# SYSTEMS MOVEMENT DATA ACQUISITION

**Cătălin-Iulian CHIVU**
Transilvania University of Brasov, Romania

**Abstract.** In the present paper, author introduces a system that is used to plot the experimental characteristics of rotational or linear mechanisms movement. The system presented here has an incremental transducer, a microcontroller connected to the computer through RS232 interface and software of data processing and graphic representation of the acquired data.

**Keywords:** pneumatic muscle, microcontroller, characteristics

## 1. Introduction

The system presented in this paper is used to plot the longitudinal displacement of a pneumatic muscle with controlled input flow that actuates a mechanism for load control. The system includes also a force sensor [1]. The maximum displacement of the open end of pneumatic muscles is proportional to the total length of the muscle [2]. This aspect enforced the author of the present paper to create a flexible system to determine the open end displacement of the muscle independently on the muscle constructive parameters, with high precision and that do not change the mechanical characteristics of the system by introducing perturbations as supplementary weight, inertia or friction forces. The system presented here was used in a laboratory to plot mechanical characteristics of pneumatic muscles.

## 2. System description

### 2.1. Incremental transducer

The incremental transducer is made of an integrated incremental sensor, which has on one side of it the infra-red light source (an IR LED), and on the other side a complex optic transducer with two channels (figure 1). The integrated incremental sensor may be used in two different ways: either it is fixed, a transparent disc with equiangular opaque slots rotating in front of it, or mobile, translating along a transparent band that has equidistant opaque slots. The number of slots/mm determines the sensor resolution. For the application presented in this paper was used an incremental transducer similar to one used in an ink-jet printer, with 1200 step/inch resolution, which represents almost 50 step/mm (47.244 steps/mm). The transducer is supplied to 5V and has two outputs for each channel, both of them, CMOS and TTL compatible.

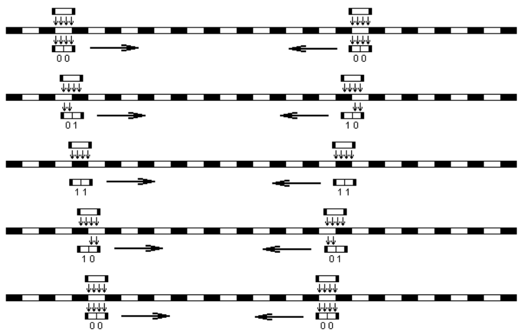


Figure 1. Functionality of the incremental transducer

The two outputs may be connected to input of the microcontroller [3]. The two outputs of the transducer must be decoded. Decoding process is done using an automat, which counts the steps made by the transducer, also considering the movement direction, according to Table 1. The automat output is an absolute numeric value that corresponds to the instantaneous position of the transducer.

Table 1. Binary code of the incremental transducer

| | Left movement | Right movement |
|---|---|---|
| Logic values of the two channels | 0 0 | 0 0 |
| | 1 0 | 0 1 |
| | 1 1 | 1 1 |
| | 0 1 | 1 0 |

### 2.2. Decoding automat

The decoding automat has two binary inputs (from the two channels of the incremental transducer) that never change in the same time, because the two optic sensors are one step out of phase [4]. Figure 2 shows ten states of the incremental transducer, five consecutive movements to the right and five consecutive movements to the left. IR transmitter is positioned in the upper side of the slotted tape and in the lower side there is the double two channels IR receiver. The outputs of the two channels will be logic 0 if they will receive light from the transmitter and, respectively logic 1 if they don't.

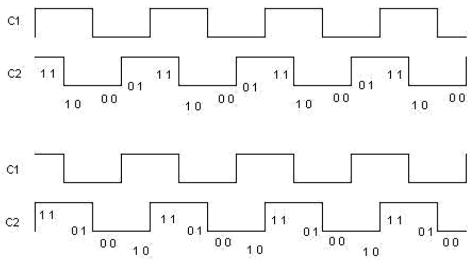

Figure 2. Logic states sequence of the two outputs

The two channels bits are used as inputs of the decoding automat. The automat has a associated state to each phase of the incremental sensor, thus being defined four states, noted by S0, S1, S2 and, respectively S3 that correspond to 00, 10, 11 and 01 binary code. To determine the next state, the decoding automat should consider both, two inputs and current state.

Based on the above considerations can be underlined the functional rules of the decoding automat:

- if the current state is S0, corresponding to 00 input binary code, then the state may be:
  - S0 if no input bit changed (from S0 to S0 state through T00 stationary transition);
  - S1 if first input bit is changed from 0 to 1 (from S0 to S1 through T01 transition);
  - S3 if second input bit is changed from 0 to 1 (from S0 to S1 through T03 transition);
- if the current state is S1, corresponding to 10 input binary code, then the state may be:
  - S1 if no input bit changed (from S1 to S1 state through T11 stationary transition);
  - S2 if the second input bit is changed from 0 to 1 (from S1 to S2 state through T12 transition);
  - S0 if first input bit is changed from 1 to 0 (from S1 to S0 state through T10 transition);
- if the current state is S2, corresponding to 11 input binary code, then the state may be:
  - S2 if no input bit changed (from S2 to S2 state through T22 stationary transition);
  - S3 if first input bit is changed from 1 to 0 (from S2 to S3 through T23 transition);
  - S1 if the second input bit is changed from 1 to 0 (from S2 to S1 state through T21 transition);
- if the current state is S3, corresponding to 01 input binary code, then the state may be:
  - S3 if no input bit changed (from S3 to S3 state through T33 stationary transition);
  - S0 if the second input bit is changed from 1 to 0 (from S3 to S0 state through T30 transition);
  - S2 if first input bit is changed from 0 to 1 (from S3 to S1 through T21 transition);

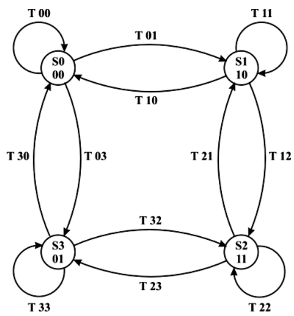

Figure 3. The transition diagram of decoding automat

Figure 3 presents the transition diagram of the incremental sensor decoding automat, based on which are concluded:

- the T00, T11, T22 and T33 are static transitions because the automat is passing from one state to the same state, as result of no input bit changes. This is possible when during two consecutive samples of the two input bits there were no change of these bits, meaning that the incremental sensor didn't move one step (it didn't move at all or it did a small movement). Thus, the static transitions pass without any changes upon the numeric value of the sensor (position counter is unchanged);
- the T01, T12, T23 and T30 are dynamic transitions, because the automat is passing from a state to a different one as result of one input bit changes. This is possible when the incremental sensor, displaced more than a step. In this case, one and only one of the two input bits changed its logic state during the period between two successive sequences of the bits, so the incremental sensor displacement exceeds the boundary between two successive steps. One of these transitions may appear when the

sensor is moving in the conventional positive sense. In this case the numeric value associated to the sensor position always increase with one (the position counter is incremented);

- the T03, T32, T21 and T10 are also dynamic transitions, because the automat is passing from a state to a different one as result of one input bit changes. This is possible when the incremental sensor, displaced more than a step. In this case, one and only one of the two input bits changed its logic state during the period between two successive sequences of the bits, so the incremental sensor displacement exceeds the boundary between two successive steps. One of these transitions may appear when the sensor is moving in the conventional negative sense. In this case the numeric value associated to the sensor position always decrease with one (the position counter is decremented);
- transitions between S0 and S2 state, from S2 to S0 state, between S1 and S3 and from S3 to S1 state are forbidden transitions and thus there aren't represented on the transition diagram of the position transducer decoding automat. Even there are forbidden transitions they appear when, during two consecutive samplings, the two input bits both changes. This is the case when the incremental transducer velocity is higher than the maximum velocity of the decoding automat (incremental transducer moved during two successive sampling more than one step, crossing in this way two bounders between steps). Decoding automat of the incremental sensor do not particularly analyse these transitions because in these cases it can't clearly detect the two movement direction of the transducer. Thus, it can appear errors given by the difference between the real position of the transducer and the numeric value associated with the transducer position. The maximum speed for a correct operation of the incremental transducer is measured in steps per second and depends on the sampling frequency of the automat binary inputs.

The decoding automat of the incremental transducer was implemented in the microcontroller program based on hardware interruptions. The two outputs of the incremental sensor (corresponding to the two channels) were connected to one of the microcontroller port through two pins configured as inputs that support hardware interruptions. Thus, on each change of any of the two binary inputs state is generated a hardware interruption that determines execution of one specific program subroutine. There were defined two subroutines, associated to the two hardware interruptions, one for the change of the first input bit and the other for the change of the second input bit. The two hardware interruption subroutines are very similar as structure, but not identically, and they detect the difference between transitions that generate the interruptions and decide to increment or decrement the numeric value of the position counter. Because the interruptions structure is a very efficient one the stationary transitions should not be considered in the microcontroller program.

In addition, because the forbidden transitions appear only at out speed of the transducer and because anyway it cannot be taken any decision these forbidden transition will not be included in the microcontroller program. It is just considered that the maximum speed of the transducer will not be crossed.

The output of the incremental transducer decoding automat is a numeric value directly proportional to the displacement of the transducer either from the start of the system or from the last reset. This displacement is the desired parameter of the experimental measurement presented in this paper. This numeric value should be, periodically, transmitted to the computer and used to represent its evolution in time. This periodically communication was done using the clock of the microcontroller and USART communication module. The clock determines a hardware interruption, associated to a specific subroutine that initiates the periodically transmission of the desired data to the computer. The properly transmission is done by the USART module.

### 2.3. The program of data processing and graphical representation

The program of data processing and graphical representation was done in Delphi language. The program has two modules: the acquisition module, which receives the date from the microcontroller; the processing module, which computes these data and graphically represents them.

The acquisition module is connected to the COM1 serial port and waits the data. The microcontroller continuously transmits the data. These are numeric values on 16 bits (2 bytes), covering an enough range (approximately, between $-2^{15}$ and $2^{15}$ steps or between -32768 and 32767 steps that represent and interval between -694 mm and 693 mm, equivalent to 1.4 m). To ensure a correct processing, data are received and memorized in pairs of 2 bytes.

The processing module has one window interface with a minimum number of buttons and may graphically represent the numeric values obtained from the microcontroller by the acquisition module. The time and space diagram of the transducer may be redraw for each new value transmitted by the microcontroller if there are known the transducer resolution and the transmission time period. These two parameters (resolution and transmission time period) define the drawing of the grid used to determine the exact position at a specific moment. Figure 4 represents a graphical representation for transducer time response.

The diagrams from figure 4 were obtained by manually moving the transducer close to a mechanical limit stop. Thus, it can be observed the transducer time response. This diagram offers information regard: position of the mechanical limit stop, amplitude of transducer stroke and the transducer velocity during a specific period of time. In figure 4 the distance between the grid red lines (the darkest ones) was set to 10 mm and the distance between the yellow grid lines (the lightest ones) is 1 mm.

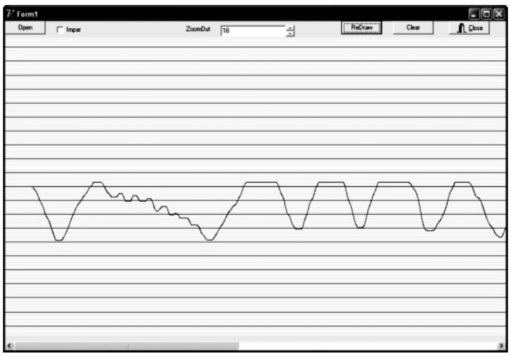


Figure 4. General overview of transducer stroke

## 3. Results, conclusion and future work

During the experiments the data acquisition system for movement characteristics proved to be a very performing one. The maximum velocity depends on the algorithm implemented to the microcontroller, on the microcontroller clock frequency and on the transducer accuracy. In the case presented in this paper the velocity is approximately 1,000,000 steps/second and considering that a step represents a 50 part of a millimetre results a velocity of 20 m/s. For a given system, which has a transducer with an accuracy of *n* times greater than one presented in this paper, the measurement maximum velocity will be *n* times smaller than that corresponding to the transducer presented here. Thus, for a transducer with 1µm accuracy the maximum velocity will be 1m/s.

In the case of a rotational system if there is used a transducer with 1min. accuracy ($2.90888 \cdot 10^{-4}$ rad or 1/60 grad) it will be obtained a maximum angular velocity of 290.888 rad/s or 145.444 complete rotations per second or 8626.646 rot/min.

The system presented in this paper has many practical applications starting with experimental representation of some characteristics for different mechanical systems and continuing with dynamic equilibrate systems, determination of optimum ignition angle for fuel burning engine, controllers for different types of systems (control the interstice of the electro-erosion machine-tools, control the orientation of astronomic telescope or of the radiolocation antenna).

The author of the present paper developed, based on the system presented in this paper, a control system for a DC motor. This control system is based on a PID controller, software implemented in microcontroller's memory. The system allows the control of angular position of the motor (the position accuracy being influenced by the incremental transducer step) or the control of angular velocity of the motor. The intern encoding of position was done on 32 bits, allowing a displacement between -2147483648 and 2147483647 steps. Thus, for a positioning accuracy of 1 micrometer results a maximum displacement between -2.147 and 2.147 km.

As conclusion, the system developed may be used in any application that analyse the dynamic characteristics of a system and as future work, the author want to adapt the control system to an actuation that uses pneumatic muscle controlled with flow control electro-valve [5].

**References**

1. Chivu, C.I.: *Static model and simulation of a pneumatic artificial muscle*. RECENT Journal, vol. 8, No. 3a, October 2007, p.239-243, ISSN 1582-0246, Brasov, Romania
2. Wojtecki, R.G.: *Air logic control for automated systems*. CRC Press Publishing House, ISBN 0-8493-2057-7, Florida USA, 2004
3. Bishop, R.: *The mechatronics handbook*. CRC Press Publishing House, ISBN 0-8493-0066-5, Denver, USA, 2002
4. Levine, W.S.: *The control handbook*. CRC Press Publishing House, ISBN 81-7224-785-0, Florida USA, 2000
5. Shell, R.L., Hall, E.L.: *Handbook of industrial automation*. Marcel Dekker Inc. Publishing House, ISBN 0-247-0373-1, Cincinnati, USA, 2000