

Control Strategies for Web Handing

William B. Gilbert
Converting Solutions
Siemens Industry, Inc.
5300 Triangle Parkway, Suite 100
Norcross, GA 30092
Phone: 678.314.4222
Fax: 770.871.3888
Email: William.Gilbert@Siemens.com
Website: <http://www.usa.siemens.com/converting>

Table of Contents

<i>Abstract</i>	3
<i>Overview / Criteria</i>	3
Usability	3
Performance	4
<i>PLC Based Control</i>	5
PLC Control Functionality	5
PLC Control Architecture	5
<i>Drive Based Control</i>	6
Drive Control Functionality	6
Drive Control Architecture	6
<i>Motion Based Control</i>	7
Motion Control Functionality	7
Motion Control Architecture	7
<i>Technology Trends</i>	8
Safety	8
Remote Access & Diagnostics	9
<i>Conclusions</i>	9
<i>References</i>	9

Abstract

There are several architectural strategies that can be considered for web handling drive system controls. Current industrial technologies permit the web handling controls to be implemented in either a Programmable Logic Controller (PLC) (typically the same as the machine control), directly in the drive system, or through a motion controller.

PLC based web control has long been a traditional choice for machine builders for a number of reasons. The PLC provides a single platform for both automation and drive control with a centralized control structure. PLC based systems offer a suitable level of usability, however they can be limited in high end performance capability and in their options for process level programming.

Drive based control typically offers distributed control architecture, peer to peer networks and an increased level of performance due to faster processing times. Graphical engineering tools are common for drive based systems and are a preferred programming environment due to their ability to visualize and document the web control processes.

Motion controllers offer the highest level of performance and functional flexibility. Their inherent capability of providing position data can help increase web handling performance on several fronts. Motion controllers also permit the line integration of axis motion functionality such as positioning, electronic gearing, and cam functionality in the common web controller. They are not limited by memory constraints and typically utilize the full range of programming languages.

This paper will review the merits of the three control architecture strategies under the criteria of usability, functionality and performance. It will also give a brief overview of new technologies that simplify and enhance machine safety, system access and diagnostics.

Overview / Criteria

Usability

Usability defines the control system's ease of use in the areas of engineering, commissioning, and maintenance. The following points apply to each of the control system options, PLC, Drive based and Motion Control.

A common engineering tool utilizing a common database for machine and drive control is recommended. Individual engineering tools for each controller (PLC, Drives, etc.) should be avoided. The engineering and programming connection to the system should be through a single point with efficient routing to each drive or controller location in the system. Additionally multi-user editing is an important feature for complex and large projects.

The programming language used for the web control should be considered for usability. The programming language should be sufficient for implementing the critical tasks, easy to use and understand. We find that the ideal programming language for the web control or drive processes to be graphical function chart. Web handling control is a process and a graphical programming editor offers the most efficient method to develop, visualize, support the process and produce the system documentation.

The engineering platform should offer efficient and common diagnostic and troubleshooting tools that include integrated online monitoring capability, time and frequency based trace tools and a drive axis commissioning control panel.

Control and drive hardware platforms that store programs on removable media are ideal. The compact flash cards permit easy swapping of hardware without the requirement of program or parameter file downloading and retain current machine settings.

Integrated or "canned" web handling functions such as sectional tension controls, center-winders, rotary knife, and flying saw, etc. decrease engineering time, simplify commissioning and maintenance. Ideally the source code of these functions is open to allow customization of the functionality.

Performance

The performance of any machine or system will be directly related to the controller’s actual process throughput time. Process throughput is the time that it takes the control to receive feedback (from the motors and tension sensors), scan and process the control loops and send the respective commands to the individual drives.

Three components contribute to the process throughput time; the controller scan time or process cycle, the communication network cycle time and the drive loop cycle times. In systems that the controller and network cycles run independently or “asynchronous”, the process throughput times can vary significantly due to the timing of the feedback signals, process loops and reception of the transmitted setpoints.

Process throughput is optimized when all of the individual scan cycles are executed synchronously. In this case of a synchronous system, all of the data will be processed and transmitted in a single clock cycle, making the process control truly deterministic.

Figure 1: Asynchronous Scan Cycles

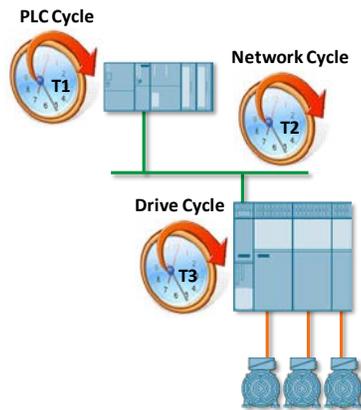
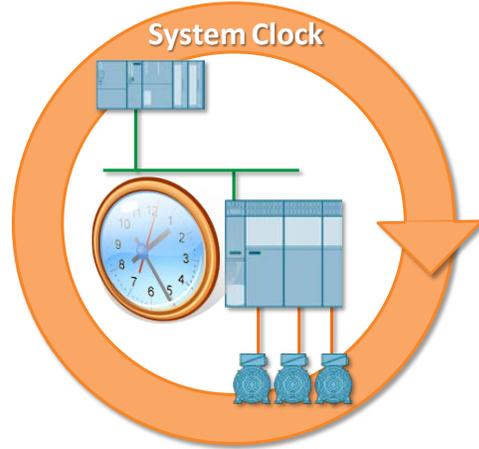


Figure 2: Synchronous Scan Cycle



Because of the signal timing uncertainty in **asynchronous** networked systems, the process throughput must be calculated as the worst case;

$$\text{Process Throughput} = (\text{Tension control cycle} * 2) + (\text{Network cycle} * 2) + (\text{Drive Cycle Time} * 2)$$

Process throughput in control systems that are **synchronously** connected, is calculated as;

$$\text{Process Throughput} = (\text{Tension control cycle} * 2) + \text{Drive Cycle Time}$$

Note: The calculation assumes that the data is read in one cycle, processed and transmitted next synchronous cycle.

Each system component will have different ranges of system cycle or scan time performance. See table 1 for individual component scan time examples.

Table 1: Reference scan times for individual controller and device options

Device	Common Cycle Times	Notes:
PLC	~5 – 100 ms	Depending on Program size and processor power
Drive - (program)	~1 – 8 ms	Configurable program cycle time
Drive - (loops)	~62.5 – 250 us	Normally fixed
Motion Control	~0.25 - 8 ms	Configurable program or background task cycle times
Network	~0.25 – 8 ms	Normally configurable

The optimum process throughput period for each machine will depend on the web speed, machine mechanics, acceleration times and dynamic requirements. Note that this time must include all of the tension controlled drives in the system.

A distributed control scheme (with multiple processors) is theoretically better equipped to close individual loops at faster speeds. Centralized systems with single processors require greater processing power as the number of control loops increase but have the advantage in coordinating setpoints from a single source. “Average” process throughput time range for tension loops could be from single digit ms to mid double digit ms range.

PLC Based Control

PLC Control Functionality

Programmable Logic Controllers (PLC's) are the standard for implementing industrial machine control and automation. They have long been utilized for implementing drive web control and are very popular with machine builders.

PLC based systems provide a high level of usability through a centralized control structure when applied as web control systems. Overall usability will depend on the level that the drive system is integrated into the PLC engineering environment. The processor performance and memory capability are normally scalable; that is as the system size grows it can be just a matter of scaling up to the next level processor and or memory.

Ladder logic is the original and most common primary programming language of PLC's. Since the PLC was developed to replace relay logic control systems, it was only natural that the initial language closely resembles the diagrams used to document the relay logic. Ladder logic is extremely competent in the programming of discrete and analog I/O (inputs/outputs) but less than ideal for the usability requirements of the process functionality for web handling control. Most PLC engineering software will include several programming languages. The most common include, Ladder diagram (LAD), Function chart diagram (FBD), Structured text (ST), Instruction list (IL), Sequential function chart (SFC). Graphical function chart programming may be a possible option, however it is not common.

The PLC is very suitable for low to mid-range performance systems, as axis count and performance requirements increase it may be challenging for the PLC to match the performance of drive based and motion based systems, especially if the PLC does not utilize isochronous communications. Other performance pitfalls can be incurred when there are high speed access requirements for the system inputs and outputs, i.e. tension sensors and other feedback communications. The use of multiple PLC's often leads to issues of dead time compensation between machine sections.

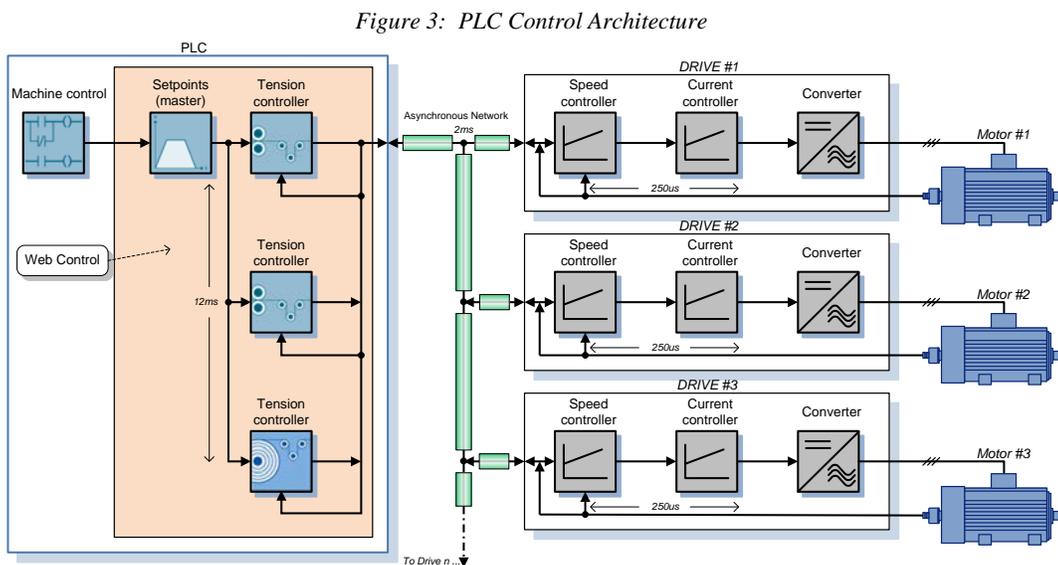
PLC Control Architecture

Figure 3 depicts a centralized control architecture with the machine, drive and web control programs implemented in a common PLC. A common network interfaces the PLC and its control programming to the individually controlled drives. The depicted system has a cycle time of 12ms for all of the web control loops, an asynchronous network with a 2ms cycle time. In this example, The web control, network scan and drive cycle are not synchronized.

The calculated process throughput for this example is as follows;

$$\text{Process Throughput} = (\text{Tension control cycle} * 2) + (\text{Network cycle} * 2) + (\text{Drive Cycle Time} * 2) \\ (12\text{ms} * 2) + (2\text{ms} * 2) + (250\mu\text{s} * 2) = 28.5\text{ms}$$

Performance Reference: In this example, a web traveling at 600mpm, will displace 2.85cm before a new correction is made.



Drive Based Control

Drive Control Functionality

Drive based control architecture is very common for web handling applications. It offers a high level of performance due to its distributed or decentralized control architecture and peer to peer communication. A peer to peer network is a dedicated drive based network transmitting data among the system drives in a typically synchronous manner. Processing speeds in the drive are typically much faster than in the PLC. High speed local I/O can offer direct responses to the control system and high speed position capturing.

Drive based control programming normally utilizes a graphical editor as the engineering tool. Graphical function chart editors provide a good visualization of the process during the engineering and easy to read documentation of the programs. Standard functions for converting and web handling are also common for drive based systems. Other programming language options may be somewhat limited at the drive level and utilizing graphic function chart for logic and I/O is not as convenient. However this is not a normal requirement at the drive level. Integrated drive commissioning and tuning tools are normally standard in the drive engineering tools.

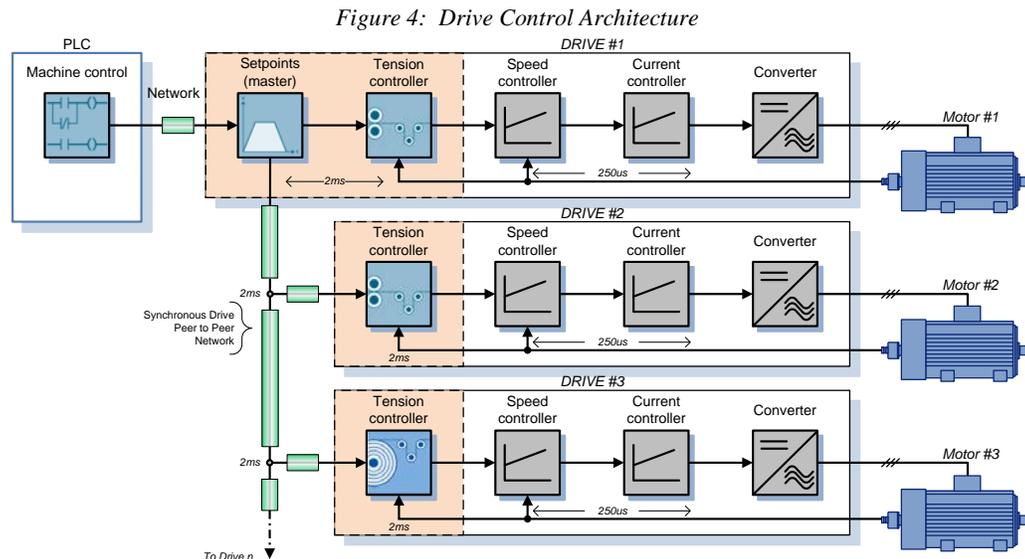
Drive Control Architecture

Figure 4 depicts a drive based distributed control architecture with a PLC for the machine control and drive and web control loops implemented directly in the drive system. The depicted system has a cycle time of 2ms for all of the web control loops, a peer to peer network that is synchronous with the drive program cycles.

With the given data, we can calculate the process throughput for this example drive based control system as;

$$\text{Process Throughput} = (\text{Tension control cycle} * 2) + (\text{Drive Cycle Time})$$
$$(2\text{ms} * 2) + (250\mu\text{s} * 2) = 4.5\text{ms}$$

Performance Reference: In this example, a web traveling at 600mpm will displace 0.45cm before a new correction is made.



Motion Based Control

Motion Control Functionality

Motion control is a sub-field of automation, where in addition to the velocity, the position of drives or “axes” are controlled or measured. The motion control based system offers equal or better process throughput performance to the drive based system with additional performance and usability benefits.

The motion controller’s inclusion of the position loop adds an increased level of performance and improved dynamics. The position loop allows driven axes with significantly different inertias to accelerate at the equal rates without complex feed forward and critical current loop adaptation. In center-driven winders and unwinds, tension control accuracy can be improved with position based diameter calculators. Improved draw control precision and repeatability can be achieved through the electronic gearing of axes. Position control improves splicing and tail control accuracy over time based PLC systems. The ability to use of position setpoint (virtual masters) verses following noisy or unstable actual feedback signals improves stability.

The motion controller will have no comparable memory or performance limitations. Integrated commissioning tools, (tracing, diagnostics & remote control panel) are standard. High speed I/O access for position capture of individual drive axes and cam or “PLS” outputs are also standard features. The full range of programming languages are standard in most in motion based controls.

Integrating motion control converting applications with the main drive control system is a key in the effort to improving productivity. Converting lines have often utilized individual or “stand-alone” motion controlled sections, (for example; rotary knives, flying saws, electronic gearing). When incorporated in a common controller platform, usability is increased. Product information, and other data conveniently shared, and production data can be recorded and saved via the same system that manages the web controlled drives, along with connectivity to plant MES system. Perhaps the only detraction to the motion based control is that with the higher level of performance and flexibility they require a higher level of system knowledge. Additionally some level of machine control can be handled in the motion controller, eliminating the PLC in machines.

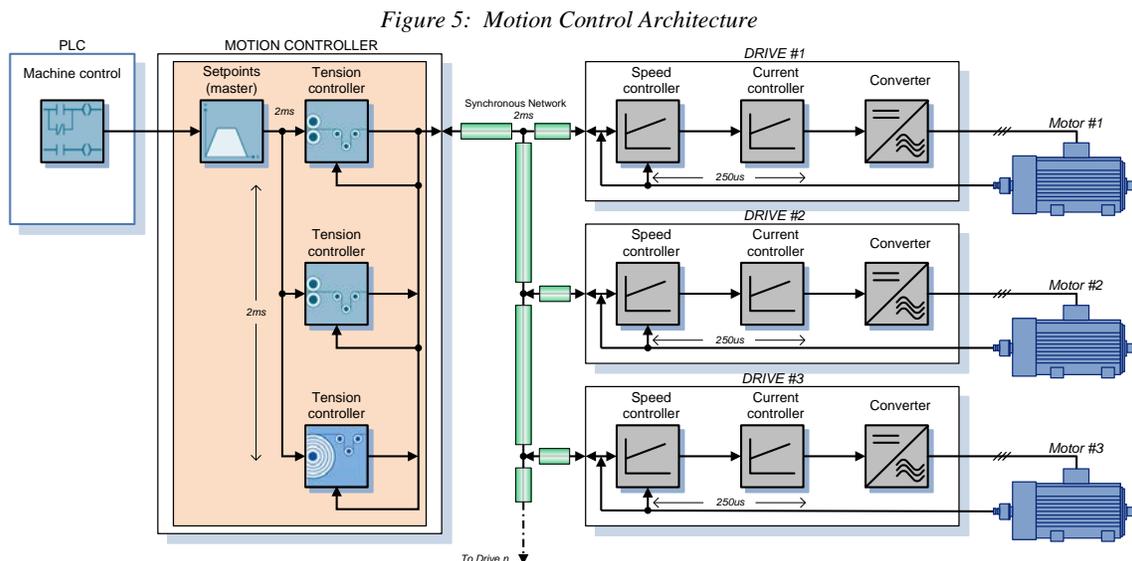
Motion Control Architecture

Figure 5 depicts a centralized control architecture with PLC machine control and the web control system implemented in a single motion controller. The depicted motion based system has a cycle time of 2ms for all of the web control loops, and a common network that is synchronous with the drive and control loop program cycles.

The process throughput for the example motion based control system as;

$$\text{Process Throughput} = (\text{Tension control cycle} * 2) + (\text{Drive Cycle Time})$$
$$(2\text{ms} * 2) + (250\mu\text{s} * 2) = 4.5\text{ms}$$

Performance Reference: In this example, a web traveling at 600mpm will displace 0.45cm before a new correction is made.



Technology Trends

Safety

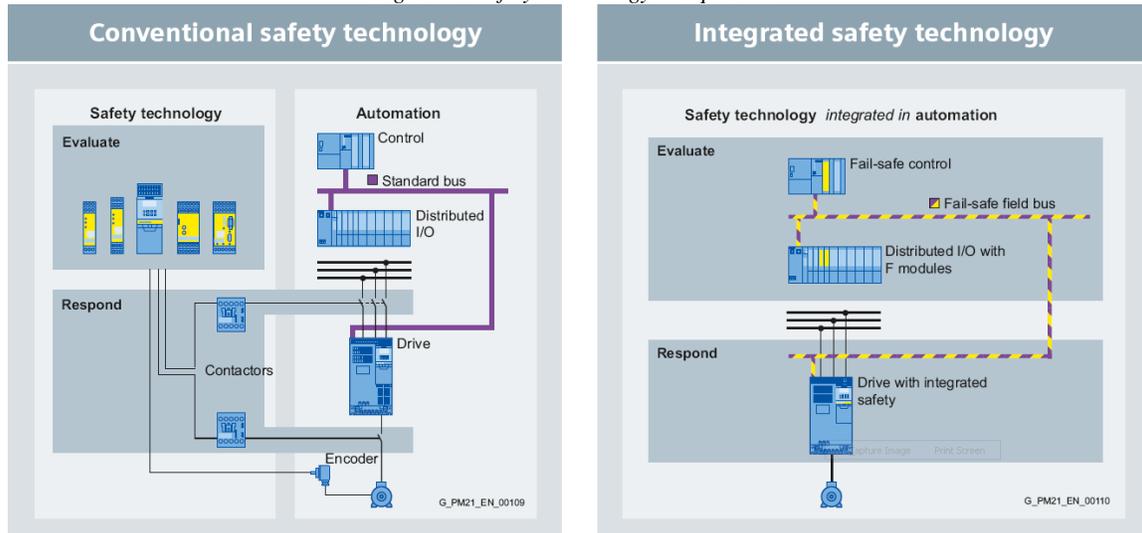
More and more, integrated drive safety is becoming an important part of every machine and control system. As the requirements for machine flexibility and productivity continue to increase, new safety requirements must still be met. Conventional safety technology is at its limits in this respect. Integrated safety technology can be an asset in meeting design and performance criteria and still satisfying today's and future safety requirements.

Using drives with integrated safety technology can result in the elimination of previously required electromechanical components and their associated wiring. The transmission of safety-relevant signals are done via standard field buses, which reduces the complexity and the overhead of wiring. This considerably simplifies the implementation of safety concepts. In addition, they allow for considerably more efficient safety concepts, both in terms of functionality and in terms of response times. This commonly relates to increases in productivity.

Benefits of integrated Safety

- Highly effective safety: - Integrated through all safety components, sensors, drives to the central processor.
- Cost Savings: - Due to reduced hardware and installation costs
- Easy system engineering and maintenance: - By means of safety-related communication via standard field buses
- Effective and fast diagnostics: through a high degree of safety data availability
- Increased productivity: - Fast troubleshooting and comprehensive diagnostics functions reduce downtimes

Figure 6: Safety Technology Comparison



The following list outlines commonly available integrated safety functionality for drive systems.

Functions to stop a drive:

- | | | |
|--------------------------|-------------------------------|---------------|
| ■ Safe Torque Off (STO): | Torque is safely switched off | (Stop Cat. 0) |
| ■ Safe Stop 1 (SS1): | Active braking, then STO | (Stop Cat. 1) |
| ■ Safe Stop 2 (SS2): | Active braking, then SOS | (Stop Cat. 2) |

Motion monitoring functions:

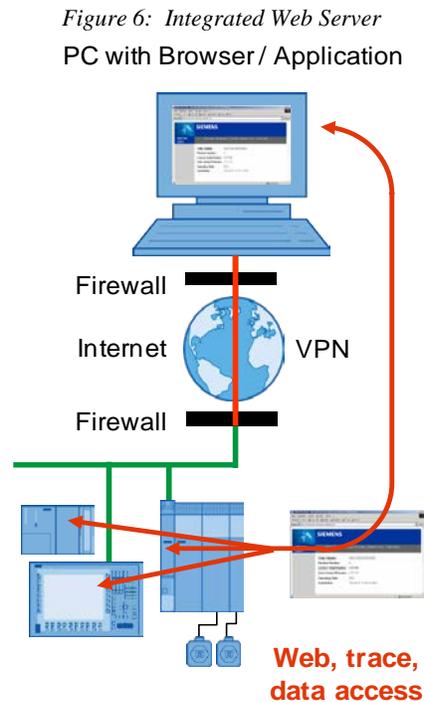
- | | |
|-------------------------------|--|
| ■ Safe Direction (SDI): | Safe direction of rotation |
| ■ Safely Limited Speed (SLS): | Speed is safely limited |
| ■ Safe Speed Monitor (SSM): | Check back signal if speed falls below a limit |
| ■ Safe Brake Control (SBC): | Brake is controlled safely |

Position monitoring functions:

- | | |
|----------------------------------|--|
| ■ Safely Limited Position (SLP): | A traversing range is safely limited |
| ■ Safe Cam (SCA): | Safe software output cams |
| ■ Safe Operating Stop (SOS): | The drive position is safely monitored |

Remote Access & Diagnostics

Integrated Web Servers are now becoming standard offerings in many control components including drives. An integrated web server has many benefits, among them; the ability to obtain diagnostics without use of proprietary engineering tools. Any web-capable PC/notebook/notebook with an Internet browser is typically sufficient. Custom web pages can be created for specific faults, alarms and parameter reading and writing. Firmware and program downloads are also possible without the standard engineering tools.



Conclusions

The task of selecting the optimum control architecture can be an interesting challenge. Each of the systems reviewed here have their benefits and limitations.

PLC based solutions offer a medium level of performance with a reasonable level of usability.

Drive base solutions offer a higher level of performance through their distributed control structure. They typically have the ideal development and maintenance tools for web handling applications, offering high level of performance and usability.

With the continuing demands on production speeds, product quality and considering the general trend of motion based control, integrated drive and motion are not only well suited but fast becoming common place in the industrial landscape. The concepts and benefits of motion control based solutions that offer the highest level of performance and usability can be extremely attractive for both new converting lines and retrofits.

References

- [1] William Gilbert, "Motion Control for Converting", Aimcal Fall Conference, Charlotte, NC 2010.
- [2] Siemens, Isochrone Mode Function Manual, A5E01088188-01, 03/2007
- [3] Safety Integrated for Drives and Motion Control, Siemens Brochure 6ZB5711-0AE02-0AA6, 2011