively unplanned evolution
- Result: “lowest common denominator” approach to EPICS/controller interface
- Usually single-axis, single-destination (point-to-point) move
- Extensions are difficult to achieve in current framework (e.g. streaming points in real time)

Multi-Axis & Multi-Move Sequences

- Multiple axes can be grouped together into "coordinate systems"
  - Easy to command fully synchronized motion of all axes in C.S.
  - Can be abstracted to look like single-axis command
- Coordinate system can execute multi-move sequences of one or more axes
  - Sequence of pre-planned moves executed in order
  - Can be abstracted to look like single-move command

Kinematics Subroutines

- Permit specification of motion in “work” (tool-tip) coordinates, even with very different underlying actuator geometries
- Permit mathematically non-linear mapping between tool-tip and actuator coordinates
- Inverse kinematics (tip -> actuator) solved at small intervals along paths
- Kinematic subroutines can be written in Script or C



Axis Matrix Transformations

- Permit change of programming reference frame
  - Supports mathematically linear transforms
  - Scaling, offset, rotation, skew, mirroring
- Useful to reference to placement of “workpiece”

$$
\begin{bmatrix} A \\ B \\ C \\ U \\ V \\ W \\ X \\ Y \\ Z \\ AA \\ \ldots \\ TT \\ UU \\ VV \\ WW \\ XX \\ YY \\ ZZ \end{bmatrix}
=
\begin{bmatrix}
k_A & 0 & & & & & & & & & \ldots & & & & & & & 0 \\
0 & k_B & 0 & & & & & & & & \ldots & & & & & & & \\
 & 0 & k_C & 0 & 0 & 0 & & & & & \ldots & & & & & & & \\
 & & 0 & k_U & k_{UV} & k_{UW} & 0 & & & & \ldots & & & & & & & \\
 & & 0 & k_{VU} & k_V & k_{VW} & 0 & & & & \ldots & & & & & & & \\
 & & 0 & k_{WU} & k_{WV} & k_W & 0 & 0 & 0 & & \ldots & & & & & & & \\
 & & & 0 & 0 & 0 & k_X & k_{XY} & k_{XZ} & 0 & \ldots & & & & & & & \\
 & & & & & 0 & k_{YX} & k_Y & k_{YZ} & 0 & \ldots & & & & & & & \\
 & & & & & 0 & k_{ZX} & k_{ZY} & k_Z & 0 & \ldots & & & & & & & \\
 & & & & & & 0 & 0 & 0 & k_{AA} & \ldots & & & & & & & \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
 & & & & & & & & & & \ldots & k_{TT} & 0 & 0 & 0 & & & \\
 & & & & & & & & & & \ldots & 0 & k_{UU} & k_{UUVV} & k_{UUWW} & 0 & & \\
 & & & & & & & & & & \ldots & 0 & k_{VVUU} & k_{VV} & k_{VVWW} & 0 & & \\
 & & & & & & & & & & \ldots & 0 & k_{WWUU} & k_{WWVV} & k_{WW} & 0 & 0 & 0 \\
 & & & & & & & & & & \ldots & & 0 & 0 & 0 & k_{XX} & k_{XXYY} & k_{XXZZ} \\
 & & & & & & & & & & \ldots & & & & 0 & k_{YYXX} & k_{YY} & k_{YYZZ} \\
0 & & & & & & & & & & \ldots & & & & 0 & k_{ZZXX} & k_{ZZYY} & k_{ZZ}
\end{bmatrix}
\begin{bmatrix} A' \\ B' \\ C' \\ U' \\ V' \\ W' \\ X' \\ Y' \\ Z' \\ AA' \\ \ldots \\ TT' \\ UU' \\ VV' \\ WW' \\ XX' \\ YY' \\ ZZ' \end{bmatrix}
+
\begin{bmatrix} d_A \\ d_B \\ d_C \\ d_U \\ d_V \\ d_W \\ d_X \\ d_Y \\ d_Z \\ d_{AA} \\ \ldots \\ d_{TT} \\ d_{UU} \\ d_{VV} \\ d_{WW} \\ d_{XX} \\ d_{YY} \\ d_{ZZ} \end{bmatrix}
$$

Tightly Coupling Motion and Measurements

- Accurately tying digital I/O to physical position
  - "Capturing" position on digital input
  - "Comparing" position to create digital output
- Doing this while in high-speed motion
  - Accuracy is mostly product of velocity and time uncertainty
  - Great increase in throughput for given accuracy (or accuracy for given throughput) if time uncertainty can be reduced
- Increasing (effective) position measurement resolution helps accuracy

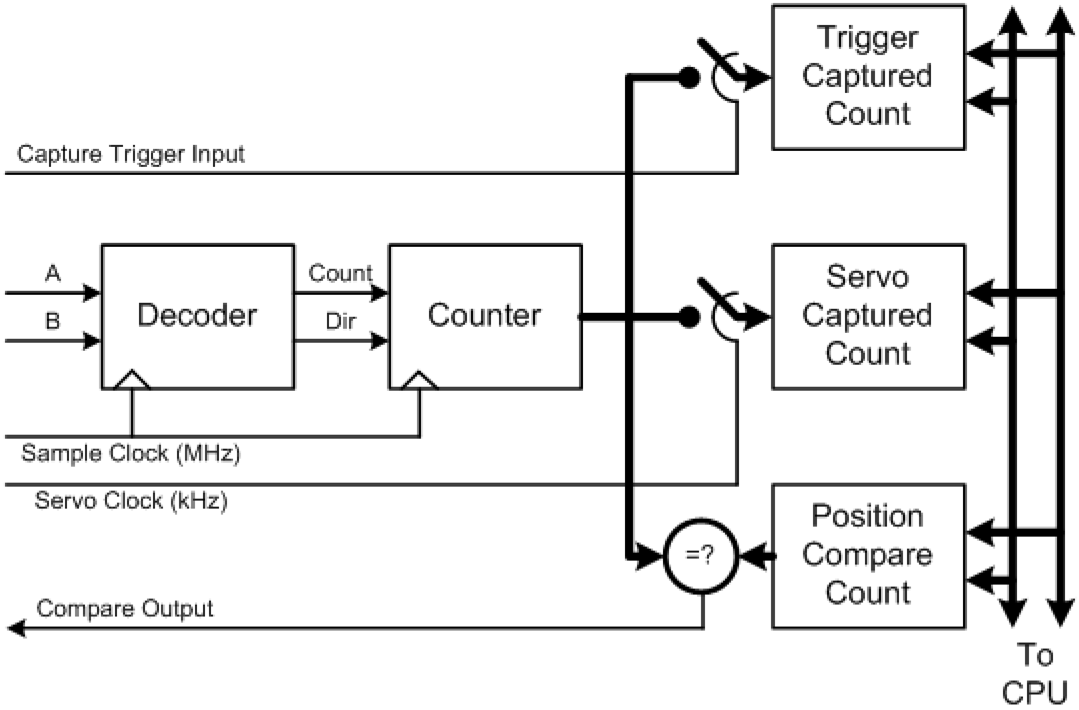
Basic Hardware Capture & Compare

At high (MHz) frequencies:

- Encoder inputs are sampled
- Counter can be incremented
- Present count can be latched on trigger
- Present count is checked against "compare" value

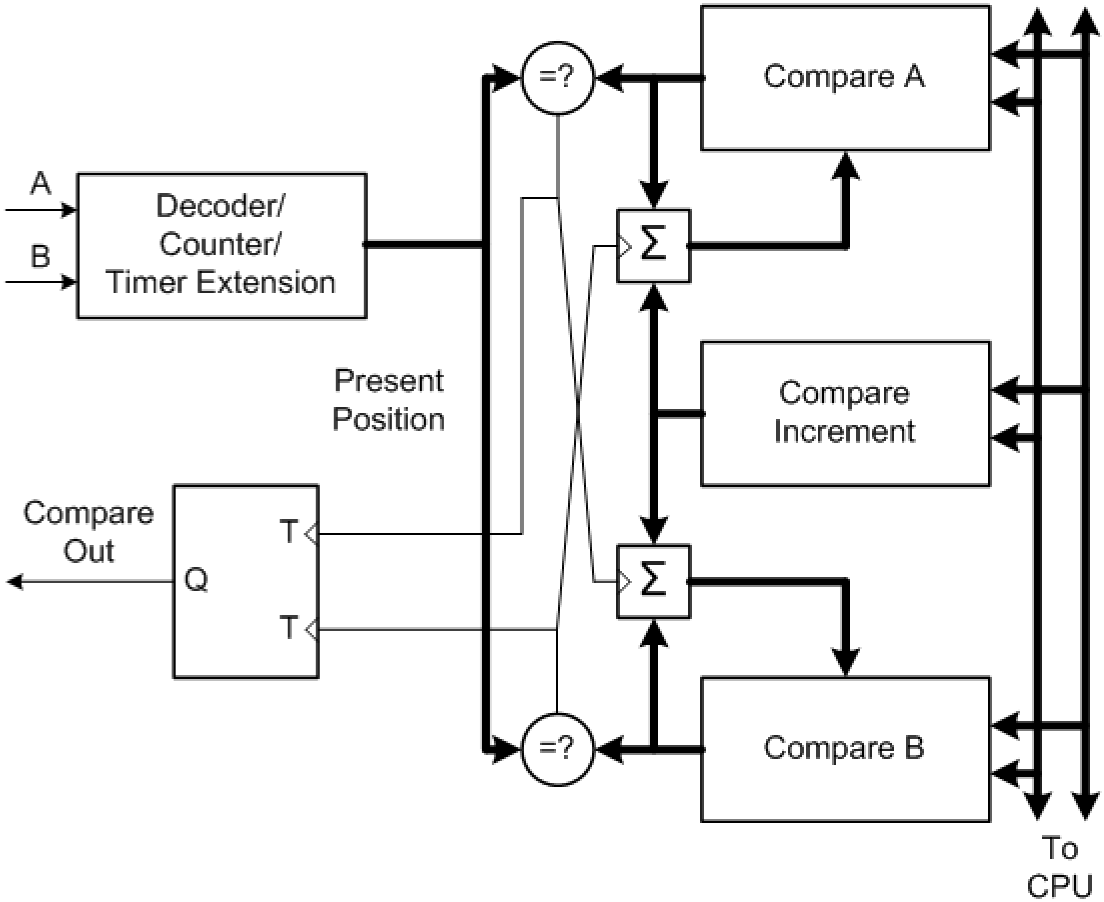
At low (kHz) frequencies:

- Counter is latched for servo use



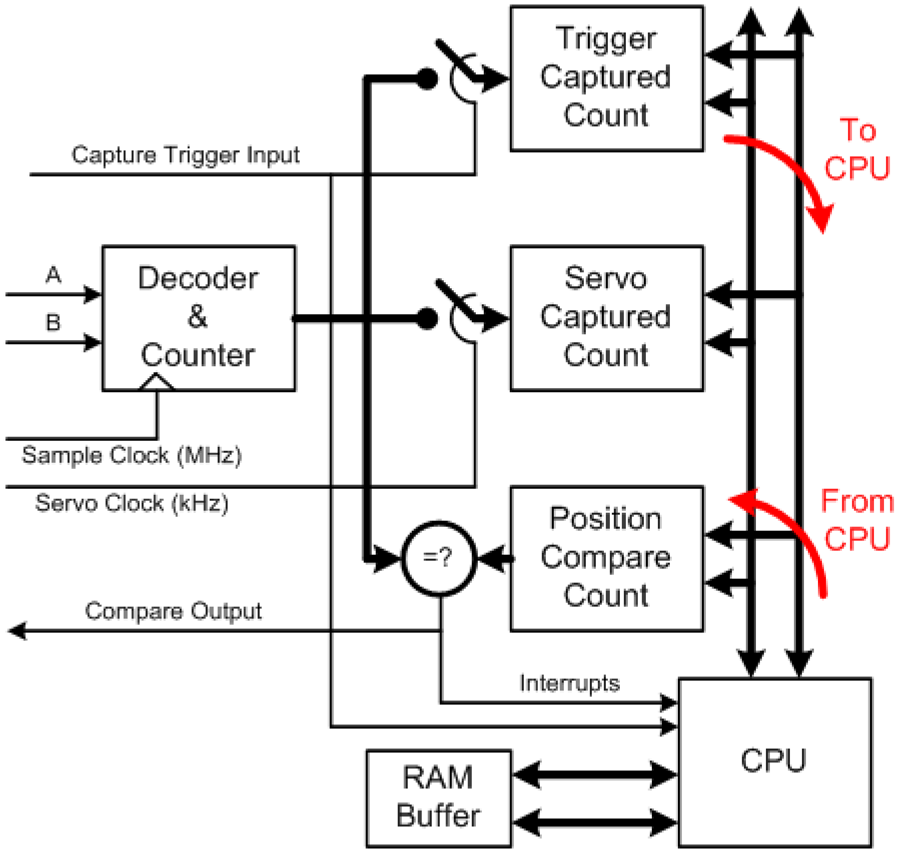
Compare Hardware Incrementing

- Two compare registers for a counter
- Each can toggle digital output
- Toggle signal also causes increment value to be added to other compare value
- Uniformly spaced pulse train can be generated at very high frequency



Capture/Compare Interrupt Update

- Capture and compare events can interrupt CPU at highest priority
- Custom ISR can react quickly to prepare for next event
- Typically store captured position to RAM buffer
- Typically load next compare position from RAM buffer
- Updates to 75 kHz
- Keep ISR short!

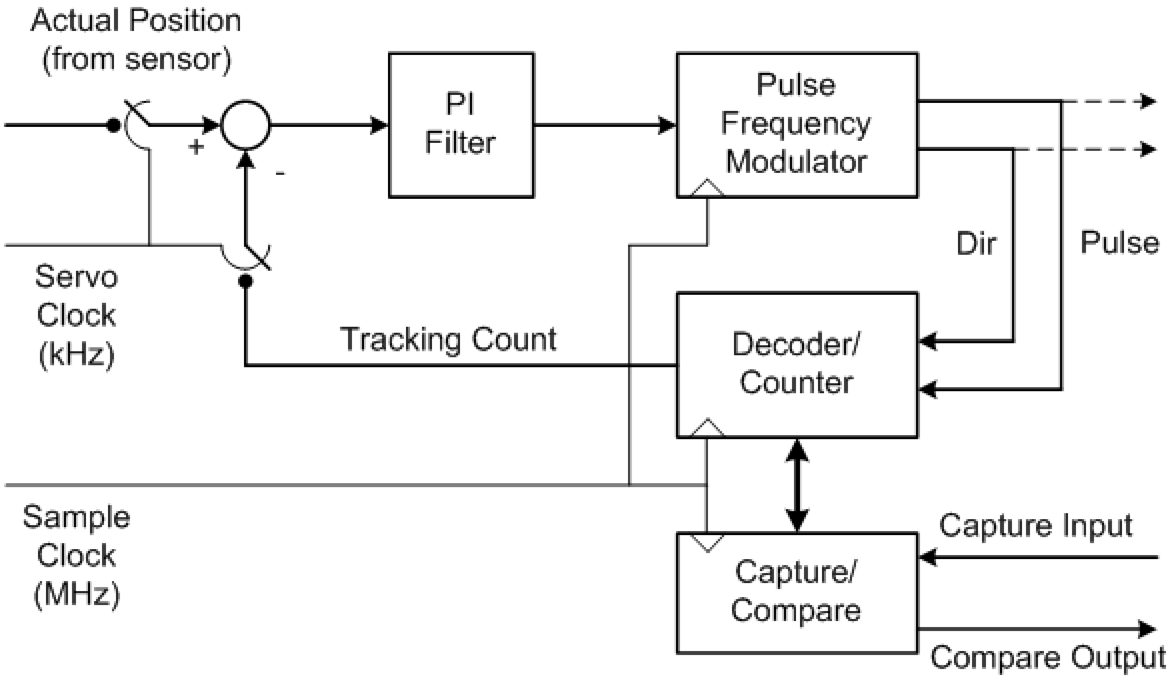


Sensors with Low-Frequency Access

- Many sensors do not support access at high enough frequency (or asynchronous access) for capture/compare purposes
  - Parallel-data interferometers
  - Serial-data encoders
  - SAR analog/digital converters
- Typically accessed at (kHz) servo rates
- To improve capture/compare accuracy, must tie to high-frequency (MHz) circuitry somehow
- Several methods can be used

Tracking Loop with Simulated Encoder

- Tracking loop reads sensor position once per servo cycle, compares it to counter value
- Error drives PFM circuit that can generate pulses at MHz rates
- Pulses feed counter
- Capture and compare use counter as for real encoder



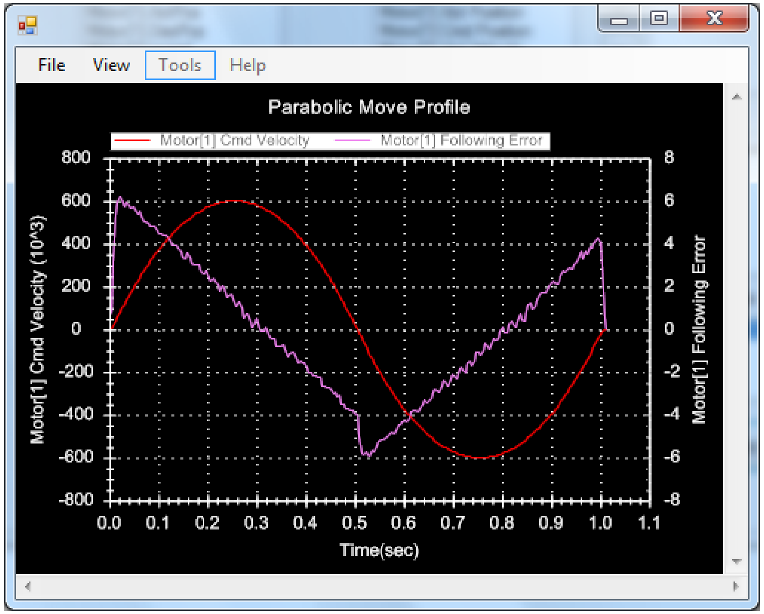
Uses of Virtual Motors

- Without physical actuator attached, have great flexibility in commanding
  - Software functions for calculating and executing move commands
  - Can use interface circuitry or not
- Under EPICS single-motor command, can act as the commanded motor
  - Executes single point-to-point move
  - Real motors can do more sophisticated sequence
  - Full coordination of real and virtual motors
- With simulated encoder circuitry, can also tie to physical I/O

DELTA TAU
Data Systems, Inc.

Synchronous Data Gathering

- Synchronous data logging at up to servo frequency
- Up to 128 memory and/or hardware registers per sample
- Key hardware registers latched on clock edges with jitter in low nanoseconds
- Capability for continuous streaming of gathered data
  - Can create real-time “scope” plot
- Easy export (in CSV format) to analysis programs



NEW IDEAS IN MOTION

Full Embedded Computer Functionality

- Power PMAC is an embedded computer with built-in machine control application
- Provides full file system and communications tool suite
- Supports user C functions
  - Custom phase and servo algorithms
  - Fast kinematics and “PLC” functionality
- Supports independent C applications
  - Can eliminate separate computer
  - Reduces communications needs
  - e.g. EPICS software