

Disciplined Approach Allows For Rapid Development of Control Systems

By Cal Swanson

Senior Principle Engineer, Single Iteration
(a division of Watlow Electric Manufacturing Company)

DEVELOPING A CONTROL system rapidly and cost effectively requires a disciplined approach that exposes design limitations early so they can be corrected before the costs and schedule go out of control. This article will explain the basic method to rapid control system development as well as ways to create a reliable system model that a virtual controller may be developed against.

A “Trial and Error” development approach is utilized primarily when the control system is presumed simple, easy, “just like the last one,” or when there is simply no time to conduct a full development cycle. It can be characterized by a relatively small investment in up-front planning, followed by multiple iterations of “build” and “test.” If something profoundly unexpected happens, the process can appear to be endless and the project may die from lack of apparent progress or the end of funding. Figure 1 illustrates the Trial and Error method of development.

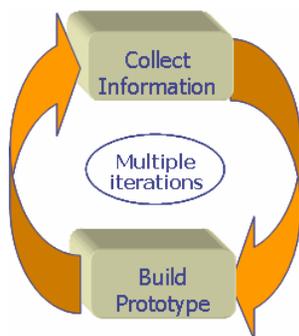


Figure 2 Trial & Error Method

Development projects typically go out of control either from underestimating the complexity of the tasks involved or because one or more unexpected obstacles violates the development plan assumptions. This is where a disciplined approach that includes a carefully planned, methodical front-end effort can actually reduce the time to deployment.

Rapid control system development relies first on obtaining a good system model of the process to be controlled followed by, or in parallel with, controller algorithm development in a graphically-oriented, interpretive software environment. The algorithm is then tested against the model and iterated upon as necessary before testing [and?] applying it to the physical process and committing to controller hardware. Figure 2 illustrates the Rapid control system development process.

System and Controller

Figure 3 illustrates, in block form, the interconnection between a controller and the system to be controlled. A great deal of complexity may reside within these blocks, for instance a system may contain a collection of subsystems and their controllers. Development of the controller then first begins with understanding the system to be

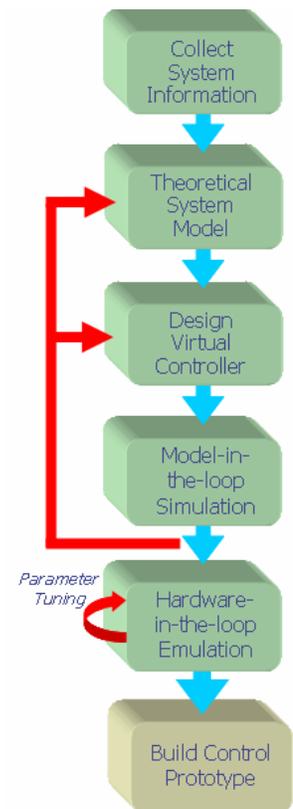


Figure 1 Rapid Control System Development

controlled, and to do so, we create a reliable model of it which is independent of any controller we may choose.

Collecting System Information

To create a good system model, is important to obtain real, physical system-specific data that fully describes the system to be controlled. Using just theoretical data at this stage can often lead to mischaracterizing the system and defeat the control development process.

Extraordinary system behavior must be noted as it is often an indicator of some greater complexity that affects control. One of the greatest contributors to cost and schedule overrun in any development lies in underestimating the complexity of the system you are modeling. In particular, look for evidence of the following when attempting to characterize a system:

- Multiple interactive inputs or outputs
- Non-stationary effects
- Unforeseen boundary or load conditions
- Time delay
- Non-linearity
- Noise and signal contamination

What kind of system model?

This process requires a good system model – one that is appropriate to the system you wish to control and one that can be reduced to mathematical and/or logical description of how the output(s) respond to input(s) for all relevant environmental/boundary conditions. For typical thermal systems, there are essentially three basic models to select from. A system could be described with one or more transfer functions relating a linear or non-linear relationship between input and output through discrete (lumped-parameter) components. Alternatively, for a more distributed or continuous system where the concern is propagation, finite element modeling may be the best choice for a system model.

A finite element system can be reduced to state space equations through tools available within MATLAB®, which subsequently may be used to evaluate a control system just as well as with transfer functions. A state-flow diagram is a third descriptive system representation where discrete decisions are made based upon system inputs and/or outputs, perhaps including previous input and output history. State flow systems may be viewed as a method to stitch together discrete control processes such that as one ends, the next begins. Many other ways exist to model systems models that have their origins in different analysis or processing techniques, such as fuzzy logic, or neural networks, but here we will focus on transfer function and finite-element models for the system description.

Transfer System Model

A transfer system model works well with linear systems that have discrete components or whose components can be represented as discrete parts. Transfer system models can be multidimensional, but due to interactions or interdependencies between the parts, the inputs and/or the outputs can complicate the system model very quickly. A hot-water-cold-water

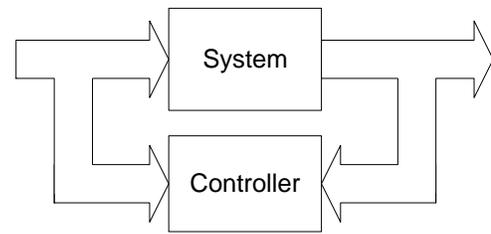


Figure 3 System/Controller Relation

temperature regulating system is an example of a system that could be modeled with transfer functions. Two inputs consisting of cold water and hot water are controlled by valves to produce an output flow stream at a certain temperature and flow rate. In this system, changing either or both of the inputs affects both output variables (temperature and flow rate). One might assume for this control that the output flow rate and temperature is the linear superposition of what the cold water valve can produce and what the hot water valve can produce. Testing the system, on the other hand, may reveal that there is also a dependency upon both the hot and cold water supply pressure and that maybe this pressure fluctuates with time as other demands on the water supply dynamically change the supply pressures and affect output. For instance, a large nearby cold water demand – like a toilet flushing – may require a different set of valve positions to maintain the desired output temperature and flow rate.

To adequately model this system and move to the next stage of controls development requires that all the parameters that affect output be accounted for and tested where possible. Thus the variables of pressure, temperature, and valve position need to be examined to create a mathematical relationship between inputs and outputs that considers responsiveness, stability, and linearity (valves stick, inlet water pressure and temperature varies). To adequately control a system, the controller must be faster than the parts and processes within the system, or it must have foreknowledge of what is to happen, so establishing the inherent bandwidth of the system to be controlled is essential. In the example of the nearby toilet flushing, the controller must be able to detect the oncoming change in conditions and adjust the valves in sufficient time to compensate for the water pressure change. If the controller or valve changes are too slow or the time required to detect the oncoming conditions are too long, then the controller may become unstable resulting in loss of control.

Finite Element System Model

When looking for temperature uniformity/distribution of a sample or structure, the finite element model may be easier to create than a transfer function model. A finite element system model is a good candidate for distributed systems where energy propagates or stresses distribute through continuous material or materials. Almost always, there are boundary conditions or structural variations that influence propagation or distribution. In thermal systems, we are primarily talking about heat flow and temperature distribution in structures. More often than not, mechanical stresses and strains accompany temperature differentials that, in turn, may distort the structure and change how heat flows. Since finite element models artificially divide (mesh) structures into small cells that are acted upon and react to adjacent cells in small discrete time-steps, they generate large and usually sparse matrices of equations that define the actions and interactions of each element. Finite element models can get into trouble when the aspect ratios and/or skew of their elements become too big – that is elements that are long in one dimension relative to another or the angles are too different. Intelligent meshing is required to combat this without growing the matrices so large that computation time becomes excessive.

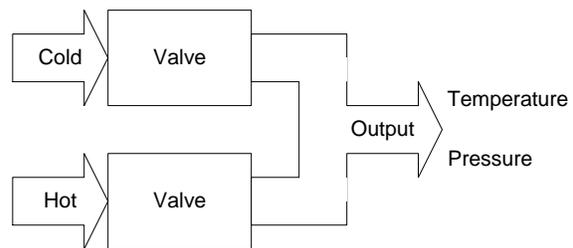


Figure 4 Hot/Cold Water Regulator

Consider the design of an extrusion plastometer, an analytical device used to determine the melt properties of plastics or polymers, which must control the temperature of the entire sample to within some small temperature deviation within some desired time. In this example, one may be lulled into the idea that the problem can be handled easily by an off-the-shelf controller and that there is no real need to follow a disciplined approach to developing the control system. In this case, there may be the complexity of uncontrolled variations in the thermal contact resistance between the heater and extrusion barrel that make it difficult to maintain sample uniformity, no matter how complex or sophisticated the temperature control is. To ensure that the desired uniformity is achieved on the sample, the system structure must be carefully defined with respect to all electrical, mechanical and environmental boundary conditions surrounding the sample, including how uniformity will be sensed and measured.

Test and Measurement: What to look for

Upon deciding the system model or models to use, the next step is to create a basic model of the process you wish to control, and then experimentally verify that the model is a good representation of the physical process. Next, you must determine how the real system differs from the theoretical – especially during special conditions or situations. Nearly all real systems have some non-linear or non-stationary behavior, and nearly all of them have noise contamination.

Testing will reveal how important these deviations are and whether their significance warrants inclusion in the theoretical model. The key in making a good system model is to identify potential problems or situations that the system is exposed to, and see how much it affects the way the system responds. This article has touched briefly upon multiple input/output systems and stationary effects, but there are other potentially major effects to consider.

Time Delay

One of the largest factors in the ability to control a process is ensuring that the required information is presented to the controller in sufficient time for it to compute and execute a response. Controlling a process with too much time delay would be like driving a car and missing your turn because you didn't see a street sign in time. Knowing the time delays in your system and where they come from is essential to control. In thermal systems this is equivalent to applying power to generate heat then not being able to pull it back when you determine it is too much.

Non-Linear Behavior

Most systems exhibit some non-linear behavior, and usually if the non-linearity is small it can be ignored. However, when non-linear behavior composes perhaps 10% or more deviation from linear, it probably needs to be seriously considered. Continuous non-linear behavior is, in many respects, easier to characterize and compensate for than discontinuous or eventful behavior. Continuous non-linear behavior, to continue the driving analogy, is like dealing with the fact that it takes about four times as long to stop when you are traveling twice as fast. This is opposed to eventful or intermittent behavior like a dead-band, where small changes in steering wheel position have no effect turning the car, and outside of which everything seems to work normally. In thermal systems, continuous non-linear behavior may manifest as changes in material properties as temperature increases. Discontinuous behavior, on the other hand, may show up as

a temperature-dependent contact or non-contact between two material interfaces, below which there is very little heat transfer, and above which heat flows very differently.

Noise Contamination

Noise and signal distortion can be particularly troublesome to contend with in a control system, so it is important to find the sources of noise and other kinds of contamination before attempting to introduce control to a system. Noise can come from a lack of sensor signal sensitivity or from too much sensitivity as in sensor saturation. It can be external to the system, entering from uncorrelated and unaccounted for electrical/electromagnetic, mechanical, or thermal sources including problems with shielding and grounding.

Developing the System Model

Gathering Response Data

Whether you are making sure your system responds as it was designed or simply trying to characterize a system, the system you are attempting to control must be stimulated to see how it will respond. Depending upon your system, you may elect to choose from one or more different stimulus methods. For thermal systems, the most common stimulus method is the step function – that is, changing the input temperature or power from one value to another and observing what transpires over time. A related method is the impulse function where power is instantaneously applied then removed, again observing the effects over time. For electrical or mechanical systems with wide response bandwidths, frequency-specific oscillatory inputs or random noise inputs are often used because of their inherently higher signal-to-noise ratio. For thermal systems that normally respond very slowly, these other two methods usually take far too long to justify their advantages.

To be effective, step-response testing needs to be performed with the system at normal ambient conditions and close to the same operating parameters it is expected to perform in. It is important to be aware of that non-linearity within the system can affect your results, so it is often a good idea to test with a variety of amplitudes to see how responses compare and to ascertain the amount of non-linearity within the system. Data is acquired the moment heat is applied and the response to the stimulus is usually measured in multiple locations. If a model has already been developed, then the same stimulation is theoretically applied to the model and the model's response is compared to the real data. The model is then modified as necessary to approximate the real response.

Developing a Model

If a model has not yet been developed, then the decision needs to be made at this time as to what type of model would best describe the system. If it is a finite element model, then the physical properties of the system need to be gathered, or assumed and drawings made such that the system may be meshed and evaluated with an equivalent stimulus and response method. It is important at this point to ensure that the effects that the measurement sensors themselves introduce to the physical system are well represented in the model, as well to ensure that the comparison of real to theoretical is as close as it can be.

If the chosen model is the transfer function, then developing the model is less iterative as long as there is not a lot of cross coupling or interaction between multiple inputs or multiple outputs if they exist. For a linear transfer function model, the frequency domain output versus input gives a magnitude and phase (complex) response that can be fitted to the desired order analog or digital transfer function. Alternatively, the model may be kept within the time domain where it is somewhat easier to address non-linear response by maintaining higher order response terms – for example, terms that are a function of the square of an input variable.

Of concern in stimulus and response testing is that there must be some knowledge of the behavior of both the stimulator and the responder as imperfect entities – in other words, determine how the stimulator and the responder differ from the perfect process represented in the model testing.

Resolution of the Model

The work begins in resolving the model against the real system. This is often an iterative process. Once the model behaves approximately like the real thing under normal conditions, then the next step is to see how well the correlation holds for extreme or corner conditions of the operating envelope. That is, perhaps, boundary conditions are moved to the extremes of what may happen, or an intermittent source is added or removed. In this phase of model resolution, the robustness of the model is evaluated. And while some added deviation is expected and allowed, at no time should the model break down completely. If it does, then an important consideration is missing from the model and may have to be introduced.

How good does the model have to be?

The short answer is: good but not perfect. Looking at Figure 2, it is clear that there is still ample opportunity to iterate on the model should the subsequent stages show that the hardware behaves differently under control in the later stages of this development. In many cases, it is the judgment of the developer that makes the call. All the contaminant effects such as non-linearity and noise need to be evaluated for their strength so the problems that greatly influence the system can be conquered and the ones that the system is insensitive to can be ignored.

Conclusions

The efforts to accomplish the above appear to require a great deal of time and effort when compared to the trial-and-error approach to control development that is often used. The risk is always in whether the system to be controlled is simple enough that a quick-fix, off-the-shelf solution will work or not. In order to evaluate this risk, the complexity of the system needs to be assessed through testing. Often that leads one down the beginnings of the rapid development of control systems path illustrated above. One of the advantages of following this path is that you can devote as much or as little time desired to any stage. The advantage over the trial-and-error method is that incremental steps are taken that sequentially increase knowledge of the system so that, in the end, when you commit to hardware, you have the greatest chance of success and the least likelihood of having to scrap your system and start over.