# Servo Control Systems 2: Digital Servomechanisms

Mark Readman control systems principles.co.uk

ABSTRACT: This is one of a series of white papers on systems modelling, analysis and control, prepared by Control Systems Principles.co.uk to give insights into important principles and processes in control. In control systems there are a number of generic systems and methods, which are encountered in all areas of industry and technology. These white papers aim to explain these important systems and methods in straightforward terms. The white papers describe what makes a particular type of system/method important, how it works and then demonstrates how to control it. The control demonstrations are performed using models of real systems designed by our founder and senior partner Peter Wellstead, and have been developed for manufacture by TQ Education and Training Ltd in their CE range of equipment. This white paper uses the computer based control and simulation tool CE300 to demonstrate a digital servomechanism for position control of a servo using an optical position sensor.

## 1. Introduction

Position control systems are an important component of many industrial products. Examples are found in disk drive’s, automotive products, robotics, process control and many others. The Control Systems Principles white paper Servo Control Systems 1 has described some of these applications and the basics of servo control. The first white paper was for direct current (DC) servos, however many modern servo control systems are implemented digitally using digital sensors. The aim of this white paper is to describe, in simple terms, how digital servos are implemented, using digital sensors and direct digital control system implementation. All the illustrations are done using the CE300 Logic Trainer.

In all servomechanisms one of the most important components is the position sensor. It measures the position of the servomotor and converts into an electrical signal that the control system can interpret and use. Digital optical position sensors are particularly important because they are non-contact and thus not subject to the wear and noise of analogue position controllers. A further advantage of optical sensors is that they can be used in severe environments where strong magnetic fields and RFI exist. Optical digital position sensors use a special coded disk to measure shaft position. Several different types of rotary shaft encoders used to measure relative or absolute shaft position, however a commonly used form of angular position encoding is based on the Gray code. In this white paper we will show how to control angular position using a Gray code shaft encoder. We assume that you have read the first Servo Control white paper, so that there is very little control theory in this white paper. Instead we concentrate upon the basics of digital servo implementation.

## 2. Position control

A basic position control system consists of a servomotor, position sensor and controller as shown in Figure 1. This block diagram is redrawn in Figure 2 to show how the position control system is implemented on the CE300. All the signals in Figure 2 are digital signals with only two levels, high and low. The shaft encoder outputs the position of the shaft in Gray code. This is then converted to binary using a decoder. A four-bit adder is used to compare the measured shaft position with the reference shaft position. The inputs to the adder are two four bit binary numbers. The adder is placed in subtract mode. So the output from the adder is a four-bit number representing the position error. Some logic is then used to decide when the motor should be turned on and in which direction the motor should go. When the error is zero then the motor is turned off. So the motor control signal is either on or off. This is called bang-bang control in the control literature because the size of the control signal is independent of the magnitude of the error.

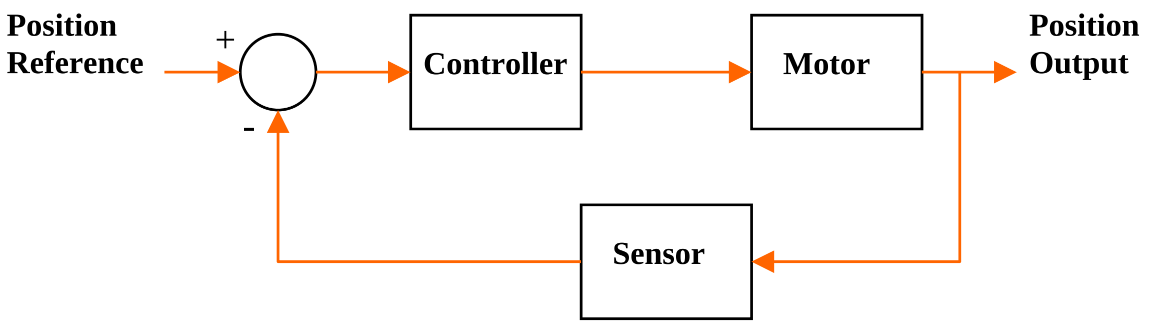

**Figure 1. Position Control System**

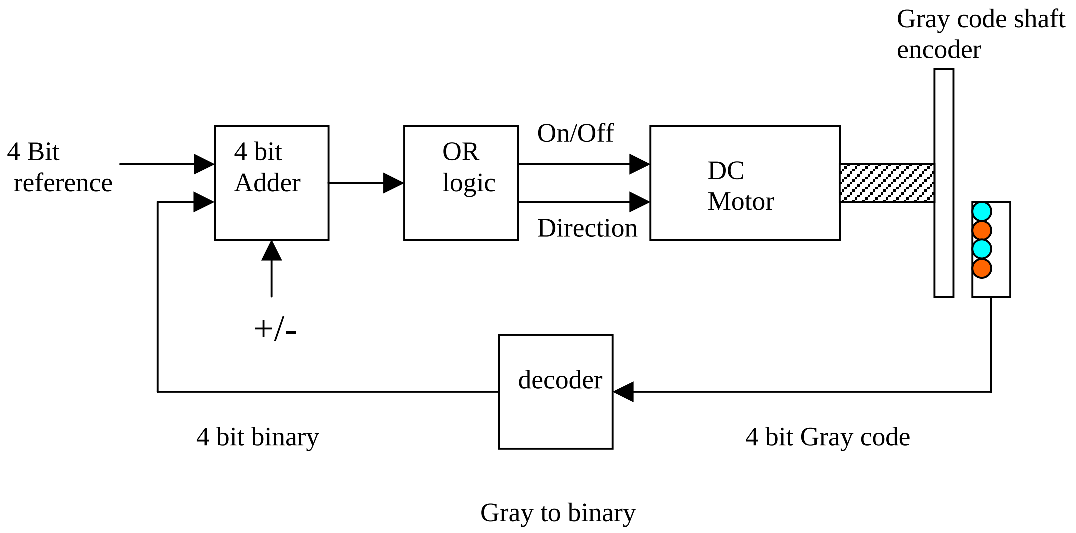

**Figure 2. CE300 Position Control System**

## 3. The optical shaft encoder.

Shaft encoders are transparent discs with the angular position digitally coded on (in black), so that by reading the code an optical sensor can determine the angular position of the shaft. In practice 12bit or 16 bit encoders will be used, giving high degrees of position accuracy, of approximately 0.1 deg and 0.05 deg resolution respectively. In our examples however we will use a 4 bit optical shaft encoder, as shown in the Figure 3. This is much less accurate than would be used in an industrial digital servo, but it is much easier to explain the operation with just four bits. The disk shown in figure 3 is the type supplied with the CE300. It is made from clear plastic with the Gray code shown in black. Gray coding is used because is that there is only a one bit change between any two adjacent positions of the encoder. This reduces the possibility of an error being introduced as the encoder is rotating. With alternative binary shaft encoder all the bits can change simultaneously between two adjacent positions. This introduces the possibility of errors occurring due to noise and timing differences in the logic. For example, as shown in Table 1 going from decimal 7 to decimal 8 causes all four bits to change in binary while only one bit changes in Gray code. Of course noise and timing errors are present when using Gray code however because only one bit changes at a time the chance of an error occurring is significantly reduced. The CE300 optical shaft encoder uses 4 bits to encode the position of the shaft. With just these 4 bits we can only encode 16 positions, so that the encoder resolution is 360/16=22.5 deg.

For use in a digital controller the output of the shaft encoder has to be converted to binary. This allows the error calculation to be done in binary. Fortunately, as shown in the next section, the conversion of a Gray coded position into Binary coded position is simple.

**Figure 3. CE300 Gray code shaft encoder**

| Decimal | Hex | Binary | Gray |
|---|---|---|---|
| 0 | 0 | 0000 | 0000 |
| 1 | 1 | 0001 | 0001 |
| 2 | 2 | 0010 | 0011 |
| 3 | 3 | 0011 | 0010 |
| 4 | 4 | 0100 | 0110 |
| 5 | 5 | 0101 | 0111 |
| 6 | 6 | 0110 | 0101 |
| 7 | 7 | 0111 | 0100 |
| 8 | 8 | 1000 | 1100 |
| 9 | 9 | 1001 | 1101 |
| 10 | A | 1010 | 1111 |
| 11 | B | 1011 | 1110 |
| 12 | C | 1100 | 1010 |
| 13 | D | 1101 | 1011 |
| 14 | E | 1110 | 1001 |
| 15 | F | 1111 | 1000 |

**Table 1: Number systems, showing how just one bit changes at a time**

## 4. Gray to Binary Decoder.

The conversion from Gray code to binary is performed with a simple logical decoder. The decoder logic uses exclusive OR gates connected as shown in figure 4 below. The block diagram shown in figure 4 was taken from the CE300 block library.

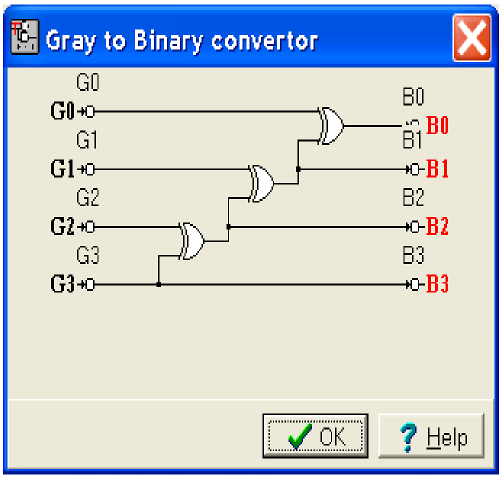

**Figure 4. Block Diagram of Gray Code to Binarv Decoder**

The block diagram expresses graphically the logic equations for translating Gray code into binary logic, where $\oplus$ is the logic symbol for XOR. The equations for this conversion for 4-bits are shown in equation 1. This equation set can be extended to any number of bits.

$$\begin{aligned} B_0 &= G_3 \oplus G_2 \oplus G_1 \oplus G_0 \\ B_1 &= G_3 \oplus G_2 \oplus G_1 \\ B_2 &= G_3 \oplus G_2 \\ B_1 &= G_3 \end{aligned} \qquad \text{(eqn.1)}$$

## 5. The summer.

The decoded binary shaft position is fed directly to an adder to be compared to the shaft position reference. The reference signal is also coded as 4-bit binary giving a possible 16 different reference positions. For reference purposes the traditional symbol for an adder is compared to the logic symbol used for a 4-bit adder in the CE300 in figure 5.

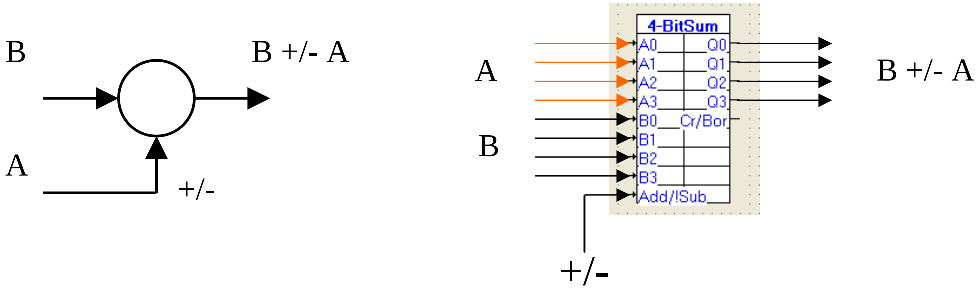

**Figure 5. Summer Symbol and 4 Bit Adder.**

## 6. Decision logic

In this form of digital servomechanism, the control signal is computed directly using logical (on/off or bang-bang) control as shown in figure 6. In this control scheme, the output from the adder is the position error signal in 4-bit binary. If this error signal is not zero then we want the motor to move. This is achieved in a logical controller by feeding the outputs from the adder to an OR gate, so that the output will only be zero if all the bits are zero, (e.g. the error is zero). The most significant bit Q3 is used to determine the direction of the motor. Note that in Figure 6 there are two OR gates used because the maximum fan-in is 3 for each gate on the CE300.

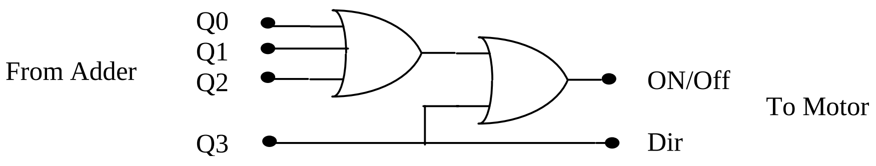

**Figure 6. Decision Logic.**

## 7. CE300 Implementation: Putting it together.

The complete digital position servo program is shown in Figure 7. A 4-bit binary reference position is generated using a digital source connected to each of the reference inputs B0 to B3. Toggling the digital sources generates the 16 reference positions. The actual shaft position is obtained from an input block and decoded into binary. The binary shaft position is then also fed to the adder.

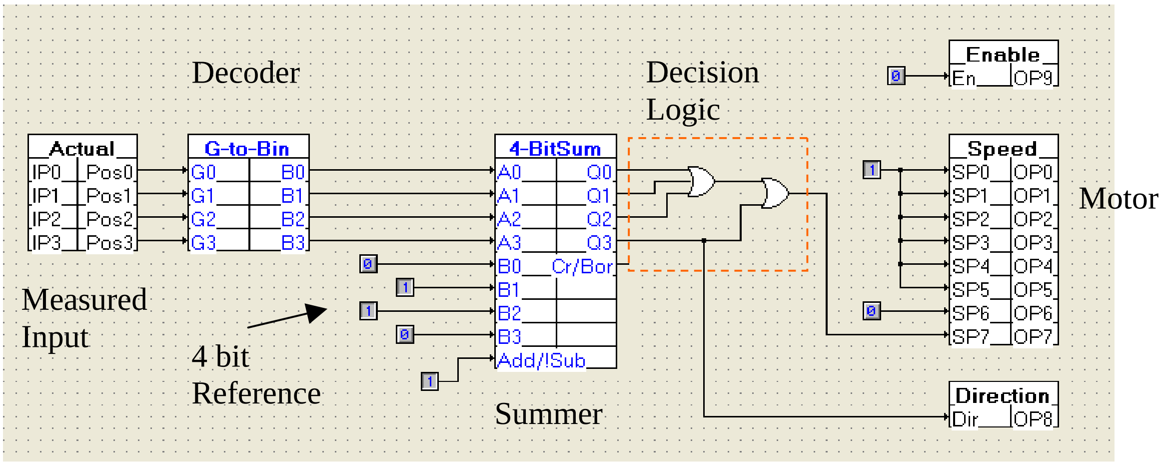

**Figure 7. CE300 Position Servo Control**

By adding 4-bit DA blocks the reference and output signals can be viewed on a chart recorder as shown in Figure 8. Here the reference signal for position is generated from a signal generator that is fed to a 4-bit AD converter to generate a 4-bit reference signal.

In Figure 8 the digital servo system is shown with the output following a simple step input. However, the response of the servo to different waveforms can also be explored. Moreover, it is possible to generate an approximate frequency response by varying the sine wave reference frequency. Of course a 4-bit implementation of a sine wave will look quite different to the continuous time version because of the quantization errors inherent in a digital controller.

In Figure 9 Light Emitting Diodes (LED's) are connected to the reference and measured signal to give yet another visual indication of the servo in operation. For any setting of the reference the LED's should match in the steady state. Four-bit digital to analogue converters have again been used to convert binary signals to analogue form for display on the chart recorder. The chart recorder shows the motor shaft (blue) tracking the reference (red).

As noted previously, the quantization error will be significant. Specifically, because we are using a 4-bit shaft encoder, an error of +/- approximately 11 deg is possible on the actual servo shaft, even though the binary signals match up. In figure 9 the green trace is the motor on/off signal. The motor speed can be increased by turning on SP0 to SP6. Effectively this is increasing the controller gain so now the motor responds faster to an error signal.

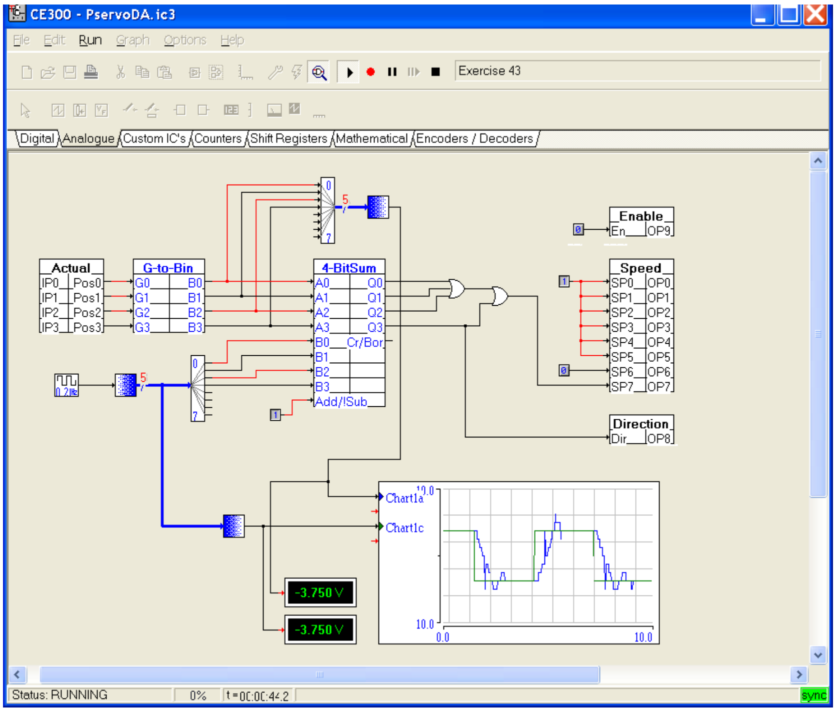

Figure 8. CE300 Position Servo Control with AD and DA

Figure 10 shows the digital servo control system when it is tracking a 0.2 Hz square wave and a 0.1Hz sine wave. The reference amplitude of 7.0 volts is plotted in red. Notice that the output from the optical shaft encoder is showing some noise. The lower part of the figure shows sustained oscillation when the servo reference is constant and set to the fastest speed setting. High gain setting causes a limit cycle and the motor shaft oscillates about the desired reference position. This is a commonly encountered feature of control systems and is called limit cycling. It is a generally undesirable phenomena in feedback systems. Even so limit cycle oscillations like this tell us something about the system. In this case the limit cycle is caused in part by the communication lag between the PC and the CE300 Logic Trainer.

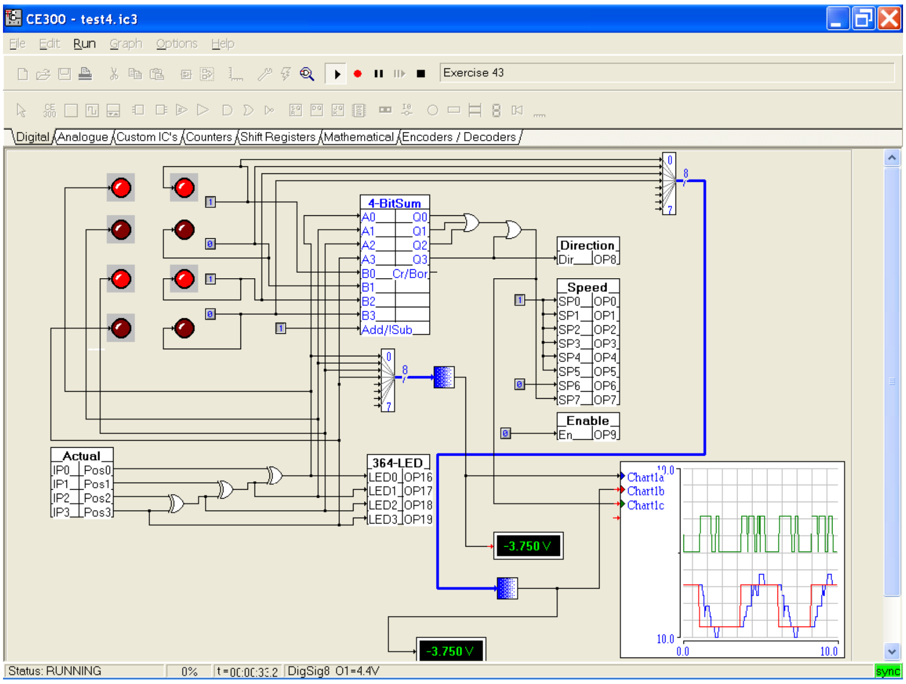

Figure 9. CE300 Position Servo Control with 4 Bit LED

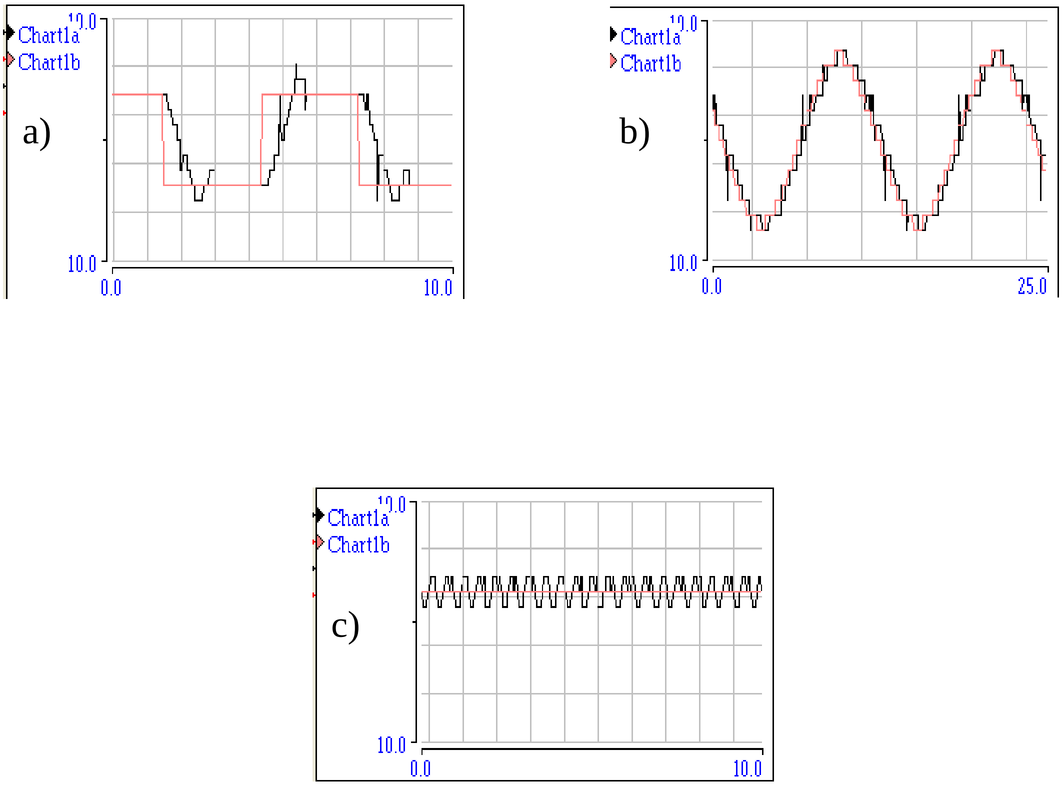

Figure 10. Position Servo Output for a) Step Response b) Sinusiodal Response c) Limit Cycle

## 8. A Simple Model

In this section we show a simple model (Figure 11) of the feedback system. This can be used as a starting point for analysis and simulation. The simulated results show in Figure 12. using this model are a reasonable representation of the results obtained with the CE300 servo. The time delay is set to 0.1 sec and the gain can be varied to simulate the effect of changing the motor speed. The numbers shown are approximate and will vary from system to system. In the above block diagram the gain K controls the motor speed. The time delay T is used to take into account communication between the computer and servo and other processing delays. For the CE300 equipment used for this white paper the values of K=10 and T=0.1 were used.

In the model a quantizer is used to simulate the effect of the 4-bit Gray code shaft encoder. The motor time constant of 0.4 seconds is again an approximation and the values used will be slightly different depending on the equipment used. Despite these approximations, the above simulation gives qualitative results that are similar to the actual servo. For example, the simulations corresponding to Figure 10. are shown in Figure 12 .

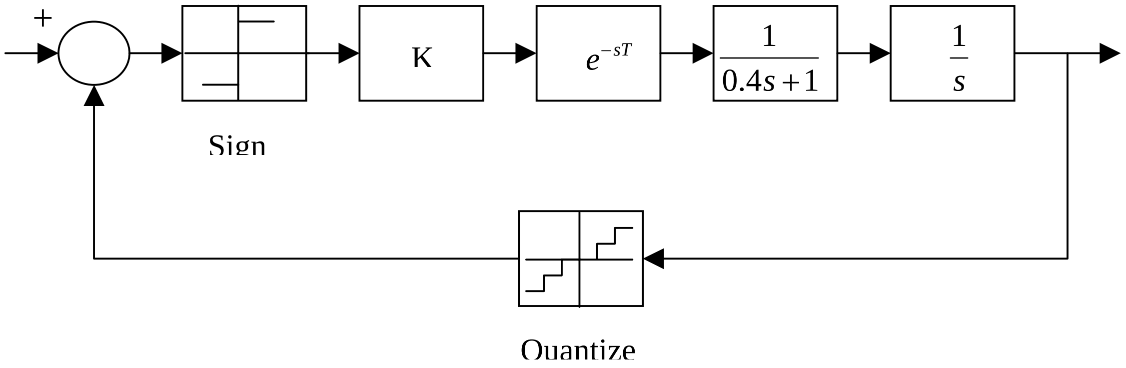

**Figure 11. Model of the CE300 Position Servo.**

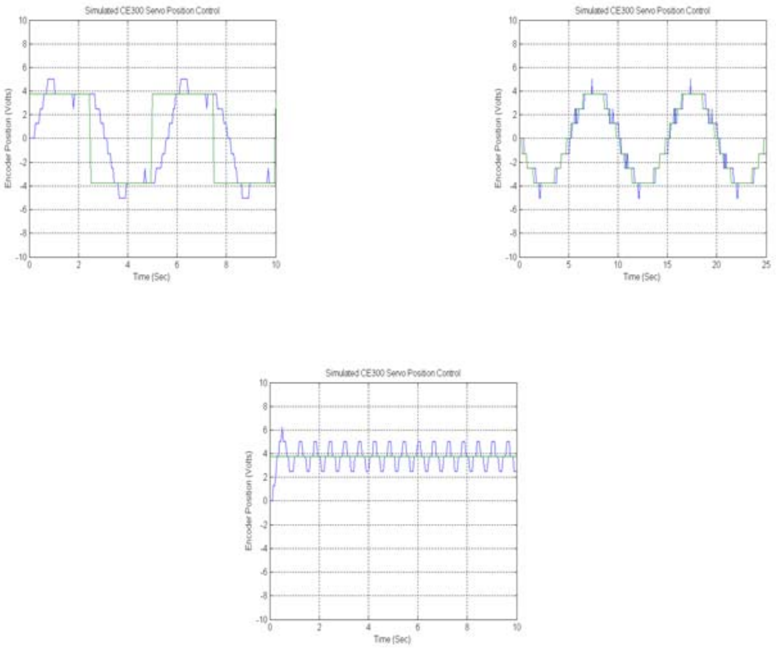

**Figure 12. Simulated output: a) Step Response b) Sinusiodal Response c) Limit Cycle**

## 9 Feedback control of an UP/Down counter

Dynamic systems where feedback is used to control a logic system are very common. In the example below we show how feedback can be used to control an up/down counter. The behaviour of this system is similar to the servo control example using a shaft encoder. We can think of the motor as a digital counter. As the shaft rotates the count increases of decreases depending on the direction of the motor. Thought of in this way we can use the same feedback circuit to control the count in a binary counter. Gray code is not used however because there is no shaft encoder.

A CE300 simulation to control an up/down counter is shown in Figure 13. The reference in this example is a signal generator fed to an analogue to digital converter. This provides a four-bit binary reference count. The output from the adder is fed to the OR gates in the same way as before, but now a clock is toggled on and off which allows the up/down counter to count. The direction signal is used in a similar manner to change the direction of the count. DA converter blocks are used to give a visual indication of the count and reference. The counter tracks the reference signal in a similar manner to the servomotor. (Note: All DA and AD blocks are set to 4 bits).

The clock speed in this simulation is equivalent to motor speed where

$$\text{clock speed} \approx 16\frac{\text{RPM}}{60}$$

and the motor speed varies from approximately 15 RPM and 40 RPM as measured at the output shaft. The motor speed can be measured by measuring the disk rotation period on each speed setting. With four bits we get the equivalent of 16 clock pulses per rotation. The main difference here is that the motor dynamics are no longer in the loop. This is sometimes called a discrete event system. Another common application using counter countrol is dual slope integrating AD converter. When the counter reaches the desired reference count it will be the equivalent binary representation of the reference input.

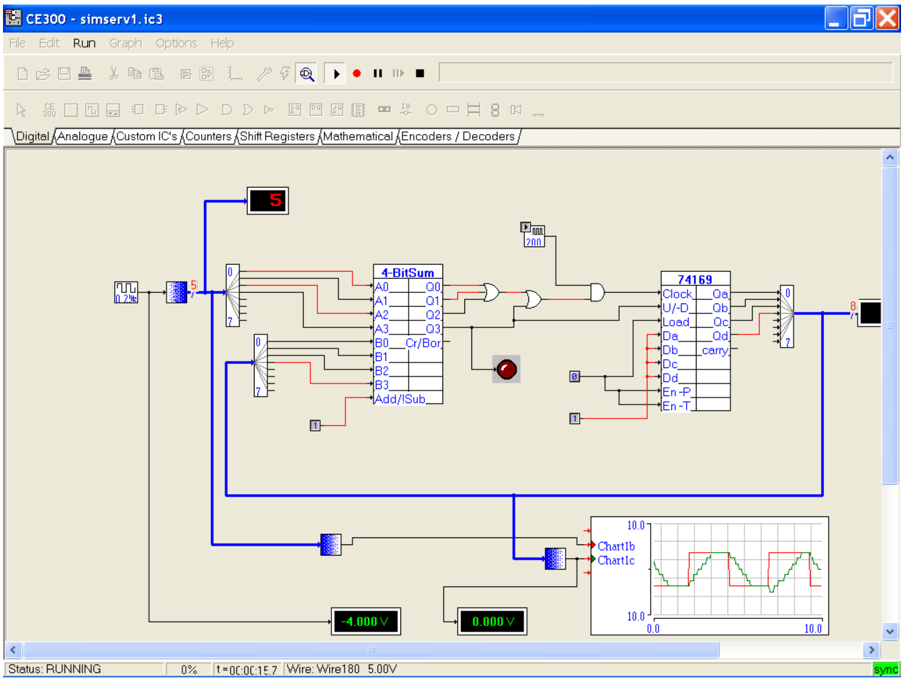

Figure 13. Feedback Control of an UP/Down Counter

## 10. Conclusions

In this white paper we have shown how the CE300 Logic Circuit Trainer can be used to control the CE300b position servo. A simple simulation model has been developed which can be used to explore the expected behaviour of the CE300 position servo control. We have deliberately not included to much theory as the aim of this white paper is to explain the logic techniques that are used in digital servomechanisms.

## 11. A Final Word

It is not possible to answer questions about our white papers, unless we have a contract with your organisation. For more information about the CE300 Control and Simulation Software go to the TQ Education and Training web site using the links on our web site www.control-systems-principles.co.uk or use the email info@tq.com. There are many books on position servo control and Gray code, and even more world wide web resources. We are particularly indebted to the references listed below

## 6. References

1. Dorf, R C and Bishop, R H, Modern Control Systems, (9th Ed) Prentice Hall, (New York), 2000.
2. M Healey, Principles of automatic control. (3rd ed.) English Universities Press, (London), 1975

## APPENDIX: The Experimental Set Up

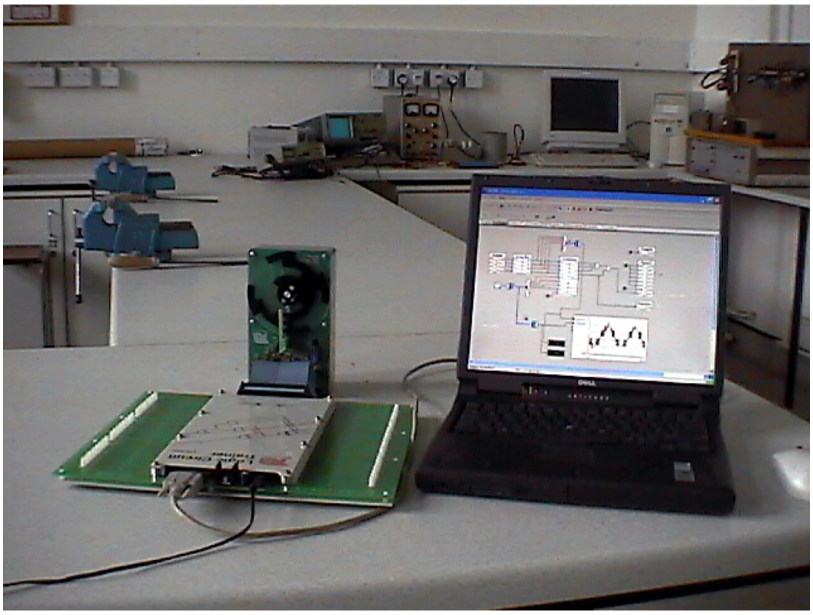

Figure 14. CE300 Logic Trainer with CE300b Servo.