Description of Shaft Encoder Construction and Testing

Louis Brandy

The part chosen for the encoder was the Hamatsu P5587 photoreflector. The device consists of an IR emitter and a phototransistor pair. It is a 5-pin device with the following layout:

Using a pull-up resistor, the device will be at 5-volts if black paper is in front of it, and 0V if white paper is in front of it. Using this simple property, a shaft encoder can be constructed. The following is the actual design implemented on the circuit board that was tested:
The 330 ohm resistor simply regulates the current that flows through the diode. The 3.3k resistor is simply a pull-up resistor and it’s value has very little effect on performance of the digital output.

The second part of the encoder is the encoder disk. These are circles divided in equal slices of alternating black and white. Mounting these disks to the wheel, with the photoreflector fixed in place and facing the wheel creates the encoder. Below is a reproduction of one:

The actual encoder disks were made in auto-cad. Three resolutions were made for experimenting: a 16 section (above), a 32 section, and a 64 section.

**Testing**
The first test done on the shaft-encoder was to hook it up to an oscilloscope and see what kind of signal it gave. The signal appeared quite crisp, but it was near impossible to tell what types of noise or bounces were occurring at the extremely slow frequency. This, at the very least, verified the encoder was at least wired correctly.

Again, there is no easy way to test the frequency of the signal and compare that with the angular speed of the wheel. The best way to measure if the shaft-encoder is doing its job is to actually write the software to make the robot move a certain amount of ticks and measure the distance traveled. Over the course of 20 trials, the distance the robot traveled after two wheel-rotations was measured. The distance was chosen after a bit of trial and error. Having the distance too long would lead to errors due to motors not being the same speed, and having the distance too short amplifies the simple measurement errors. These numbers were compiled and the mean, the standard deviation, and the range were calculated. This test was done for the 16, 32, and 64 sectioned disks.

What follows is the tabulated data of the three tests done (All numbers in inches):

<table>
<thead>
<tr>
<th>16 section</th>
<th>32 section</th>
<th>64 section</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.1875</td>
<td>26.25</td>
<td>26.375</td>
</tr>
<tr>
<td>26.375</td>
<td>26.25</td>
<td>26.375</td>
</tr>
<tr>
<td>26.375</td>
<td>26.25</td>
<td>26.4375</td>
</tr>
<tr>
<td>26.5</td>
<td>26.375</td>
<td>26.4375</td>
</tr>
<tr>
<td>26.5</td>
<td>26.4375</td>
<td>26.4375</td>
</tr>
<tr>
<td>26.5</td>
<td>26.4375</td>
<td>26.4375</td>
</tr>
<tr>
<td>26.5</td>
<td>26.4375</td>
<td>26.4375</td>
</tr>
<tr>
<td>26.5</td>
<td>26.4375</td>
<td>26.4375</td>
</tr>
<tr>
<td>26.5</td>
<td>26.5</td>
<td>26.5</td>
</tr>
<tr>
<td>26.5</td>
<td>26.5</td>
<td>26.5</td>
</tr>
<tr>
<td>26.5</td>
<td>26.5</td>
<td>26.5</td>
</tr>
<tr>
<td>26.5</td>
<td>26.5</td>
<td>26.5</td>
</tr>
<tr>
<td>26.5</td>
<td>26.5</td>
<td>26.5</td>
</tr>
<tr>
<td>26.5</td>
<td>26.5</td>
<td>26.5</td>
</tr>
<tr>
<td>26.5</td>
<td>26.5</td>
<td>26.5</td>
</tr>
<tr>
<td>26.5</td>
<td>26.5</td>
<td>26.5</td>
</tr>
<tr>
<td>26.5</td>
<td>26.5</td>
<td>26.5</td>
</tr>
<tr>
<td>26.5</td>
<td>26.5</td>
<td>26.5</td>
</tr>
</tbody>
</table>
What follows is some secondary calculations about the datasets:

<table>
<thead>
<tr>
<th></th>
<th>16 section</th>
<th>32 section</th>
<th>64 section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean:</td>
<td>26.55</td>
<td>26.475</td>
<td>26.5125</td>
</tr>
<tr>
<td>Std.Dev</td>
<td>0.153897</td>
<td>0.118932</td>
<td>0.094242</td>
</tr>
<tr>
<td>Range</td>
<td>0.6875</td>
<td>0.4375</td>
<td>0.375</td>
</tr>
<tr>
<td>Theoretical</td>
<td>0.834461</td>
<td>0.41723</td>
<td>0.208615</td>
</tr>
</tbody>
</table>

Much as expected, as the resolution increases, the standard deviation and total range go down. The theoretical values are obtained by dividing the circumference by the number of sections. In theory, the encoder knows nothing about anything smaller than this number and it’s provided as an ideal number for comparison. Two other sources of error keep the estimated numbers from being much closer to the actual values observed. First, there is measurement error because this is a difficult measurement to do to a high degree
of precision. Secondly, the starting and stopping of the robot is very suddenly and jerk
and often causes slight shifts and overshoots.

The final robot used the 32 section version because the 64 version proved to be too error-
prone in various noisy conditions and especially on reflective floors (ie, tile). With a
better covering to ensure little outside interference, the 64 version would probably work
extremely well as well.

Sample Code

TMSK2  EQU     $1024
TFLG2  EQU     $1025
PACTL  EQU     $1026
PACNT  EQU     $1027

* Number of Pulses before interrupt
* Negative, in 2's compliment

NUMP   EQU     $E1

ORG     $0000
PCOUNT  RMB     1   * Memory location saves current count

ORG     $00CD
JMP     PAO_ISR

ORG     $2000
LDS     #$1FF
LDAA    #$00
STAA    PCOUNT
LDAA    PACTL   * initializations required (Your input pin may vary)
ORA     #%01010000
ANDA    #%01011111
STAA    PACTL
LDAA    TMSK2
ORA     #%00100000
STAA    TMSK2
LDAA    #$FF
STAA  PACNT

CLI

HERE  BRA  HERE

* In here, put code to
• load PCOUNT, and do something with it
• Ie, print it out

*****************************************************************************

PAO_ISR  LDAA  #00100000
STAA    TFLG2    * ISR to catch inputs

LDAA    PCOUNT
INCA
STAA    PCOUNT
JSR    S50E

LDAA    PACTL
ANDA    #01000000
BEQ    PAO1

*Put 0 in here
LDAA    PACTL
ANDA    #11110000
STAA    PACTL
BRA    PAO2

PAO1
*Put 1 in here
LDAA    PACTL
ORA    #00010000
STAA    PACTL

PAO2  LDAA  #$FF
STAA    PACNT

RTI