How to write control sequences

The development of prescriptive, detailed sequences as part of the design documents is now more critical than ever. As newer, more complex systems are applied to achieve greater energy efficiency, there is a need for equally new and complex sequences of operation.

By Jay Santos, PE, Principal, Facility Dynamics Engineering Inc., Columbia, Md.

02/01/2008

The development of prescriptive, detailed sequences as part of the design documents is now more critical than ever. As newer, more complex systems are applied to achieve greater energy efficiency, there is a need for equally new and complex sequences of operation. Additionally, with commissioning becoming commonplace, detailed sequences are necessary for the commissioning engineer to develop proper functional testing. Without detailed prescriptive sequences, the commissioning engineer must referee various interpretations of a sequence of operation. Altogether, the responsibility for developing the sequences rests with the engineer and it is critical to be well documented early in the building process.

This article presents an organizational process to help manage the thought process needed to write and create good sequences of operation.

Divide and conquer

An effective way to organize and prepare control logic is to categorize the logic into two distinct types—process- and supervisory-level control logic. Process-level control logic is the control logic associated with the direct control and response of the controlled device. In a typical control loop it consists of the interaction of the process variable, the control response, its setpoints, and any special limits and conditions associated with that loop. The supervisory level logic is more similar to overhead logic. It involves the collection of data from multiple sources, analyzing this data and sending a specific instruction to the process level control. It also may involve special alarming, complex reset strategies or the logic for operating a redundant system. There are no specific rules for discriminating between these two levels of logic but it is a convenient way to think about control sequences.

An eight-step process

An overall process for the development of control logic involves the following basic eight steps listed below and then described in turn:

1. Identify the devices to be controlled in each system
2. List the nature of the control signal required for each of these devices
3. For each controlled device, select the process variable
4. Define the relationship between the process variable and the controlled device
5. Identify the various parameters that are required for this relationship
6. Define the control logic required to produce the parameters
7. Determine discrete or analog conditions that apply to each process
8. Define the logic or write the sequences that describe the discrete and analog conditions necessary.

Identify the devices to be controlled in each system. First, look at one building system holistically. For example, an air-handling unit may consist of the following devices to be controlled, as shown in Figure 1:

- Fan start/stop & fan variable speed drive
- Mixed air dampers
- Cooling coil valve
- Heating coil valve
- Humidifier.

For each of these controlled devices, the next seven steps need to be considered.

List the nature of the control signal required for each of these devices. For this air handler, this list of required control signals would be:

- Fan start/stop: digital output (DO) point for the fan starter circuit
- VSD: analog output (AO) point
- Mixed air dampers: AO point or points
- Coiling coil valve: AO point

- Heating coil valve: AO point.

Select the process variable (Pv) for each of these control loops. For our example this list of process variables might look like this:

- Fan start/stop: Pv = time, internal time schedule
- VSD: Pv = static pressure, duct sensor (AI) point
- Mixed air dampers: Pv = temperature, mixed air temperature sensor (AI) point
- Coiling coil valve: Pv = temperature, discharge air temperature sensor (AI) point
- Heating coil valve: Pv = temperature, discharge air temperature sensor (AI) point.

Define the relationship between the Pv and the controlled device. Typical control relationships can be:

- A control loop response
- Two position control
- Floating control
- Proportional control
- Proportional plus integral (PI) control
- Proportional plus integral plus derivative (PID) control
- Time schedule
- Mathematical relationship.

Identify all the parameters required for the relationship defined in the previous step. In many cases these parameters are the output variables of secondary control logic. For instance, the discharge air temperature setpoint might be reset by the return air temperature. This example of secondary logic is a reset control loop and requires its own measured variable (see the cooling example in Figure 1). Alternatively, this parameter may come from higher up in the control system. An example may be the resetting of setpoints based on the exceeding of an energy demand threshold. In this case, where supervisory logic is used, the network architecture and its robustness become critical to the effectiveness of the control. In other words, if you are using the network to communicate critical control parameters, speed may be an issue. This may influence specifications related to hardware, networking, and associated loading of the control system from a bandwidth perspective.

Define the control logic or write the sequences to produce the parameters identified in the previous step. Identify and define the measured variables that are required for this logic. Define the relationship between the measured variable and the output of this control for each parameter. Describe this relationship in the sequence of operation or in the control logic.

Determine discrete or analog conditions that apply to each process. The conditions referred to here are typically a test that needs to be proven a certain way before subsequent event can take place. This also is a useful place to emphasize the requirement that a particular condition may need to be hard-wired if the design calls for it—or if a software-based interlock is OK. Examples of discrete and analog conditions:

- The end switch on an outside air damper must be proved open on a 100% OA Unit before the fan runs
- The system must be in an occupied mode before the outside air damper opens
- If the supply fan is not moving air, the outside air dampers will shut and the return dampers will open
- If the outdoor air temperature exceeds 68 F, send the outdoor air dampers to minimum position
- If the discharge air humidity exceeds 90%, shut off the humidifier.

Define the logic or write the sequences that describe the discrete and analog conditions necessary. This requirement may require secondary logic, which also may require additional measured variables. It also may use information already required and collected elsewhere in the control sequence for this system or others in the building.

Once this has been done for each controlled device in a system, you are ready to review it from an overall performance perspective. If you've done a good job of defining the basic processes, included all the required limits and conditions, and identified all the supervisory control logic, then the system should be well coordinated with the other control processes of the system. This “inter-loop” control action is some of the more complex logic and easily overlooked during the design process, but this logic usually impacts energy efficiency and operability the most and as such needs careful consideration.

Check and check again

It is not unusual to rewrite sequences or re-do logic diagrams two, three, or four times before one gets it right. But consider the alternative—a general sequence, leaving out numerous details and relying on the installing contractor to make these design decisions. This process may or may not work out depending on the circumstances. As a useful exercise, take a sequence of operation that you use and attempt to develop a logic diagram for the sequence. Typically, you'll find numerous instances that the verbal sequence doesn't contain clear direction on specifics on control of the system. Each time this occurs, rewrite the sequence more clearly to fill in the details. You'll see that the verbal sequences get much longer. You'll also appreciate all the decisions that are being made (without you) by the installing contractor that will affect the system performance.

The process of writing good control sequences or developing detailed logic diagrams for HVAC systems is not trivial. The logic that gets programmed into the direct digital controls system is the brain of the HVAC system and as such determines the performance of the system. Developing good engineering direction in this area is key to the success of the installation. The development of these sequences deserves more time in the design of HVAC systems than it is typically given. The above process is one way to organize thoughts in preparation of sequences or logic diagrams.

Author Information

Jay Santos is a principal and co-founder of Facility Dynamics Engineering. Santos has directed the development of DDC/control master plans for numerous institutions and organizations. He teaches DDC courses for the University of Wisconsin, North Carolina State, and the PGE Energy Center in San Francisco. The original idea for the process discussed in this article came from course material developed by Robert Schultz, PE, chief engineer, TAC Americas, and it was presented at various DDC Courses taught by Santos and Schultz.

Use logic diagrams to communicate

I have been teaching HVAC Controls for about 25 years. The communication method necessary for a teacher/student process is similar to what is required for an engineer/contractor. I have found that teaching using logic diagrams works very well for communication a sequence of operation. Logic diagrams (see Figure 1) are a great visual tool in the classroom and work equally as well as a means for communicating sequences as part of a design. While teaching sequence development, it became obvious that we needed to develop an organization process for approaching the development of sequences in various HVAC applications.

For those that frequently develop detailed control sequences, this process may seem second nature. The experienced control engineer may skip numerous steps or combine multiple thought process while developing this control logic for a given application. In controls language, one needs to develop a control loop for each controlled device. This control loop typically consists of a sensor, controller function and controlled device (see Figure 2). Additionally, the designer needs to consider special conditions related to the control loop and finally overall loop interaction within the system and with outside variables and parameters.

Are engineers losing control of sequences of operation?

Over the past few decades, there has been a tendency to provide less detail in the development of a sequence of operation. It is interesting to reflect on what might have contributed to this trend. First, over the past couple decades, we have transitioned from discrete pneumatic logic to direct digital controls (DDC) in the HVAC business. The fundamental differences in these hardware platforms had a profound effect on the design of these systems. Pneumatics consisted of individual components that typically had a discrete function (like a switching relay, a high-signal selector, or a controller). So, one actually could lay out the logic using the functions of these devices on paper or reverse-engineer it from tracing how things were connected in the field.

With DDC, the control logic is buried in software—it's not as “visible.” As this transition occurred, we also transitioned from our pneumatic control technicians to DDC technicians who now require HVAC, controls, electrical, computer, and networking skills. Second, with a focus on increased liability, it argues that performance specifications are safer then more prescriptive sequences. While more general sequences are safer from a liability perspective, they are not in the best interests of the projects or the ultimate inheritors of the systems.

Don't forget the networks

Although this article is focused on the process of the development of sequences, it is worth noting that there are hardware and networking implications of sequences that need to be considered. Many similarities in hardware and networking platforms are in our industry, but there are many differences as well. Certain systems are able to handle a complex sequence of operation of an air handler or a variable air volume box in a single controller. Others will require multiple controllers to do the same logic or require the use of higher end supervisory controllers to perform some of the functions. This is not as robust as if all the logic was resident on a single device. In critical applications this should not be allowed except for the supervisory logic that originates from elsewhere in the system.

The engineer needs to be aware of the limitations that certain products have in this regard. Also, if the control sequence requires communication across the network by design (for example, the highest demand air handler communicates to the chilled water distribution system, or air terminal demand reset mode or setpoints in the air handler), then particular attention needs to be paid to the architectural robustness of the system. In these cases, one is relying on the network for basic control function and its speed and robustness becomes critical to the basic function of the control logic. This type of sequence cannot be required “blindly” without specifying networking performance criteria. The engineer will need to specify performance criteria for the system while fully loaded (all trending and alarming features operational). This may require limitations on the number of devices allowed on sub-networks in a building depending on the manufacturer.

Post a comment Log in or create an account to submit your comment for this article.

No comments