Commonality between hydraulic valves driven by general-purpose CANopen I/O and hydraulic CANopen drives

Heikki Saha, Bertil Bäck

Hydraulic valves have been driven by separate valve control electronics for decades. The tradition is becoming obsolete due to ever increasing performance, controllability, maintainability and re-usability requirements of the target systems. However, there are still a lot of functionally constrained fully hydraulic systems which need to be upgraded first to electrically controlled. There are also huge number of simple proprietary control systems needing upgrade from proprietary to standard technologies and components. Especially for such kind of systems, it is important to achieve easy and straightforward upgrade path from use of proprietary valve amplifiers into coil-mounted drivers with current-controlled valve. Further step is replacing separate valves and drivers with intelligent hydraulic drives. It is also important to utilize commonality between lower- and higher performance drive solutions. This presentation shows hands-on case examples how the presented system integration challenges can be solved with standard CANopen devices according to device profiles for general-purpose I/O-devices (CiA-401) and hydraulic drives (CiA-408). The main conclusion is that commitment to CANopen actually enables not only the required approaches, but also manufacturer and device independence with large number of interchangeable devices. Another significant result is that the lowest level applications can be implemented just by integrating standardized devices, without application software project.

Introduction

Traditional control system approaches for simple open-loop drives based on application programmable devices are presented in Figure 1. The most traditional approach is presented in the top and is based on proprietary HMI and driver units. Extensive SW design and testing are required, because all functions are applications specific. Due to the same reason, different implementation tends to be used in each application. Typically the HW is also application specific or at least system integrator specific, introducing HW development and maintenance.

Slightly better approach is presented in the middle, where the approach is still to use application specific HMI and driver SW implementations, but implemented with COTS PLCs. The approach enables utilization of development and maintenance of HW and some basic SW done by the PLC vendor.

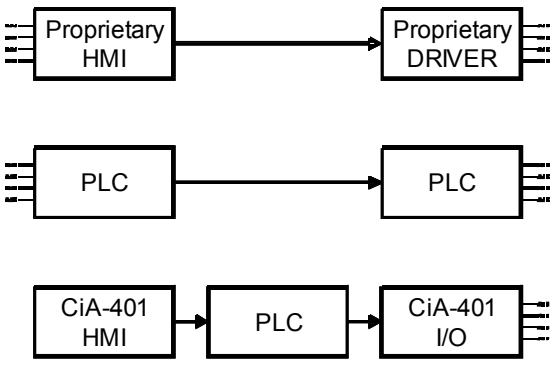

Figure 1: Traditional drive approaches of simple systems

The best commonly used approach is based on the standardized HMI and driver units and a PLC for application processing. While I/O interfacing is standardized, there is still a lot of application programming, maintenance and testing required for each system, regardless of the simplicity of systems. For the simplest direct drive applications, functions implemented in the PLC could

be replaced with function provided by standard I/O-devices /7/ or intelligent hydraulic valves /10/.

This paper presents how to stop the wasting of resources for developing and testing of functions, which are commonly supported by COTS device vendors. This paper also presents how commonalities between general-purpose I/O-devices and intelligent hydraulic valves can be utilized for implementing different levels of functionality without changing the primary integration interfaces.

The simplest open-loop drives are presented first. Three different options for drives are show to emphasize the functional commonalities and flexibility of the approach. Next, monitoring capabilities between the presented options are compared to clarify the achievable dependability of each option. Standard HMI functions are presented after the drive options. The last sections present, how more complex system variants can be derived by adding sensors and PLC based on the increased system requirements.

Simple CANopen approaches

Figure 2 presents simpler approaches for simple open-loop drive applications. Widely supported CANopen device profiles for generic I/O-devices and hydraulic valves make these approaches very attractive. All approaches enable implementation of flexible implementations without application programming. Instead, only system design according to standardized process is required /4/ /5/.

The topmost approach has one major problem caused by discrete I/O-cables. Significant part of the instrumentation is outside the onboard diagnostics capabilities and neither exact error category nor exact location can be determined by the system.

The middle approach is based on single-coil drivers directly mounted on the coils, which mainly solves the diagnostics inaccuracy problem by removing the discrete cables.

The bottom approach is based on a driver and control electronics integrated into valves. Integrated electronics has an access to the valve itself and is able to provide exact and detailed information of the valve behavior.

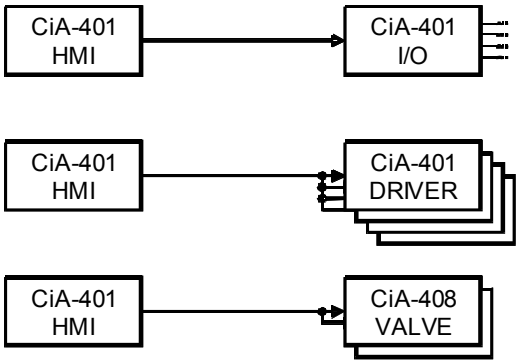

Figure 2: Improved drive approaches of simple systems

General-purpose CANopen I/O devices

CiA-401 defines the simplest interface – presented in Figure 3 – for driving on/off or proportional valves. Due to the simplicity, also driver error management is based on a simple state-machine consisting of two states /7/. Solution is powerful, especially when standardized error value parameter object is supported. It enables system integrator to define application-specific safe output value for error conditions.

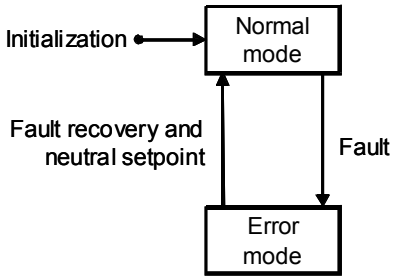

Figure 3: CiA-401 error state machine

State-machine is required for keeping the driver safe. If only shutdown on error were used without a state transition, error may start to oscillate on and off and cause safety risks. With state-machine, error turns the device to a dedicated error state, where it remains until the exit condition is met. Such a mechanism prevents safety risks caused by errors oscillating on and off, and error recoveries hidden from the rest of the system.

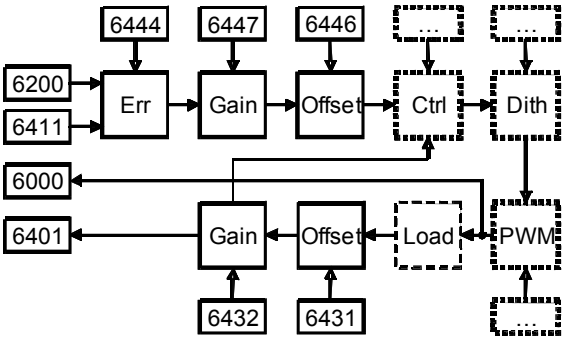

Figure 4: Simplified block diagram of a CiA-401 compliant valve driver

Blocks with solid lines are standardized by CiA-401 /7/. Blocks with dotted lines are required for valve drivers, but they do not violate the device profile, they just provide additional features beyond the I/O interface. Most of the parameters follow directly CiA-401. Default setpoint signal type is INTEGER16, which is also default type in CiA-408 compatible valves. The combination of feedback signals provides error mode feedback. Error mode is optional from CiA-401 point of view but it is a necessary safety function from system point of view.

CANopen-compliant valves

While generic I/O-devices are simple, CiA-408 compliant hydraulic valves /10/ can have more flexible control behavior and error management, based on device state machine presented in Figure 5.

The most important configurable parameter is *DeviceLocal*, which enables optional use of automatic device state transitions instead of remote state control. Such configuration enables the valves to automatically enter *DeviceModeActive* after startup and an error recovery requiring only a neutral setpoint /10/. Thus, in simple open-loop drive applications valves can be used in a similar way than I/O-devices. The main consequence is that most of the simple systems can be implemented without time consuming and expensive application programming.

Figure 6 presents a simplified structure of CiA-408 compliant valves /10/. Most of the blocks are conditional – if functionality is supported, then it shall be implemented according to CiA-408. Thick arrows define signal paths, where may exist multiple signals.

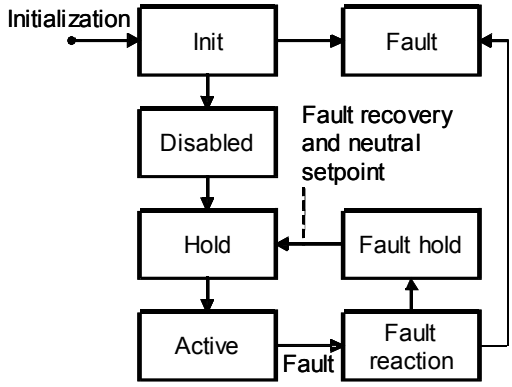

Figure 5: CiA-408 device state machine with only most essential state transitions documented

Device control and *program control* blocks manage the operating principles of the valve – e.g. a selection between local and remote control and control-mode of the internal controller /10/.

Each control mode has dedicated group of blocks – *setpoint conditioning* – for adapting the setpoint signal to the system requirements. Its sub-function provide same or more flexible behavior than offset and gain in I/O-devices. Valve calibration and compensation functions fall under *output conditioning* and *actual value conditioning* provides a standardized interface for calibration data of integrated feedback sensors /10/.

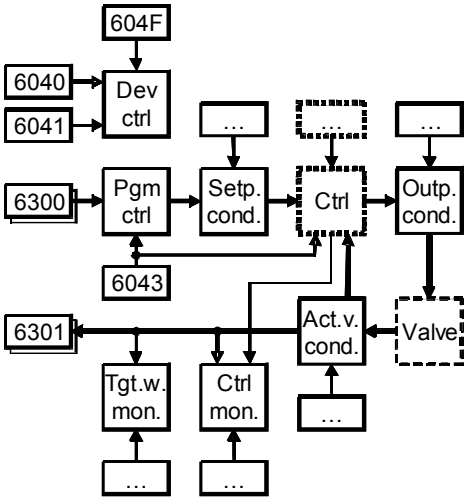

Figure 6: Simplified block diagram of a CiA-408 compliant valve

Target window monitoring and *control monitoring* blocks provide standardized monitoring functions for the integrated controller. Controller algorithms are outside the scope of CiA-408, which clearly points the area for competition –

control performance instead of interface /10/.

Monitoring issues

Detailed view for physical interfacing of the valves is presented in Figure 7. The topmost variant presents the most traditional approach, where separate I/O devices are installed apart from the valves. Cables are the weakest components, because of the unidirectional analog signaling principle. I/O-device is not able to determine exactly, what kind of failure exists and where. The average diagnostics inaccuracy is few meters.

The second approach in the middle presents an improved approach with single coil drivers mounted directly to the coils. Main benefit is that there are no more cables between drivers and coils, which significantly increase the accuracy of diagnostics – diagnostics inaccuracy drops down to few centimeters. Furthermore, required assembly work and service effort can be reduced due to simplified cabling. Furthermore, simplified cabling significantly improves assembly and service work by reducing number of errors.

The third approach in the bottom presents the control electronics integrated to the valve. System integrator attachable connections between valve driver or controller and the coils do not exist anymore. Moreover, integrated electronics can have direct access to several measurement points of the valve itself. Thus, diagnostics capabilities cover the coils and also the hydraulic parts of the valve. Setpoint and status signals are compatible between CiA-401 and CiA-408.

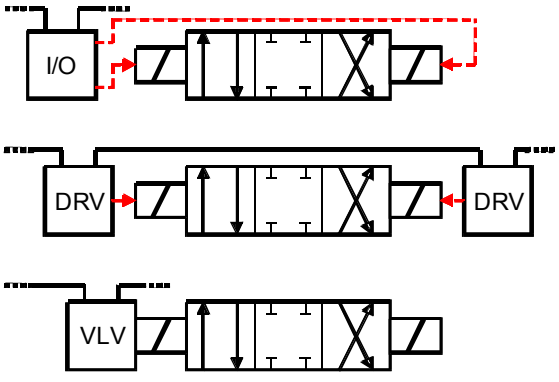

Figure 7: Diagnostics capabilities in different approaches

Very basic CANopen communication services provide extensive fault monitoring features. NMT state-machine together with NMT- and heartbeat-protocols provides the most basic device monitoring for both startup phase /2/ /3/ and run-time /1/ /2/. RPDO timeout monitoring provides fast and reliable validity information for setpoint signals /1/. Application layer state-machine can not be used, because single directional valve needs two drivers.

Properly configured error behavior supplements the presented methods. After detecting e.g. communication or device failure the device can change its NMT-state, which is continuously updated to the whole network with heartbeat protocol. In the case of device error, invalid signal values can be blocked inside the device by setting the device to NMT stopped state, in which the device is not allowed to transmit any process data /1/.

CANopen HMI

Following CiA-401 in the valve drives was presented earlier in this paper. The same device profile also applies for the HMIs. HMI in this context means joysticks, pedals, buttons and indicator lights /7/. The basic idea remains unchanged – following standardized basic structure and interfaces provides vendor- and product-independent implementations.

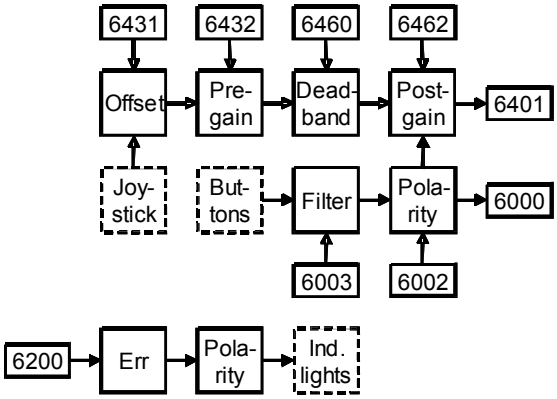

Figure 8: Simplified block-diagram of a CiA-401 compatible HMI

A simplified block diagram of CiA-401 compliant HMIs /7/ is presented in Figure 8. Joystick interface is the most essential one. To be able to hide analog joysticks behind the standardized interface, certain functional blocks are required. Nonlinearities of the potentiometers and

inaccurate mechanical center points are compensated with *Offset* and *Pre-gain*. Because output of *Deadband* varies from zero to maximum – deadband, additional *Post-gain* is required to make the output signal reach the maximum values.

Indicator outputs are very simple. Error block is needed for enabling application specific error states for the indicator lights. Polarity control improves the adaptability further by enabling the use of either active-high or active-low signals /7/.

More complex systems

There will often be a need for more complex product variants or products at some point. Then it is important that maximum number or components – both hardware and software – can be re-used. If very simple open-loop direct drive systems need to be improved with application specific functions, application processing platform – e.g. PLC or display – can just be added. Very important notice is that application requirements actually define the system complexity, not CANopen.

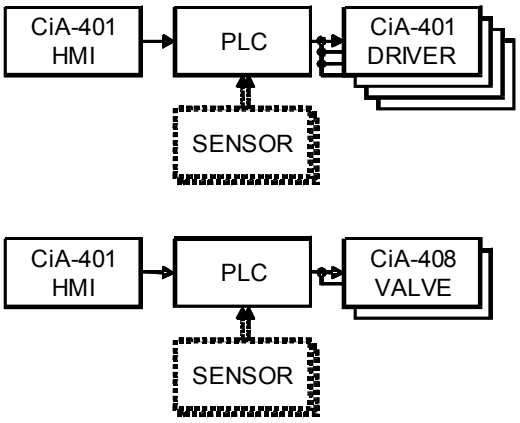

Figure 9: Improved drive approaches of complex systems

If a PLC exists in the system, various sequences, application state-machines and control loops can be implemented. When a PLC is controlling the CiA-408 compliant valves, it is recommended to use remote device control to prevent unintentional system actions before intentional clear of errors. CiA-408 device state-machine is a standardized application level safeguard providing more flexible functionality than the simple CiA-401 error state machine /7/ /10/.

If additional measurements are required, system can be expanded by connecting additional sensors /8/ /9/ /11/. Figure 9 presents an example, how a system based on HMI and valve drivers or valves can be expanded just by adding a PLC and some sensors.

NMT-master

In very simple direct drive applications, HMI is a logical device to support NMT-master functionality. In more complex systems it is better to assign the PLC as an NMT-master. Then the PLC application can also use default SDO-channels for parameter accesses, if required by the NMT-startup routine or applications. It is recommended to use only the default SDO-channels for maximizing the compatibility, because all CANopen devices do not support additional SDO-channels.

Various approaches can be selected for NMT-startup, depending on the system compatibility and safety requirements /2/ /3/. The simplest approach to use *NMT reset communication* for all nodes and *NMT start all nodes* is absolutely prohibited, because such approach does not support any protection against system inconsistency. If there are devices, which do not belong to the system configuration, they are started. Also valid nodes with invalid node-IDs are started.

The most compatible approach is to just start up the devices individually /2/. There is a risk, that if a wrong device exists at valid node-ID, the system will start without noticing the invalid system setup. The probability of such errors can be strongly reduced by following the CANopen system management process in the systems configuration design and management.

Much safer system startup can be reached, when device type of each device is checked during the startup. Such approach provides maximum manageable compatibility – each device need to be only correct type /2/. Most device profiles use additional information field in a clever way by providing more detailed information of the device structure or class /7/ /8/ /9/ /10/ /11/. However, the system consistency is mostly managed by the system management process /4/ /5/.

More detailed device identity checks – defined in CiA-302-3 – can be activated, but the more detailed checks are used the less different components can be used as drop-in spare parts /3/. Again, application requirements introduce the constraints, not CANopen or its device profiles.

It is also possible to start the system in CANopen level without any checks and let the application – if any – to perform the checks. It is not recommended, because there does not exist a standardized process for transferring configuration information. It is also a troubleshooting nightmare, because of the proprietary behavior.

Parameter management

Parameter management is straightforward, but only as long as CANopen system management process /2/ /6/ /12/ is followed. In very simple systems, all parameter values are set offline and loaded to the nodes during system assembly and spare part change. If field calibration or system specific device parameters need to be modified, external tool is needed. Parameter categories are listed briefly in Table 1.

Table 1: Parameter categories

Category	Description
Calibration parameters	Factory calibration values
Run-time adjustable parameters	*Optional* field calibration values
	Optional system specific device parameter values
Off-line adjustable parameters	Position specific device parameter values
	Position specific communication parameters

When a PLC or other device supports parameter modifications, two more challenges occur. First, how to define which parameters can be modified? Publishing all parameters will consume too much memory and too many parameters will confuse the operators. Second, how to manage parameter attributes, such as minimum, maximum and default values?

One can see from Table 1, that majority of the parameters need to be written only before installation. Only minimum set of parameters need to be accessed run-time. CANopen system management process solves both of the presented challenges, but it is outside the scope of this paper. Interested readers can learn more from literature /4/ /5/ /6/ /12/ /13/.

Maintenance

Previous sections described how development effort of very simple systems can be minimized by utilizing CANopen concepts. Required maintenance effort can also be reduced significantly. Because very simple systems can be implemented without application programming, all application SW maintenance and testing effort can be saved. Instead, device vendors maintain and test the devices. Because of higher production quantities of the standard components, maintenance and testing cost for single device is much lower than with application specific devices.

In very simple systems, the system design consists of parameter values stored in standardized design files. In application SW based approach majority of the design is in application SW source code, which always contains something target HW and compiler specific parts. It is obvious, that standard design files defining the detailed operation of standardized devices are easier and more efficient to maintain in the long term.

There is interesting evolution going on in hydraulic valves. Currently there are various traditional analog integration interfaces – multiple current ranges, ratiometric voltage and standard voltage or current signals. Only standardized integration interface enables the use of best valves for each application, without a need for dedicated drivers for each valve type. Standardized integration interface is also the best approach to prepare for rapid adoption of the future valve technologies and to ensure the component availability in production and service.

Concluding remarks

On the contrary to the "common understanding" in the industry, CANopen systems can be as simple as possible based on the application requirements. The simplicity can be achieved by thinking differently and utilizing the standardized interfaces. For example very simple systems can be implemented without application programming, based on the products according to commonly supported device-profiles.

Hydraulic drives can be scaled feature-wise by utilizing the commonalities – same setpoint signal type and comparable device state-machine – between generic I/O-devices and hydraulic drives. The main prerequisites for utilizing the functional commonalities are the deep understanding of the corresponding device profiles and the use of components compliant with the profiles.

Very simple systems can be developed further by adding sensors and application processing platforms without a need for discarding any existing device.

When compared with discrete wiring, the commitment to the components supporting standardized CANopen interfaces and services not only enables more efficient development and maintenance, but also improves the systems dependability.

References

/1/ CANopen Application layer and communication profile, CiA-301, CiA

/2/ Additional application layer functions Part 2: Network management, CiA-302-2, CiA

/3/ Additional application layer functions Part 3: Configuration and program download, CiA-302-3, CiA

/4/ Electronic device description Part 1: Electronic Data Sheet and Device Configuration File, CiA-306-1, CiA

/5/ Electronic device description Part 2: Profile database specification, CiA-306-2, CiA

/6/ Electronic device description Part 3: Network variable handling and tool integration, CiA-306-3, CiA

/7/ Device profile for generic I/O-modules, CiA-401, CiA

/8/ Device profile measuring devices and closed-loop controllers, CiA-404, CiA

/9/ Device profile for encoders, CiA-406, CiA

/10/ Device profile fluid power technology proportional valves and hydrostatic transmissions, CiA-408, CiA

/11/ Device profile for inclinometers, CiA-410, CiA

/12/ IEC 61131-3 programmable devices, Implementation and user guideline, CiA-809, CiA

/13/ Saha H., Benefits of intelligent sensors and actuators throughout the systems life cycle, The Twelfth Scandinavian International Conference on Fluid Power, May 18-20, 2011, Tampere, Finland, ISBN-978-952-15-2517-9, pp. 169–181

Dr. Heikki Saha
Research Engineer
Sandvik Mining and Construction Oy
UGM Applied Research
P.O. BOX 100, FIN-33311 TAMPERE, Finland
Phone: +358 (0)400 346 537
Fax: +358 (0)205 44 121
Email: heikki.saha@sandvik.com
http://www.miningandconstruction.sandvik.com/

Bertil Bäck
R&D Manager, HW
TK Engineering Oy
Yrittäjänkatu 15, FIN-65380 VAASA, Finland
Phone: +358 (0)50 588 6895
Fax: +358 (0)6 357 6320
Email: bertil.back@tke.fi
http://www.tke.fi/