Rapid Control Prototyping (RCP)

Rapid Control Prototyping or RCP is a process which lets the engineer quickly test and iterate their control strategies on a real-time computer with real input/output devices. Rapid control prototyping is a variant of hardware-in-the-loop (HIL), but is distinct and popular enough to warrant its own name (and acronym). Rapid control prototyping differs from HIL in that the control strategy is simulated in real-time and the "plant," or system under control, is real. See Figure 1.

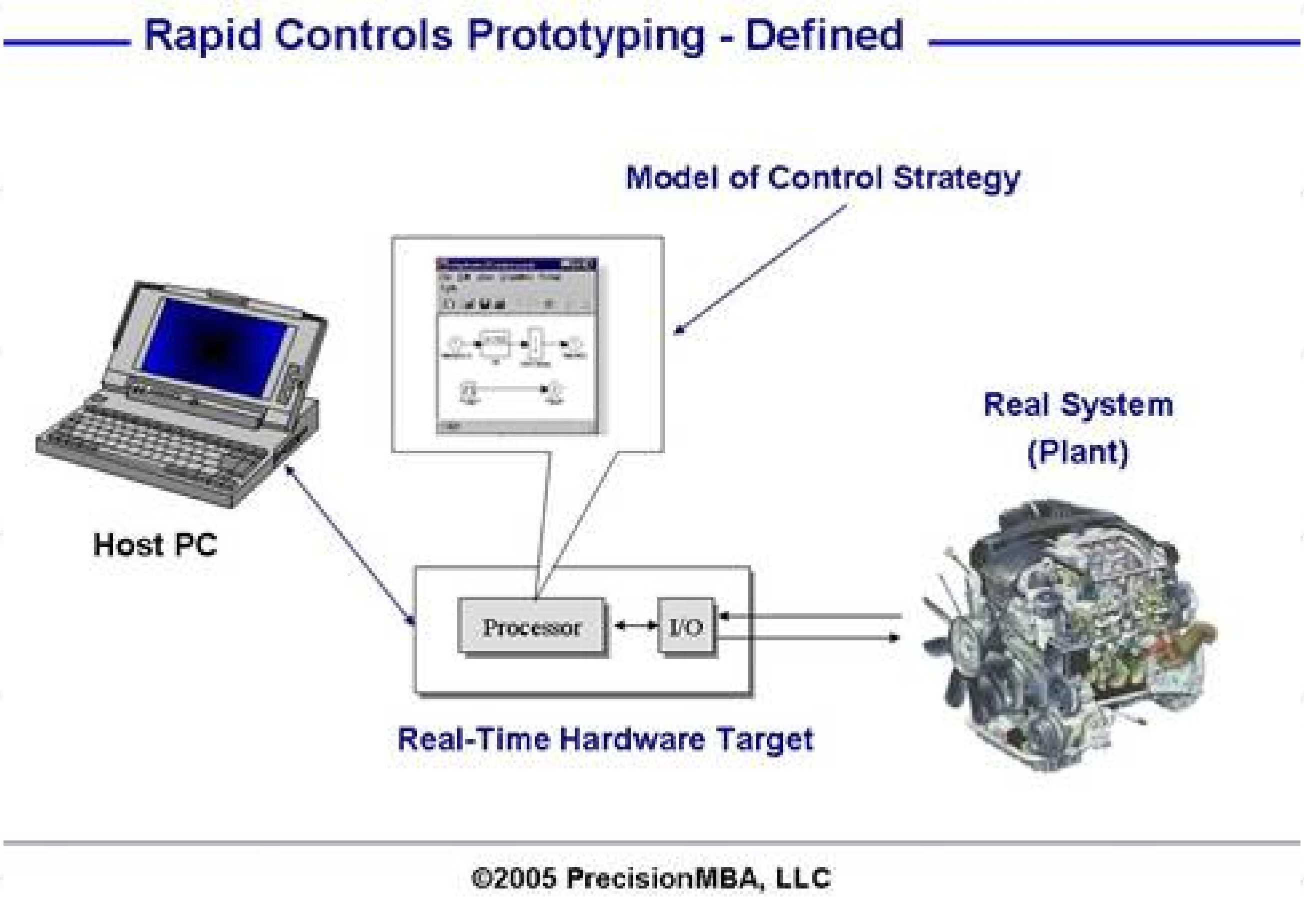

Figure 1. Rapid control prototyping definition: Control algorithm is simulated and plant is real.

Rapid control prototyping exploded onto the US automotive market in the mid-1990s. It was a welcome solution for managing the increasing complexity of the control engineer's task. Rapid control prototyping is now the typical method used by engineers to develop and test their control strategies. Rapid control prototyping was first used for developing power-train control strategies. The simple reason is that is where the software was – in the engine control and transmission control unit. It has since been adopted industry wide in applications such as anti-lock braking, anti-roll, vehicle stability, active cruise control, and torque distribution.

Rapid control prototyping is growing in other industries as well. The aerospace industry is second in size and is growing in the application of this technology. Applications such as Full Authority Digital Electronic Controller (FADEC) strategy development for jet engines, active braking systems, flight control systems, navigation systems, Un-manned Aerial Vehicles (UAVs), and tracking systems. Rapid control prototyping is also used in medical device development and in industrial automation applications such as servo-control. Low cost rapid control prototyping systems are being used as teaching tools in most university graduate-level courses in classical and modern control theory.

The typical rapid control prototyping system is comprised of the following components (see Figure 2):

1. A math modeling program such as Simulink™.
2. A symbolic I/O blockset for Simulink™. This is sometimes called the real-time development environment.
3. A real-time target computer. These are typically embedded computers with I/O such as analog, digital, and serial.
4. A host PC with communications link to target computer.
5. A Graphical User Interface (GUI) application to download and control the real-time process.

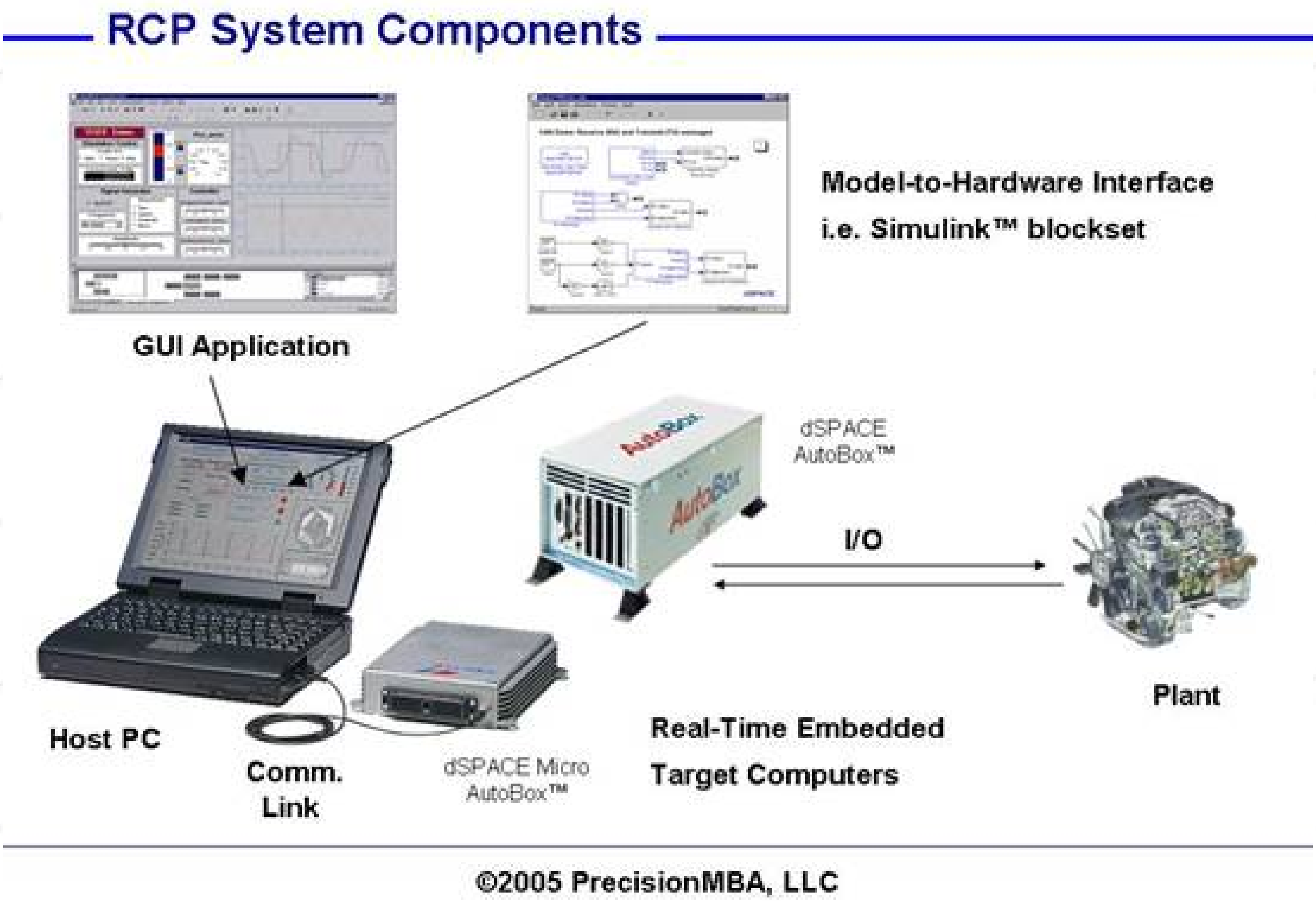

Figure 2: Typical rapid control prototyping system components.

So why is rapid control prototyping so popular? The reason is that rapid control prototyping algorithms are developed as symbolic models, not as C-code. Rapid control prototyping algorithms are designed using a popular math modeling package such as The Mathworks, Inc. Simulink™ product. The controls engineer can concentrate on developing his/her strategy in a familiar modeling environment and does not have to worry about translating the model to C-code. This is very significant as most control engineers are not C-code experts nor do they typically have the skills to "port" C-code to a real-time target. By virtue of an automated "build" process, the rapid control prototyping system does this work for you.

The typical development procedure goes like this (see Figure 3):

1. Control strategy is developed using Simulink™ symbolic math modeling application.
2. Symbolic input/output (I/O) blocks are imported into the math model and

"wired" to the appropriate points. This is essentially a symbolic way to add input/output capability to your control strategy.

3. The "build" process is invoked. This process does the following:
    a. The model is "read" and an automatic code generator renders compile-able "C-code."
    b. The C-code is "cross" compiled and linked with target specific code such as a scheduler, I/O routines, and communication routines specific to the real-time target.
    c. The end result is an executable program for the target computer.
4. Using a Graphical User Interface (GUI) program the executable is downloaded to the target. The program can now be controlled and instrumented by the GUI program. This is referred to as experiment control.

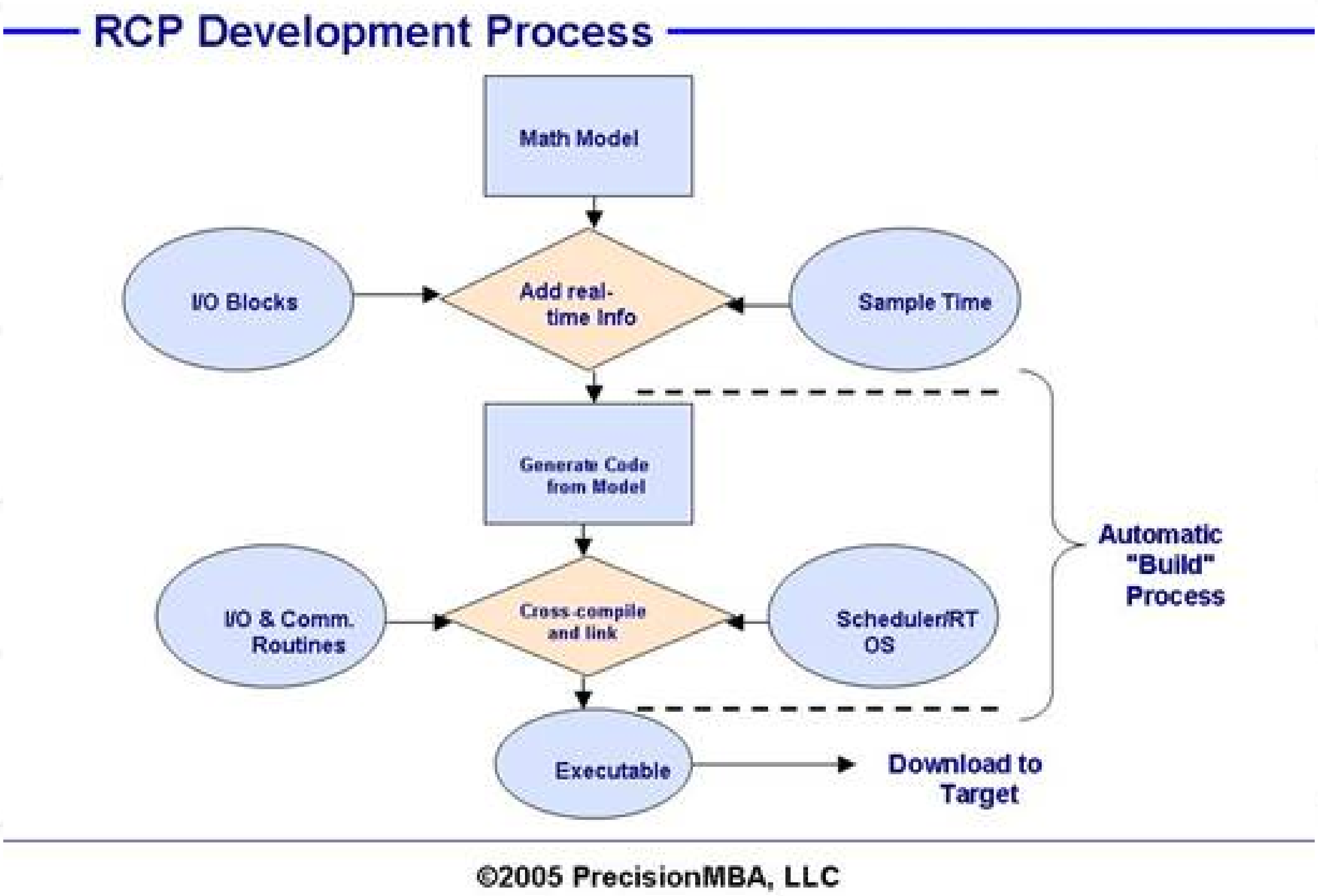

Figure 3: Rapid control prototyping development process

The GUI application is the user's window into the real-time operation of the control strategy. Variables can be monitored, graphed, or logged. The strategy can be altered only by "tuning" gains – the structure of the strategy cannot be changed. For example, if you had a PI controller and you wanted a PID controller you would have to change the model and repeat the build process.

The user of a rapid control prototyping system will quickly find that they are more productive in their convergence on the best control strategy. This convenience and productivity is not without cost. A typical rapid control prototyping system including math modeling software, GUI, and target hardware is typically around $30-45,000.00 depending on the vendor. This entry level price may cause some sticker shock, but the payback period is short due to gains in engineering productivity and performance. There are ways to economize depending on the final usage methodology.

If you are considering rapid control prototyping and need advice, please contact us.

Home