SmartEdge Platform – Control

Tommy Mullane, Senior SoC Architect, Industrial ASIC BU

Contents

Introduction

There is clear distinction that needs to be made between management and control within the context of Edge Devices. Typically control needs to be realtime, whereas management is more intermittent and not taxed with realtime constraints. Considering the system latency from the Edge to the Cloud and back again, this is typically too great a delay for real-time control loops to manage. Hence real-time control invariably will need to remain at the Edge, whilst management, such as diagnostic checks and firmware updates can be done via the Cloud. This paper gives an overview of control systems that are typically used in an industrial context. It covers the actuators used and related AFE's (analog front ends), in addition to some examples of complete control loops.

In addition to calibration there are other schemes that can compensate for measurement inaccuracies, these will be discussed in the relevant sections below.

Actuator Overview

Actuators are the critical electro-mechanical devices on the control side of the loop. Quite simply they covert electrical signals into energy. Everyday examples of these are the ubiquitous speakers (where electrical signals are converted to sound) and screens (where electrical energy is converted to light). In the industrial segment, actuators are used to convert electrical signals to mechanical energy, typically to position or accelerate objects.

Industrial actuators are used for linear and rotary control. Non-electric actuators comprise of hydraulic (jack, brake, pump) or pneumatic control. Electrical actuators comprise of motors, solenoids and piezoelectric devices.

Motors comprise of AC, DC and Stepper types, the later which converts the incident electrical pulses into discrete mechanical movements. Regardless of type of motor, the fundamental working principle remains the same: when a current carrying conductor is placed in a magnetic field, it experiences a mechanical force.

Solenoids operate under a similar working principle as motors. An electrical current is passed through the coil, thereby creating a magnetic field that exerts force on the rod (plunger).

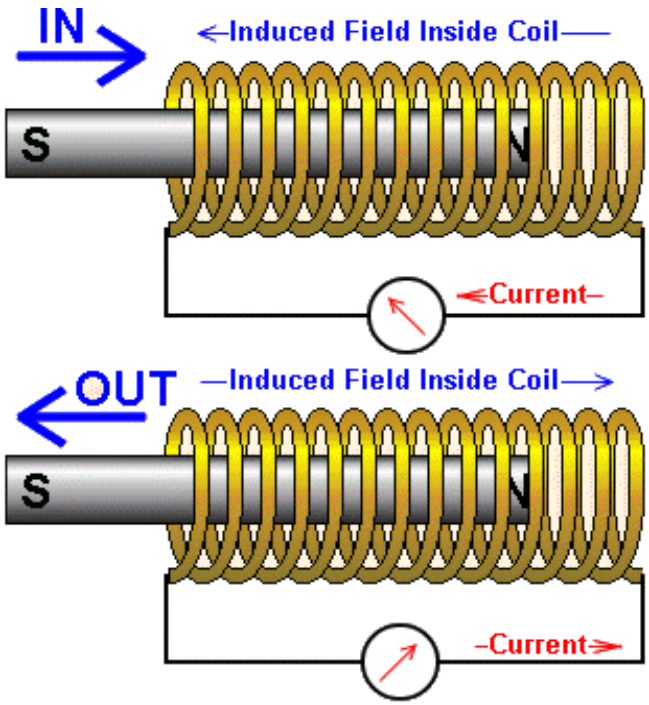

Figure 1: Solenoid current drive control

Solenoids can operate under voltage or current drive, where voltage levels can be 12V to 24V DC, or 110V to 230V AC. Applications for solenoids typically range from valve control, to relays to contactors and circuit breakers.

Piezoelectric devices operate under a different working principle. The application of voltage to a piezoelectric material causes mechanical strain and resulting displacement. Examples of such materials include quartz, and certain ceramics.

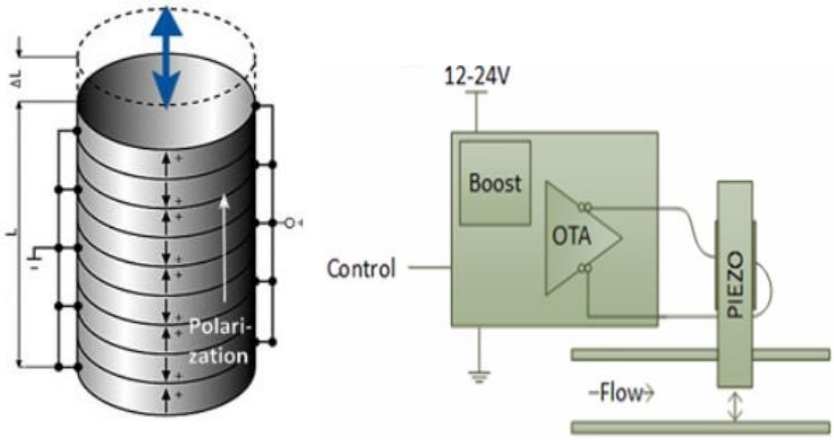

Figure 2: Stacked Piezoelectric device and illustrative drive circuitry

The advantage of utilizing piezoelectric actuators is the rapid response, precision movement and low power (there is no holding power required). However, they can require a high-voltage drive and exhibit high hysteresis. Typical applications for these devices include valve positioners, precision mechanics, and in the consumer space haptic feedback.

AFE's to drive Actuators

The key element in the Analog Front End (AFE) is the DAC, followed by the high-voltage buffer that is required to drive the high actuator voltages. The DAC requirements needed for control are included here:

- Typically, 10-14bit resolution
- < 1LSB DNL to guarantee monotonicity in the control loop
- 10μs is typically enough for settling time
- 100KHz conversion is sufficient
- Low power

Considering the relatively low conversion rate, a resistance-ladder DAC architecture is used for such requirements. Depending on the load characteristic and the related voltage range, a buffer is typically required between the DAC and the actuator.

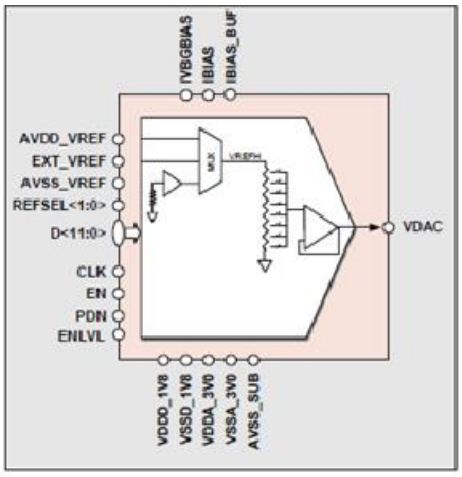

Figure 3: An example of a Renesas 12-bit control DAC IP block

An example of a control DAC driving an amplifier that drives the Piezo positioner in a value is given here, where 20dB additional gain is provided by external buffers in order to match the 24V range of the device.

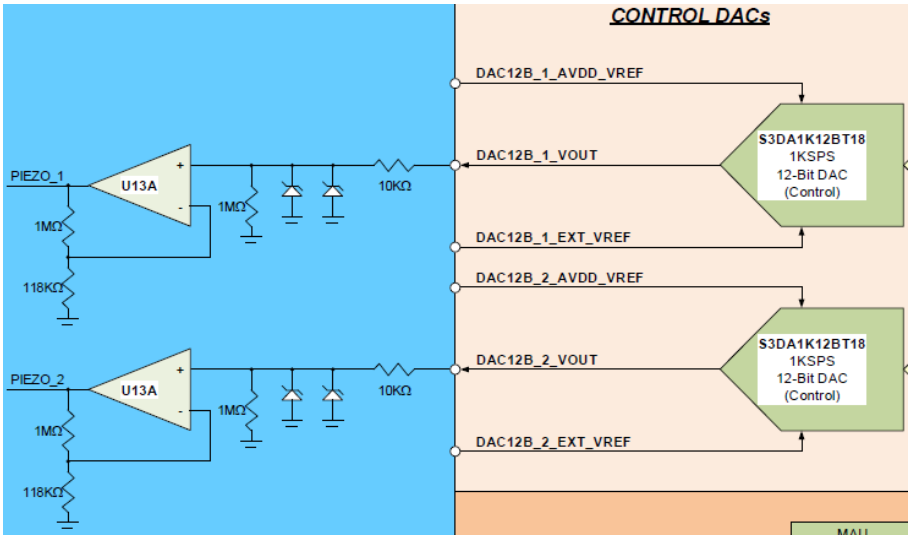

Figure 4: Dual channel drive consisting of 12-bit control DACs and PGAs

As an aside the other area where DACs are used in the current industrial control segment is in HART BUS requirement. Here 8-bit R-ladder DACs are sufficient, mated with on-chip transmit filtering.

Closing the Control Loop

Every control loop is different, but all loops tend to share the same elements. Those elements include sensing, measurement, analysis and response.

As an example, see the following valve control system:

Figure 1: System Positioning Algorithm for Logix 3400IQ Digital Positioners

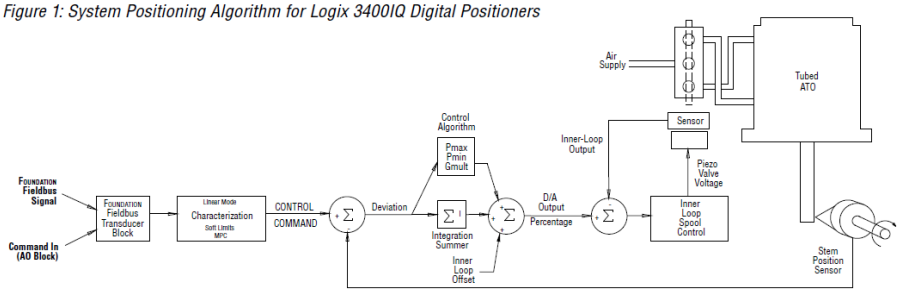

Figure 5: Valve control system

Here we can see a valve that is controlled using a compressed air system steered by a set of piezo controllers which in turn are controlled through the setting of a voltage. The system senses the pressure of the compressed air at various points and also directly monitors the position (rotation) of the valve and feed these inputs together into the control loop to reach the desired set point.

Systems like this with multiple sensor inputs provide interesting challenges in how to deal with possibly conflicting information from more than one sensor. For example, you may find that the position sensor is indicating that the valve has not reached the correct set point even while the pressure sensors are telling you that the compressed air system should have moved it where it needs to go. Balancing these competing information sources, scaling their importance and deciding on a decision tree of consequent actions, for if their information diverges is an interesting challenge, that must be understood, completed and encoded into the resulting chip in order to get the best system performance.

A more easily understood example might be that of temperature control of an environmental chamber used in the characterization of diode lasers.

The lasers need to be characterized under set environmental conditions. So their temperature needs to be controlled to within strict limits – roughly tenths of a Kelvin. The environmental chamber can be flooded with warm or cool air as required. The temperature of the devices is measured with a set of thermistors placed close to the devices themselves.

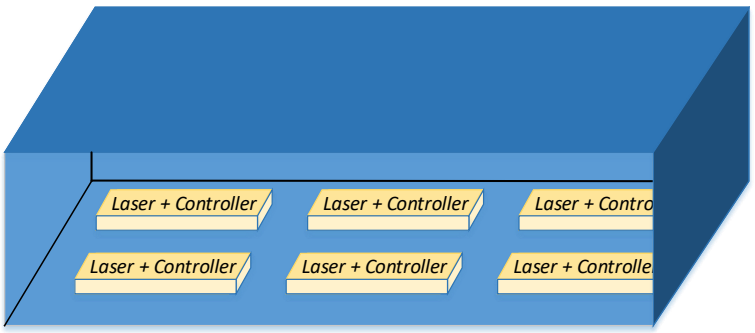

Figure 6: Thermal Chamber with lasers and controllers

So, we have a relatively simple system where we can measure temperature at a small number of points and have a single thing we can control – air temperature entering the chamber. The question is how we do his control?

First, we need to establish the difference between the temperatures we want and what we currently have – the temperature the thermistors are measuring. This gives us a temperature error for each thermistor. We can then combine these to minimise the worst-case temperature error by getting the root mean square of the error. This is the min/max algorithm and it will try to minimise the worst-case errors we are seeing.

We now have a single measurement of the temperature error and we can control the input air temperature of the chamber. The simplest control method is to multiply our error by a single number, let's call it P, and use the result as a difference we should apply to the air temperature set point. So:

$$Air\ Temperature\ Setting = P * Error$$

This will move the air temperature in such a way as to reduce the error. How quickly it reduces the error and how small it makes the error depends on the size of P. However, by itself this scheme will never be able to completely remove the error. To do that we need to add some concept of memory.

If as well as multiplying the current error by P we add in another term that sums the previous error over some window and multiples the result by I we get a PI controller:

$$Air\ Temp\ Setting = P * Error(0) + I * \sum_{n=-1}^{-N} Error(n)$$

where Error(n) is the error measured at time n.

Our temperature controller is probably not going to have to deal with fast moving changes in the temperature of the devices it is testing. But if it was we could also add a term that responds to the derivative of the error – a D term. This would then result in the classic PID controller.

Loop Bandwidth

Our control system will have an inherent bandwidth – the speed at which we can change the air temperature and at which the sensors will respond to change. However, if we do not pick out P and I terms correctly we will find that the system goes unstable. This can be understood by imagining what would happen if we had no I term and a very big P term, so:

$$Air\ Temperature\ Setting = P_{big} * Error$$

If we start seeing a small positive error (the devices are too cold) we will immediately turn up the air temperatures a lot. By the time we make our next measurement we find that we have gone past our desired temperature and overshot by an even bigger margin, so we now have a bigger negative error. We correct for this by changing to a much smaller air temperature.

Once again, we over shoot our desired temperature, this time going the other direction and we overshoot by an even larger amount. This process continues with us constantly overshooting the desired temperature and the error getting bigger and bigger, which in turn causes our control adjustments to get even bigger and bigger still. Eventually we hit the limits of what we can set the air temperature to and just go from max to min settings constantly oscillating around our desired temperature. This is the point at which you ask why you are employing the guy who designed your controller!

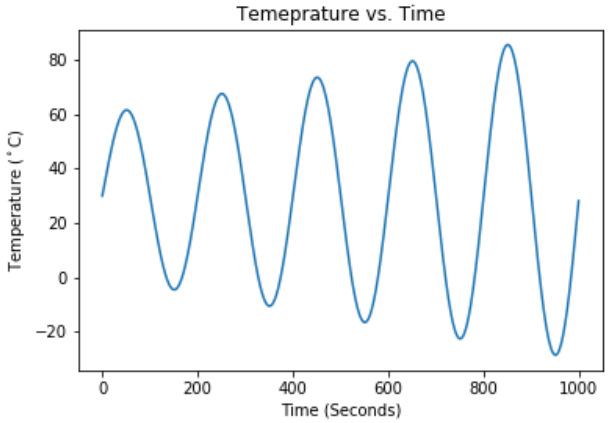

Figure 7: Temperature Vs Time for an unstable controller

Understanding the relationship between the loop bandwidth and the scaling factors (P, I and D) is crucial.

Lag

Lag is where it takes some time for a change in the actuator to produce a measurable change in what we are measuring. For example, if we set the air temperature of our environmental chamber to cool the air it might take a couple of second for its compressor-based cooler to kick in. This can also cause instabilities in our system as it means we are not observing the effect of the last change we have made in our control system.

Usually this can be overcome by delaying the loop bandwidth to a point where the loop update cycle is longer than the lag time. However, it can get more interesting in cases where the lag can vary.

For example, we might find that out environment chamber can usually adjust the air temperature very quickly. But periodically it needs to go through a dehumidification procedure in the compressor that means a delay in adjusting the temperature. In cases such as this it can be better to keep the loop bandwidth high, except when this procedure is ongoing where the loop should be effectively paused. That gives us a more reactive system in normal operation while not being adversely affected by the periodic increases in delay.

Loop Resolution

Unless there is some very specific reason, most control loops these days will be implemented digitally. Even very simple processors (PIC or Arm Cortex M0) can be integrated into a chip with the required sensor and actuators to implement the control algorithms.

However, depending on the accuracy required and on whether integration and derivative terms are being considered a greater resolution may be needed in control loop calculations.

For this reason and depending on what other functionality might be wanted on the chip it might be better to go for a more heavyweight processor (Cortex M3/M4). The M4 also offers a floating point unit which can help increase accuracy and speed of the calculations.

Whatever processor is selected, it can be integrated in the same die with an array of other analog and digital components. It is not unusual to include not just RAM and Flash with the processor and complete analog front ends for the actuator and sensing solutions but op-amps, power switches and other devices that would otherwise

need to be included on a board into the single chip in order to reduce component costs as well as reduce risks of end of life of different components.

Bringing it all together

An example of a single chip integrated solution is a controller we designed for a world leader in valve manufacture. They wanted a chip that would compute their complex control loops, implement their valve actuator system and take the required measurements with some high speed high resolution ADC technology. The whole solution had to meet very stringent environmental conditions, work in an intrinsically safe manner, comply with functional safety requirements and remain within a very modest power budget.

Renesas, formerly Dialog have used our in-depth knowledge of control systems to deliver a chip that met all these requirements at a compelling price point.

SmartEdge™ Platform

In conclusion, control loops can greatly simplify systems. Good design of the control of any system is important to prevent instability. While we have described some of the constituent building blocks and processes involved in understanding and controlling a relatively simple system, the same processes can and has been applied to systems off all sizes and complexity. The key is working with customers to thoroughly understand the issues for which a control system is being designed and tailoring the solution accordingly. Incorporation of all the elements that constitute the control loop can be readily integrated onto a single chip.

Renesas, formerly Dialog SmartEdge™ platform incorporates all the Sensor AFE (Analog Front End), Calibration, Control Loop, Communication and Security elements of a smart edge device, all integrated onto a single cost-effective ASIC chip. With more than 20 years' experience designing advanced embedded mixed-signal chips for hundreds of customers in every major region, Renesas delivers a new breed of design-centric semiconductor supplier capable of optimising its designs for every customer, yet achieving cost economies not thought possible with custom chips designs until now.

Revision History

Revision	Date	Description
1.0	Mar 01, 2018	Initial release
1.1	Dec 22, 2021	Re-brand