

Inside Machines: PC versus PLC: Comparing control options

To choose between a PLC or PC, analyze and compare characteristics that could differentiate the two technologies, such as operation, robustness, serviceability, hardware integration, security, safety, programming, and cost. Graphics illustrate some key considerations.

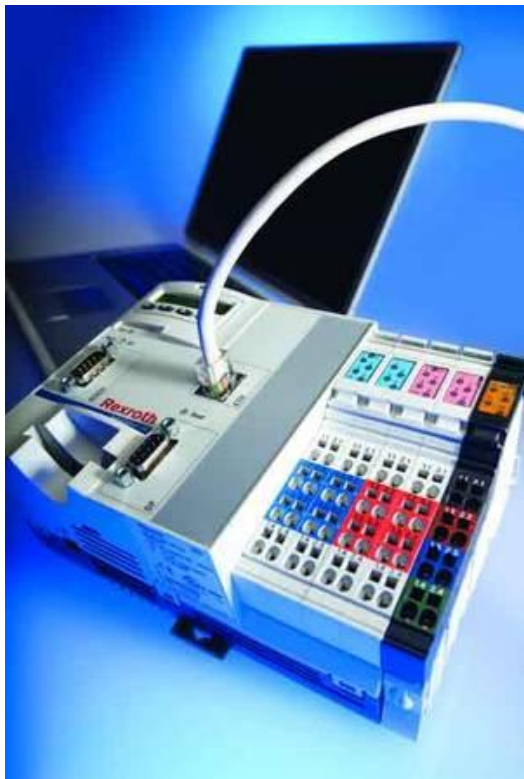
Phillip Lipson, Geert van der Zalm, Bosch Rexroth Corp.
05/11/2011



One of the most crucial decisions in the initial design phase of a machine is the selection of the control system. Since programmable logic controllers (PLCs) were first introduced in the 1970s, they have dominated the process and automation markets. For years, the PLC has led the way for advances in machine automation control. From small controls used in the automotive industry to large-scale controllers running entire factories, the PLC was the primary controller of choice.

However, starting from the early 1990s, the PC has been successfully working its way into those markets as processor speed and range increases, and the cost of those components decreases. The number of applications using a PC is on the rise, causing an accelerated development that blurs the line between

the two technologies. Ten years ago, deciding what type of controller to use might have been a clear choice; today, that is no longer the case.



The PLC was developed as a more streamlined, flexible, and reliable alternative to switch boxes and relay panels. It was dedicated only to specific tasks in the factory, and its language and structure were modeled around the switches and relay panel circuits it was replacing. Furthermore, it had to maintain its robustness and consistent performance in challenging environments that contained relatively high levels of electromagnetic interference (EMI), contamination, and vibration. As time passed, the PLC evolved to include the capabilities of motion control, advanced proportional-integral-derivative PID process control, and integrated safety, while also adopting some PC features, such as a web server and networking utilities.

The PC, on the other hand, served a higher level role in the scope of the machine. It was used primarily for complex calculations, monitoring, measuring, and factory networking as well as a user-interface to the PLC. It was usually housed in a more environmentally controlled location because it couldn't operate as reliably as a PLC in harsh factory conditions.

The PC eventually evolved to include PLC functionality, while still containing its core capabilities. Also, the PC has become a more robust controller,

allowing it to operate in the harsh environments where once only PLCs could operate. Further convergence of the technologies is evident in an analysis of their control architectures. With the addition of a PCI card, hardware drivers, and software, a PC can serve as a PLC. Furthermore, the addition of a real-time kernel can enable the PC to support more critical tasking and control algorithms. On the other hand, there are PLCs with a built-in PC that only require a keyboard and mouse to get started.

So, how does one choose between PLC or PC-based architecture? To simplify the decision-making process, it is important to first analyze and compare differentiating characteristics. Seven major areas to consider are:

- Operation
- Robustness
- Serviceability
- Hardware integration
- Security
- Safety
- Programming

Operation

When analyzing system operation, focus on how the system will run and how instructions and tasks are processed. The standard PLC has an embedded real-time operating system (RTOS) with a dedicated processor that ensures a high degree of control system reliability. Furthermore, since the PLC handles only automation and/or a process, it does not need to run other utilities, such as antivirus programs or system updates.

A PC with a real-time kernel or real-time operating system (OS) can accomplish the same reliability of control as a PLC. From earlier experiences with PCs at home and in the office, users might be wary of lock-ups of the PC (the so-called “blue screens”). However, these lock-ups could occur on any OS, including a PLC, if the OS software is not handling priorities properly. For industrial use, the software running the PC is dedicated to automation and will therefore have a minimal chance of lock-ups. Even if a lock-up would occur, the real-time kernel is not affected and continues operation.

Real-time operation is a relative concept that means any task is guaranteed to be handled within a certain time. Synchronized motion and/or advanced PID control requires a high level of real-time determinism, while noncritical

supervisory controller operations, such as monitoring error messages or sending noncritical controller commands or queries, would not.

Robustness

Robustness of the controller refers to its durability in various environments. The standard off-the-shelf PLC has no moving parts, so it can withstand harsh environments for millions of cycles. A standard PC contains moving parts, such as fans or hard disk drives, and is less suitable for environments with high vibration levels.

However, industrial PCs (IPCs) offer options such as solid-state drives, fan-less cooling, and in-cabinet mounting. These options make a PC just as durable as a PLC, able to withstand the toughest industrial or environmental conditions. PLCs and PCs have converged in this area, but the PC requires additional options to equal a standard PLC.

Serviceability

Another factor to consider is the ease and cost of serviceability, which can be measured by the repair and replacement costs over the life of the controller. For a PLC, external devices can be replaced with ease while the system is in operation. Moreover, the PLC's compact modular design makes replacing the actual controller an easy job. This saves cost because it reduces machine downtime. It's also possible to perform a hot swap with a PC, but only for USB or other external peripheral devices. If the PC has a more modular design, such as a rack or panel-mount system, replacement time is closer to that of a PLC.

It is helpful to be able to effortlessly change out a system or its components and have a resource pool of replacement parts with long-term availability. In some industries, "copy exact" policies require this long-term availability. It is easier to implement copy-exact with a PLC since the hardware and firmware don't change as rapidly as for a PC. Trying to find parts for a PC (even after a year or two) can be more challenging than for a PLC. (Ask the vendor about such policies, availabilities, and costs.)

Hardware integration

Every engineer appreciates a wide choice of options when selecting control system hardware, since there is always a need for items such as additional peripherals, memory, and a user interface. Both the PLC and PC have the

ability to control a multitude of devices using industrial communication networks. Some of the well-known networks are SERCOS, Profibus, DeviceNet, and CANbus, as well as their Ethernet-based counterparts such as SERCOS III, Profinet, EtherNet/IP, and EtherCAT.

Although the PLC and PC can offer an array of fieldbus options, the PLC has many of these options built in, whereas PCs and some IPCs need additional cards and drivers to provide a comparable offering. However, besides the typical fieldbus networks, the PC is equipped with a more open and flexible array of interfaces, such as USB, FireWire, serial, wireless Ethernet, etc.

This gives the user access to more off-the-shelf devices to handle tasks that a PLC usually could not handle, such as an advanced high-resolution imaging system, where the images would be stored, analyzed, compared, and possibly archived. A PC would be well suited for this advanced task because of the amount of memory required; a PLC would have limited storage and processing capabilities.

For some applications, the user interface is a crucial machine component. The PC has a built-in user interface, while the PLC would need one or all of the following for primary operation: switches, operator panel(s), or an industrial PC. So while the PLC can interface with devices over fieldbus and perform complex operations, it still needs a PC to handle memory-demanding tasks and make connections to other devices in the system.

Security

Security mainly deals with protecting the file system and application. This has two aspects: preventing unauthorized access from the outside world (such as virus attacks, malware, etc.) and limiting user access (such as restricting user rights, hiding files, etc.).

Traditionally, a PLC is less exposed to unauthorized access from the outside world. Because of its dedicated OS, there are very few known instances of virus attacks on PLCs. However, this does not mean that PLCs are immune to viruses. Since PLCs have enjoyed a virus-free status for years, there are no standard ways of detecting and eliminating a virus if it were to occur. Although the PC is far more susceptible to virus attacks than a PLC, strict security measures can be invoked to significantly reduce potential threats, and standard software is available to detect and remove viruses. PLCs and PCs offer different levels of user access to keep the contents as secure or as open as desired.

Safety

Depending on the operating environment, safety can be a major concern for the user. It is vital in cases where human interaction with the machine could be potentially dangerous and when safety standards in a semiconductor fabrication facility or other factory are crucial to machine acceptance.

The PLC has a long history in machine automation, designated communication channels to slave devices to tightly monitor operation, and optional integrated circuits for redundancies. Integrated safety has only recently become available on some PC-based platforms.

Programming

Device functionality is only as good as the program running on it. Therefore, the programming environment and language are crucial to optimal machine performance. One main difference between PLC- and PC-based solutions is how the code is executed. A PLC mixes scan-based and event-driven program execution, whereas PC software is typically event-driven. The scan-based execution of a PLC program might take longer because the system needs to complete the higher priority actions in the cycle first. The difference in execution style requires a different programming philosophy, and often users are committed to either one or the other.

The same holds for the programming language: PLCs are programmed using languages specified in IEC 61131-3 (ladder logic, instruction list, etc.), or proprietary vendor languages. PC-based controls can use programming languages such as C/C++/.NET.

Cost discussions

After analyzing the technical requirements and comparing the design options for your machine, one more topic needs to be considered: cost. Many applications can be solved by either a PC-based or PLC-based solution, but for some applications, the cost of those solutions can differ widely. A comparison of the relative cost of PC- and PLC-based architectures centers on four areas: performance, expandability, environment, and development effort.

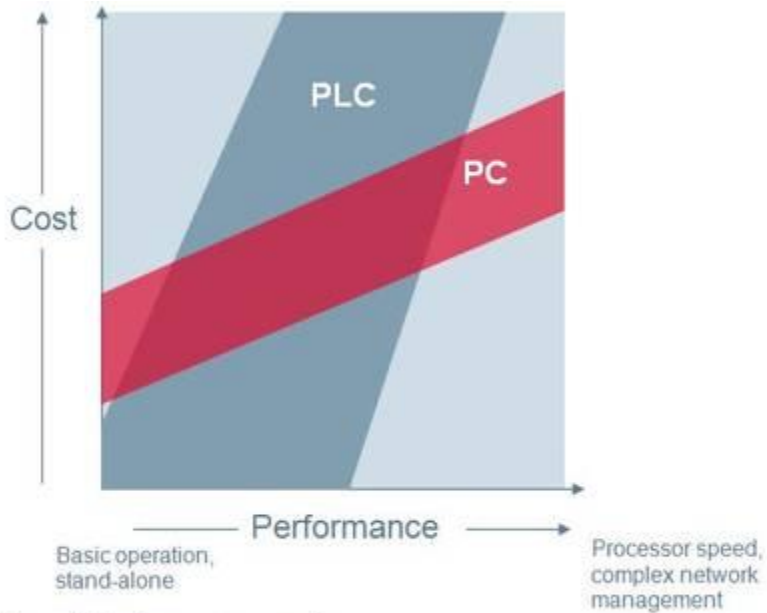


Figure 1: Performance vs. cost

When a control system must fulfill demanding calculations, handle complex network loads, and process a lot of data, a PC-based solution may be more cost-effective. The initial cost of a PC might be higher, but it provides a very powerful system and the cost only increases incrementally when more **performance** is required. PLCs, however, typically start at a lower price point and can get more expensive than a PC when high performance is needed (Figure 1).

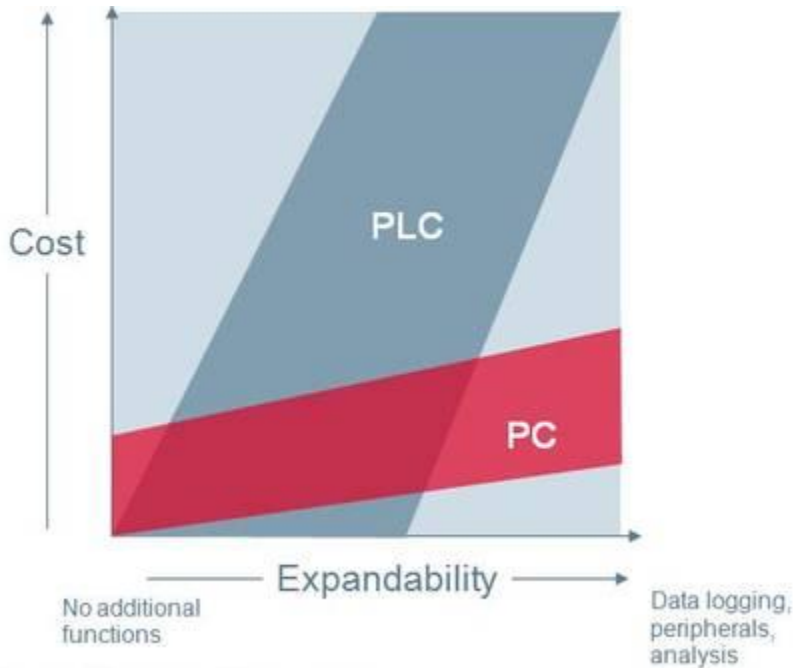


Figure 2: Expandability vs. cost

Expandability is the capability of your control system to add peripherals, increase calculation power, or add data storage. When no additional functions are needed and expandability is low, PCs tend to have a higher cost than PLCs, but PC cost does not increase much when increasing expandability. The cost of a PLC is typically lower for applications where these items are not important because of the optimized nature of PLCs, but the cost ramps up steeply when a lot of additional devices are needed (Figure 2).

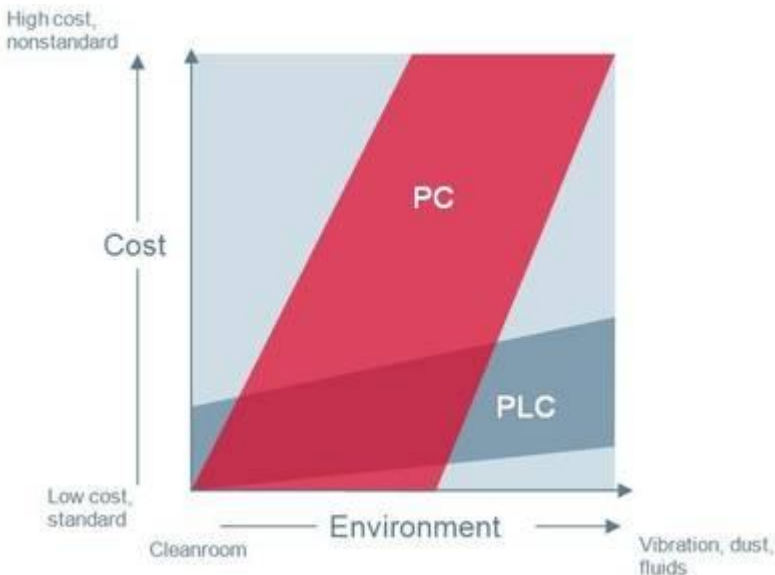


Figure 3: Operating environment vs. cost

The standard PLC was designed to withstand the harsh **operating environment** in an industrial factory setting. It is no surprise that the standard PLC costs much less to operate in rough environments compared to the PC. Adding components to make a PC more robust significantly increases the cost (Figure 3).

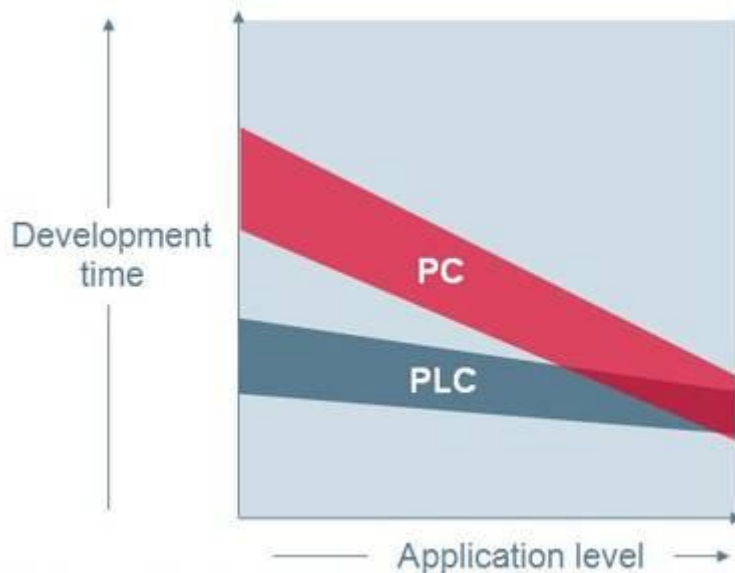


Figure 4: Development time and application level vs. cost

Software **development time** can add significant cost to a machine design project, but it often remains hidden when initially selecting the control system. If this is the first time the particular control system is applied, one can either take the time to learn the new product or hire a third party with knowledge of the product to help integrate the new system.

With a PLC that uses a common format such as IEC61131-3, chances are that even if one is not familiar with the language used, there is a large resource pool of those who are. Moreover, this platform has several built-in functions that are ready to integrate and solve certain problems quickly. Therefore, the PLC can sometimes be a less expensive option than a PC-based controller for those who are first-time or intermediate users: the development time (or cost) decreases as the level and knowledge base of the user increases (Figure 4).

With extensive expertise and intimate knowledge of the system, there will be little or no cost difference between a PC- or PLC-based solution.

- Phillip Lipson and Geert van der Zalm are applications engineers at Bosch Rexroth Corp., Hoffman Estates, IL.

Source:

<http://www.controleng.com/single-article/inside-machines-pc-versus-plc-comparing-control-options/9bf8690c6f23b11370bec90b52cb15c9.html>