Computer-based Optimal Control of Seeding Rate Based on Travel Speed and Seed Signals (Part 2)*

Laboratory Test of Proportional+Integral (PI) Control Algorithm

Pastor P. GARCIA*1, Nobutaka ITO*1, Koji KITO*1, Xiu Lun WANG*1

Abstract

This paper describes the method to control seeding rate proportional to the changes of the traveling speed using computer control system. The control algorithm is based on Proportional+Integral control method and the result of the laboratory test shows that the control system did respond immediately (about 1 second) after the traveling speed was changed from low to high or high to low however, the time required for the actual metering roll speed to be stabilized was about 3 to 4 seconds when the traveling speed changes from low to high and about 2 seconds when traveling speed changes from high to low. The desired metering roll speed was achieved within a reasonable number of sampling of the control program.

Keywords seeding machine, seeding rate control, computer control, programming, pulse modulation technique

I Introduction

The first part of this study described the basic features of a computer-controlled seeding rate based on travel speed and seed input signals. Equations to implement a proportional plus
integral control algorithm were established and the technique to vary the rotational speed of the metering mechanism using pulse-width modulation was discussed.

This second part of the study aims to describe the computer algorithm for gathering the digital inputs from the traveling speed and metering roll’s speed sensors and asynchronously sending the control output signal to the DC motor that drives the metering roll. The result of the laboratory tests of the PI control algorithm are presented and discussed.

II The Laboratory Set Up

The set up shown in Figure 1 looks very similar with Figure 1 described in part 1 except for the following changes as follows:

1. Computer
   Main controller used in the study was changed from 32-bit NEC PC-9801FX to a 32-bit IBM-PC, 133MHz with 80586 pentium processor.

2. Interface board
   Instead of the 12-bit Canopus, Analog Pro, A/D+Digital Interface Board, the four inputs and single output signals were directly connected to the 25-pin parallel printer port of the IBM PC. The printer adapter’s I/O port usage and the port addresses/pin connections of the four inputs and single output are given in Table 1 and Table 2, respectively.

Table 1 Printer adapter I/O port usage

<table>
<thead>
<tr>
<th>Port</th>
<th>Bit number</th>
<th>Pin No</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>378</td>
<td>*</td>
<td>9</td>
<td>O_Data bit 7</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>8</td>
<td>O_Data bit 6</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>7</td>
<td>O_Data bit 5</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>6</td>
<td>O_Data bit 4</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>5</td>
<td>O_Data bit 3</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>4</td>
<td>O_Data bit 2</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>3</td>
<td>O_Data bit 1</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>2</td>
<td>O_Data bit 0</td>
</tr>
</tbody>
</table>

Table 2 I/O port address and pin connections of inputs and output

<table>
<thead>
<tr>
<th>SIGNAL</th>
<th>ADDRESS</th>
<th>DATA BIT</th>
<th>PIN NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer</td>
<td>0x37A</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Travel speed</td>
<td>0x379</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>Metering roll</td>
<td>0x379</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>Seed counter</td>
<td>0x379</td>
<td>3</td>
<td>15</td>
</tr>
</tbody>
</table>

3. Metering roll speed sensor
A second photo interruptor sensor is directly connected to the shaft of the metering roll to determine its actual rotational speed. The Keyence photo sensor placed directly below the seed hopper is used to detect the presence or absence of seeds inside the hopper instead
of for counting the number of seeds that falls into the ground.

III Input and Output

There are four inputs and a single output used in the study which are all directly connected to the IBM-PC’s standalone parallel printer adapter with port addresses from 0x378 to 0x37A. This parallel printer port has 8 latched TTL output points, 5 TTL input points and 4 TTL points which are programmable for either output or input.

The parallel printer port has only one interrupt trigger. This interrupt trigger was initialized by writing 1 to data bit #4 of port address 0x37A and connecting the input signal to pin #10. In this study, the interrupt trigger was used for detecting the input signal from the external timer. The second input signal is from the first Sanyo photo interruptor sensor which is used to determine the speed of travel and connected to pin #11, data bit #7 of port address 0x379. The third input is from the second Sanyo photo interruptor sensor which is used to determine the actual rotational speed of metering roll and connected to pin #13, data bit #4 of port address 0x379. The fourth input is from Keyence photo sensor which is used to detect the presence of seeds inside the hopper and connected to pin #15, data bit #3 of port address 0x379. The specifications of the Sanyo and Keyence photo sensors are shown in Table 3.

The single output which sends the pulse signals from the computer to the DC motor controller is connected to pin #2, data bit #0 of port address 0x378.

IV Program Modules

The computer program to control the seeding rate is written and compiled using Microsoft C language, version 7. It is composed of six different modules as follows:

1. **Seeding.c**

This module is the user’s interface where control parameters such as the sampling time, initial duty cycle, and number of iterations are entered using the keyboard. This is also used to initialize the system. The data entry menu is shown in Table 4 below.

2. **Control.c**

This module includes the data gathering functions of the four inputs and the computation of the control output using the PI control algorithm.

3. **Pwm.c**

This module receives the control output from Control.c module and implements the pulse width modulation technique to vary the revolutiona...
the metering roll.

4. Timer.c

An external timer is used with a tick frequency set at 250Hz. From this single timer, this program module was developed to implement two countdown timers synchronized with the external clock frequency. Included in this timer module is a function \[\text{SetTsr( )}\] which sets up the time-out service routine that turns "on and off" the output pulse signal to control the DC motor.

5. Xignal.lib

A general purpose interrupt handler library called "Xignal.lib", developed by Auslander and Than\(^5\), is used to set up Chronos ( ), a function symbol written in Timer.c module, which run whenever a timer interrupt is detected. This is linked to the object files of the other modules during compilation.

6. Alarm.c and Panic.c

The Xignal.lib software takes over the interrupt handling function of the computer and alters its default interrupt setting. Thus, at the end of seeding operation (i.e. the tractor stops) or during troubles (i.e. no more seeds in the hopper, etc.), the program cannot just automatically exits and go back to the system but the computer must be returned back to its default interrupt settings otherwise it will hang up. To do that, these modules are called.

V Timer and Pulse Frequency Selection and Speed Calibration

1. The role of Chronos ( )

Chronos ( ), as defined earlier, is a function symbol which is used to synchronize the two software timers to the external timer’s frequency and it is installed using the Xignal.lib software. Whenever an interrupt is detected from the external timer, it suspends the execution of the control program, run the clock counting function, invoke the time-out service routine if a software timer has timed out and such a routine is installed, and resume execution of the suspended program.

2. Selection of timer and pulse frequency

The selection of appropriate timer and pulse frequencies depends upon the number of inputs and outputs to be processed and the speed of the computer’s central processing unit (CPU). In this study, different timer frequencies starting from 1000Hz and pulse frequencies starting from 20Hz were tested using trial and error. Frequent hang-ups and crashes were experienced when the timer frequency and pulse frequency were higher. It may be because there were just too many inputs and output signals to be attended to by the CPU at one time, thus it crashed. At lower timer frequencies (i.e. 50Hz), the metering roll rotated not so smoothly and in jolting movement especially at lower speed where the "ON" time was very small compared to the "OFF" time of the pulse signal. Finally a good combination of 250Hz timer frequency and 10 Hz pulse frequency was found where the program ran smoothly without crashes and hang-ups.

3. Metering speed calibration

Using the selected timer and pulse frequencies, a calibration test was conducted to find out the corresponding metering roll’s speed at different duty ratio (i.e. from 0.04 to 0.95). The result shown in Figure 2 indicates
that the revolitional speed of the metering roll is indirectly proportional to the inverse transform of the duty ratio. When the inverse of the duty ratio decreased from 25.0 to 1.056 the metering roll’s speed increased from 23.25 to 85.75 revolutions per minute. Since the calibration test was conducted with the actual load of the DC motor being connected, the control error for the DC motor during the laboratory test was minimum.

VI Determination of Proportional and Integral Constants

The Proportional ($K_p$) and Integral ($K_i$) constants used in the study were determined using Reaction Curve method. In this method it was assumed that the transfer function of the control object is expressed in first order system ($T$) plus time loss and/or dead time ($L$). The equation for solving the two constants are shown in Table 5.

First, the $K_p$ and $K_i$ were assigned an assumed value of 0.50 and 0.60, respectively and the program was ran for 20 sampling periods with 2 seconds sampling time. The required metering roll’s speed was programmed to start at 24 rpm and abruptly changed to 50 rpm after three sampling periods. From the result of the test shown in Figure 3, the values of $T$ and $L$ were estimated to be around 7.60 and 2.50, respectively. The values of $K_p$ and $K_i$ were calculated and substituted to the general equation for PI control.

Another series of tests were conducted to fine tune the value of $K_p$ and $K_i$, and finally a suitable value of 0.31 and 0.819, respectively, were found. The step response curve of the
control function using the selected $K_p$ and $K_i$ values is shown in Figure 4.

**VII Seeding Rate Control**

The general flow of the control program is shown in Figure 5 and the different formulae and nomenclatures are shown in Table 5. The required revolution speed of the metering roll was computed based on the travel speed, desired seeding rate and the calibration constants of the metering rate. The actual revolutional speed of the metering roll was directly determined using the second photo interruptor sensor connected directly to the shaft of the metering roll. Before computing the duty ratio, the control output (i.e. $m$) is tested for its lower and upper limits (i.e. 23.25 and 85.75 rpm, respectively). In the same manner, before the new computed duty ratio is sent to Pwm.c module, it is tested first for its lower and upper limits (i.e. 0.04 and 0.95, respectively).

The lower and upper traveling speed limits were set at 0.16m/sec and 0.50m/sec, respectively. Below the lower speed limit, the tractor is considered to be in stopped condition. Over the upper speed limit, the metering mechanism cannot deliver anymore the required amount of seeds per linear meter because of the speed limitation of the metering roll (i.e. up to 85.75 rpm only).

Before reading the signal to get the actual speed of metering roll, the presence or absence of seeds inside the hopper is checked first using signal coming from the Keyence photo sensor installed below the metering roll. If no seed signal is detected after one sampling period of two seconds, this assumes that there are no more seeds in the hopper. In this case or in other troubles like if the tractor stops, the program automatically gives a beep sound and a warning printed on the screen before going back to the system. In actual application, some type of alarm can be mounted for giving warning to the operator.

**VII Laboratory Test**

1. **Methodology**

A test run of the system was conducted in the laboratory to determine the response time. The source of the travel speed signal was from the photo interruptor sensor mounted over a slotted disc directly connected to the shaft of a variable speed AC motor described in part 1. The traveling speed was computed after 25 pulse signals had been gathered and a 0.60m tractor’s wheel diameter as well as a 4 percent wheel slippage were assumed.

The speed of the metering roll was determined using the same type of photo interruptor sensor used in travel speed. It was computed after 10 pulse signals had been gathered. Both the traveling speed and metering roll’s speed pulse signals were gathered using “pooling” technique.
The total test run was 78.776 seconds with a sampling time of two seconds. The traveling speed was set first at low speed of around 11 m/min (0.18m/sec) and increased to 25m/min (0.417m/sec) and finally lowered down gradually until it stopped. The initial duty ratio was set at minimum (i.e. 0.04). The red bean seeds were used and the seeding rate per meter was assumed to be 3.12 grams. The data were gathered using the computer itself by means of a special function call that simultaneously saves the actual reading of travel speed (m/min) and metering roll's speed (rpm) into a created data file while the laboratory test is going on.

2. Results and discussion

Figure 6 shows the laboratory test result. The graphs of the required and actual metering roll’s speed indicated that the control system did respond immediately (about 1 second) after the traveling speed was changed from low to high or high to low. However, the time required for the metering roll to be stabilized was about 3 to 4 seconds when the traveling speed changes from low to high and about 2 seconds when traveling speed changes from high to low. This difference in the required time to stabilize could be due to the mass of the metering roll. It means that it took more time to accelerate the metering roll than to decelerate it.

The graph of the traveling speed showed that the speed started from about 0.18m/sec and after 1 second had elapsed, it abruptly changed to about 0.40m/sec. After that the speed seemed to changed constantly although in reality the motor speed was held constant until around 7 seconds had elapsed. After that the speed gradually slowed down until it stopped. This variation of the traveling speed reading could be due to the less precision of the timer frequency being used in this study (i.e. 250Hz). It means that the time divisor (converted to minutes) in the formula for computing the speed of travel could have varied in the range of +/-4 ms. Thus, the final reading of metering speed also varied although it is supposed to be constant.

As a result of the variation of the traveling speed reading, the value of the required metering roll’s speed had also varied constantly. The combined effect of this constant variation of the required metering roll’s speed and the less timer precision used in computing the actual metering roll’s speed, had caused the actual reading of the metering roll speed to vary in the ranged of around 10 rpm.

The suggested solution to solve this condition are as follows:

a) Use of more precise timer frequency (i.e. 1000Hz and above);

b) Use of computer with faster processor so that 1 sampling cycle of the program can be completed in the least possible time (i.e. less than 1 second); and,

c) Use of interrupt technique instead of pooling technique in gathering the pulse signals from the traveling speed and metering speed photo interruptor sensor.

Fig. 6 Laboratory test result of seeding rate control system

IX Conclusion

This paper described a method of controlling seeding rate proportional to the changes in
traveling speed using computer control system. The desired metering roll speed was achieved in the laboratory test within a reasonable number of sampling of the control program. Some improvements of the system however, are still needed before actual field testing of the control system. Field testing is necessary to evaluate the response time of the control system to changes in traveling speed. Research may also be necessary to find better alternatives to some of the components used.

References


(Received : 11 Nov. 1997· Question time limit : 31 Jul. 1998)