EPICS Channel Access (CA) servers (associated with Input-Output Controllers or IOCs) are well suited for interacting with sensors and actuators engaged in the control system. In particular, PSI scientists and engineers have long experience with MATLAB applications dealing with control system data. MATLAB is deployed at PSI as an integrated part of the well-established EPICS based data management system. The data flows from the sensors, which are connected to EPICS channels, is provided by two in-house developed interface packages: Java based c_matlab and C/C++ based MOCHA.CAFE. The last one, for instance, makes all basic CA channels available to MATLAB via C/C++ library, with the required interface to EPICS protocol. To get the value from a channel with a specified name, one can use the following command:

```
value = caget('channel name');
```

MATLAB codes are interpreted and executed line by line, which makes it an ideal tool for application prototyping and testing. The performance of the MATLAB programs versus their C program equivalents is, however, questionable. With all its great features, MATLAB is a good solution for “off-line” data analysis applications where a deterministic time response is not mandatory. However, to be useful for real “on-line” data processing applications, MATLAB, with its simplicity of the code development and execution, has to be combined with the C code performance and real-time data management mechanisms. The paper describes a way to implement such a combination in the context of the EPICS CA server closed loop control.

A proposal that minimizes unwanted delays for data flows in the above mentioned scenario is shown on the left. In this case, the MATLAB program is moved from the console PC directly to the IOC. This allows one to simplify inter-connections between system components and significantly increase the system performance. The basic MATLAB package has the MATLAB Coder, which produces a C code from a MATLAB program. According to MathWorks specifications, the resultant C code is functionally equivalent to the original MATLAB program.

The original MATLAB programs can be launched on selected computer platforms under Windows, Linux or Mac OS. The embedded systems, which use ARM, Power PC or MIPS processors, cannot run MATLAB codes directly but the C code generated by the MATLAB Coder can be compiled or cross-compiled for all those architectures. Imagine that the C code is generated from a MATLAB program. How to embed it into the IOC?

One possible way is to make it a part of an EPICS sequence (SNL) program. Another way is to use a standard Array Subroutine (aSub) EPICS record. In this case, the input (inp) links of the aSub record are associated with EPICS channels, which are used in the original MATLAB program in the context of caget calls. The output (out) links of this record replace caput calls. The C code generated by the MATLAB Coder is encapsulated by the process function of the record.

To realize these ideas, an automated tool for converting MATLAB based controls algorithms into C code was created. The tool is called the MATLAB Coder to C Controls (MCCCT). Its core consists of three Linux shell scripts. One script (coderPreprocessor.sh) does the MATLAB program pre-processing. It acts on the original MATLAB program and prepares it for the C coder operation. The main goal is to replace all caget and caput calls with the entries required by the aSub record or SNL program. In addition, this script determines the type and number of elements for EPICS channels used by the original MATLAB program.

Two other scripts do the post-processing of the C code generated by the MATLAB Coder. The first post-processing script (coderPostprocessor.sh), converts the generated code into a suitable aSub record process function. The script also generates the EPICS database associated with the aSub record, which has input and output links filled with all channels involved,

- generates a standard IOC startup script to load the aSub record related database, which was generated in the previous step.
- creates a makefile to compile/cross-compile the C code, and
- produces all header files and C files, which are required for the final compilation/cross-compilation.

The second post-processing script (coderPostprocessor_snl.sh) inserts the C generated code into a standard SNL skeleton provided by the MCCCT. The script also
- creates a makefile to compile the produced SNL program,
- generates a standard IOC startup script that loads a shared library associated with the resultant SNL program, and
- produces a standard IOC SNL startup script that launches the SNL program.

The MCCCT solution looks very attractive for MATLAB/EPICS developments. The original MATLAB code, which is easy to test and maintain, can be used for prototyping, while the generated C code can be used as a production version to be embedded in the dedicated EPICS CA server. We note that the SNL part of the MCCCT solution is especially efficient for MATLAB programs doing data processing and control actions in infinite loops.

Some more MCCCT details can be found in the Appendix.