An adaptive-method for velocity estimation using time-to-digital converter

Khadgi Mitesh[#], Koul Majid[*], Manivannan M[†]

[#]*Department of Electrical Engineering,* [†]*Department of Applied Mechanics, Indian Institute of Technology Madras*
Chennai-36, Tamil Nadu, India
[#]miteshkhadgi.iitm@gmail.com, [†]mani@iitm.ac.in

[*]*Department of Mechanical Engineering, Indian Institute of Technology Delhi*
New Delhi-16, India
[*]majid_me@student.iitd.ac.in

***Abstract*—Efficient velocity estimation plays an important role in the stability of haptic interfaces. In this work, a new digital circuit is realized to reduce the noise level in low velocity estimation. The proposed adaptive-method (A-method) is based on the concept of measuring the time-interval between two or more incoming quadrature pulses from the optical encoder to the order of picoseconds (ps). A time-to-digital converter (TDC) is used initially to implement the conventional velocity estimation techniques like frequency-count method (M-method) and period-count method (T-method). Later, the two methods are implemented simultaneously using the TDC for the adaptive estimation. The range of TDC used is 0-110 nanoseconds with 420-425 ps resolution. A carry-chain was implemented in order to remove non-linearity and increase precision. Experimental results demonstrate the superiority of the proposed A-method over the conventional T- and M-methods.**

I. Introduction

A haptic interface is a special robotic device that delivers controlled force to the user in response to the voluntary movement of the device, while interacting with a virtual environment. Typical virtual environments are modelled as spring damper systems in mechanical parallel. Force-feedback to the user is exerted during the interaction of virtual position of the robot end-effector with the virtual wall. The controller generates the required feedback-force F according to

$$F = k \triangle x + Bv \tag{1}$$

where, k denotes the virtual wall stiffness, $\triangle$x is the depth inside virtual wall, B represents the virtual wall damping coefficient and v is the velocity of the robot end-effector. It has been well recognized in [1-3] that velocity estimation plays an important role in determining the stability range of haptic interfaces. Typical haptic applications, such as the rendering of virtual environments with damping, require velocity information. Damping effect is crucial to the extent a haptic device is able to emulate virtual environments with high-stiffness [3]. A haptic device needs to display a broad range of damping coefficient in order to preserve stability when in a contact with a stiff wall. Unfortunately, the ability of using large damping coefficients becomes a limiting factor affected by the velocity measurement noise [1-3]. In [6, 13], for the haptic interface control, one of the reasons for unintended vibrations to the user has been realized as the noise effect at low velocity. Haptic applications mostly involve dynamic interactions ranging from very low velocity to high velocity, therefore accurate determination of both low and high velocity is crucial for high fidelity haptic rendering. Commonly used velocity estimation techniques include M-method for high and T-method for low velocity range applications and M/T method a combination of both. M-method involves counting the number of encoder pulses in a fixed-time interval namely the sampling period whereas T-method involves counting the number of sampling periods between two consecutive encoder pulses [8]. Though the combined method appears to be suitable for haptic applications, the accuracy is not sufficient. In this paper we propose an adaptive technique called A-method to improve the accuracy of the velocity estimation at all velocity ranges using time-to-digital converter (TDC).

II. Velocity estimation techniques

In this section we discuss the existing and proposed methods for velocity estimation and their hardware realization on FPGA.

A. M-method

In case of M-method, the number of pulses (m) from the encoder during a fixed sampling period (T) are counted (Fig. 1). The angular velocity is estimated as

$$v = 2\pi m/PT \tag{2}$$

where, P = encoder pulses per revolution. The velocity resolution is given by

$$Q = 2\pi/PT \tag{3}$$

Q decreases at low velocities and at higher sampling periods. As shown in Fig. 1, the required time interval to be measured is T + T_1 - T_2, which is done using a 50 MHz clock input. It incorporates an inaccuracy of T_1 - T_2, with measurable time interval limited to T.

978-1-4577-1740-6/11/$26.00 ©2011 IEEE

A simple finite state machine with four pre-defined binary states for two input signals are used to implement the method with a high frequency clock input for high-precision counter and a low frequency clock input for update rate calculation. High frequency clock divides the time-interval under consideration into many small time-intervals and the low frequency clock samples out the required data after certain fixed amount of time. This forms the reason behind calling M-method also as *fixed-time-interval-method*.

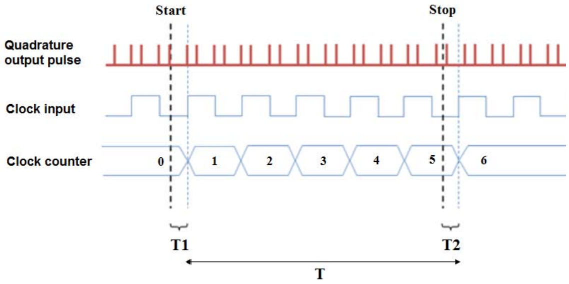

Fig. 1. Start and Stop signals with respect to the next positive edge of clock (M-method)

B. T-method

In case of T-method, the time interval between two consecutive pulses in a fixed-position interval is measured. The angular velocity is estimated as

$$v = 2\pi/mPT \tag{4}$$

where, m = number of sampling periods between two consecutive pulses. T-method is implemented using a counter using clock input with resolution of the order of nanoseconds which is further replaced with a TDC circuit with resolution of the order of picoseconds.

The picoseconds resolution is achieved using a tapped delay-line [8, 11] circuit using 64 D-Flip Flops, 64 full-adders and one 6-bit register, which is further extended to 256 D-Flip Flops and 256 full-adders to increase the range of the time-interval to be measured. 1 D-Flip Flop and 1 full-adder attached together constitutes a delay cell, when connected together in sequence forms a big-chain of 64/256 delay cells. To implement this chain, a specific kind of design embedded in FPGA is used known as *carry-chain*.

A *carry-chain* is an application-specific logic chain used to instantiate a particular connecting path between two points which gives fixed delay every time a signal goes through the path. The range of the fine-resolution time counter used in new T-method is from 0 to 110 ns, as [7, 12] mentioned high-accuracy, wide-speed range, and rapid response as important parameters for velocity estimation. Here, the time-interval under consideration is T_1 or T_2, since T_1 and T_2 are equal in quadrature pulse due to a constant phase shift of 90° between encoder inputs A and B.

C. M/T-method

In general, the above T- and M-methods are suitable for low and high velocity estimation respectively because of the inherent nature of their estimation. So, for a better dynamic measurement of velocity, M/T method is used. Velocity estimation in M/T-method is done using an M- or T-method enable bit which stores the information about the method to be used at an instant. This bit enables switching of formulations at a particular predefined threshold. M-method takes control if the velocity is above a threshold region otherwise T-method is used for the estimation of velocity.

D. A-method

The proposed A-method has been introduced to compute the time-interval between the two consecutive quadrature pulses more accurately than the M/T-method. In fact, in the A-method the concept is not merely switching between M- or T-methods as in M/T-method at a predefined threshold rather both formulations are used simultaneously at each instant. Although the A-method is a combination of M- and T-method it differs from M/T-method with respect to the fine-count resolution required for T-method in the order of picoseconds as shown in Fig. 4.

III. Hardware implementation for TDC

The circuit implemented for TDC consists of a sequence of a full-adder and a D-Flip Flop together (1-delay cell) used to implement a carry-chain. *START* and *STOP* are the two input signals generated from the sequence of encoder pulses. In practice, the time-interval between two consecutive encoder quadrature pulses change according to the velocity. For low velocity, the pulse-width becomes larger than the pulse-width at high-velocity. Fig. 1, 2 shows the quadrature output Q created by using the encoder pulses A and B which is simply a XOR logic of A and B input signal denoted as

$$Q = A(XOR)B = \bar{A}B + A\bar{B} \tag{5}$$

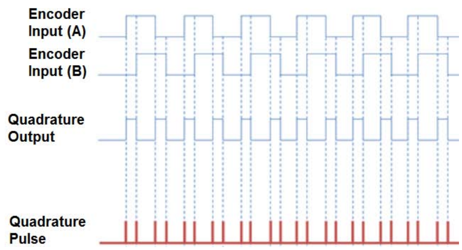

Fig. 2. Quadrature output signal created for TDC input A and B

In Fig. 3, *IN* signal is provided as an in-built logic '0'. *START* is connected directly to a full-adder and *STOP* is connected as a clock input to all the D-Flip Flops for sampling the output at every positive-edge of the *STOP* signal. The main aim of using full-adder as a buffer is to use the delay-chain for the signal propagation as *high* or *low* which can be used in the velocity estimation methods discussed in the following section.

In *A-method*, for low velocity range with quadrature output pulse-width much wider than for high-velocity range with

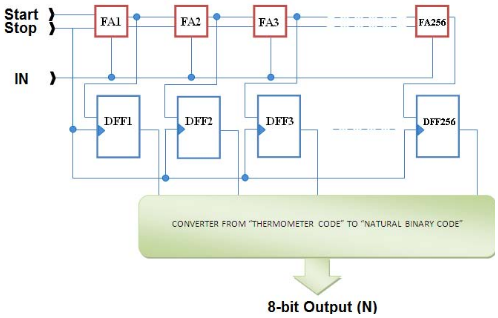

Fig. 3. Circuit design of Time-to-Digital converter

smaller quadrature output pulse-width, T_1 time-interval is measured by initiating higher resolution T-method till the next positive-edge of the clock followed by starting the clock counter till the next positive-edge of *STOP* signal. The value of T_1, (t - T_2) and T_2 is stored in different registers and using (6), the required time-interval between *START* and *STOP* signals is calculated.

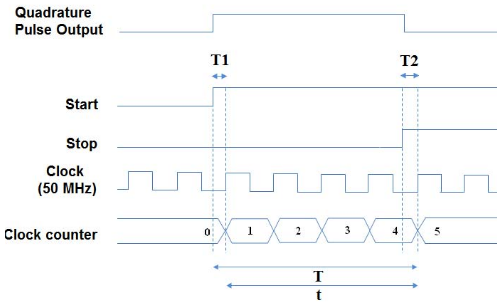

Fig. 4. A-method timing diagram

$$T = t + T_1R_1 - T_2R_2 \tag{6}$$

where, R_1 = first T-method resolution, R_2 = second T-method resolution, T_1 = start signal to next clock positive edge, T_2 = stop signal to next clock positive edge, T = time interval between start and stop signal, t = reference time interval.

IV. Experimental Results

Experiments for velocity estimation using M-, T-, M/T- and the proposed A-method were implemented on Altera Cyclone II FPGA with position input from a 1-DOF haptic device containing a 1000 cpr (counts per revolution) optical encoder. The block diagram and the experimental setup are shown in Fig. 5 and 6 respectively. The instantaneous velocity graphs were obtained using NI-DAQ(National Instruments Data-Acquisition) system. Figure 7 shows the instantaneous velocity estimation signal using M-method at 1 ms and 0.1 ms update rate using a sampling period of 20 ns (50 MHz on-board clock). M-method gives better results at 1 ms with noise-free velocity signal (i.e., unwanted oscillations in the velocity estimation curve) while at 0.1 ms the noisy behaviour in

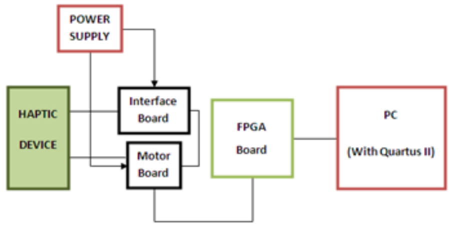

Fig. 5. Block diagram of the experimental setup

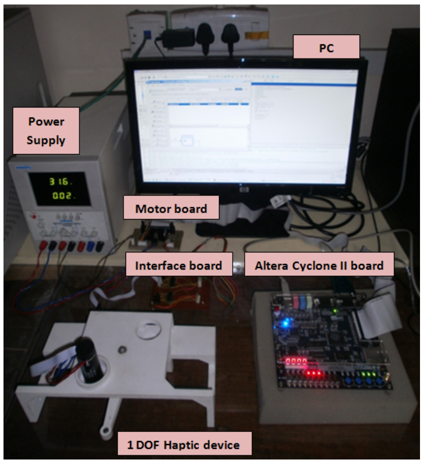

Fig. 6. Experimental Setup

velocity estimation is aggravated. Results using T-method are shown in Fig. 8. T-method is suitable for lower update rate of the order of milliseconds (ms). For update rates below 0.1 ms, the velocity signal showed oscillations / spikes when compared to the update rate of around 5-20 ms which produced good results. Experimental results using M/T-method are shown in Fig. 9, where switching between M-method and T-method occurs depending on the number of pulse counts called as threshold value (50, in our experiment). At the instant, T-method reaches the number of counts above 50, control is taken over by the M-method. In Fig. 9, the encircled part with red color indicates the time-interval into consideration where switching from T-method to M-method takes place at positions indicated by the black-colored circles. Overall the switching happens 3 times in the velocity graph. The readings are taken at 1 ms update rate. The first graph in Fig. 9 shows the velocity magnitude using M/T-method and the second graph shows the velocity magnitude when M-method is active. The third graph shows the magnified portion of the first graph for low velocities using T-method, which is slightly visible in the first graph as dark small lines on the X-axis. A simple if-else condition was used for switching between the M-method and T-method. Experimental results for the proposed A-method velocity estimation signal are shown in Fig. 10. A-method was found to be suitable for both low and high update rates. When the update rate is increased, similar to the T-method with nanoseconds accuracy, A-method produces oscillations

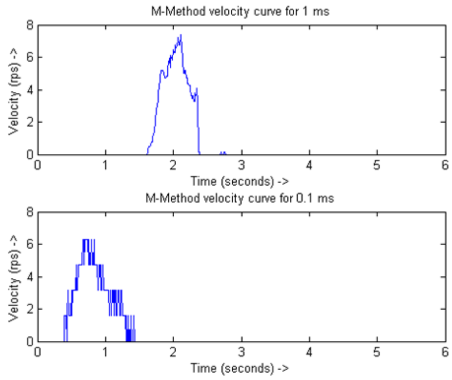

Fig. 7. Velocity estimation at 1 and 0.1 ms update rate (M-method)

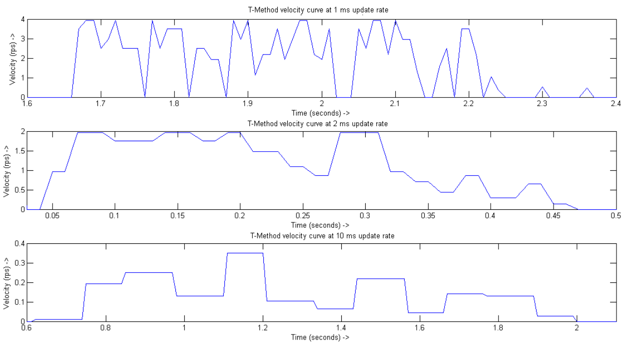

Fig. 8. Velocity estimation at 1, 2 and 10 ms update rate (T-method)

in the velocity estimation signal. As shown in Fig. 10, the velocity signal at 1 ms update rate shows better results with A-method when compared to the T-method at similar update rate.

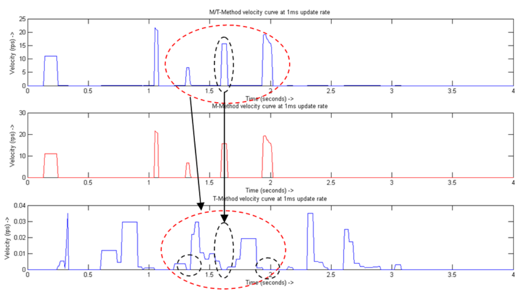

Fig. 9. Velocity estimation at 1 ms (M/T-method)

V. Summary

With update rate lesser than 0.1 ms, spikes started appearing in the M-method. Since T-method uses picoseconds resolution, a range of around 100 ns makes it difficult to compare both the M- and T-methods at same update rates. M-method measurements become accurate at lower update rate where as T-method measurements become accurate at higher update rate. When the update rate is increased, the number of pulses decreases by significant amount and even reaches zero, whereas when the update rate is decreased, more number of pulses start coming in the range which increases the accuracy by averaging the whole time-interval with the total number of pulses detected. At low-velocity range, A-method performed better than the T-method. Since A-method is a combination of M-method (with ns accuracy) and T-method (with ps accuracy), A-method has a better count resolution than T-method alone. A-method worked well for 1 ms update rate with lesser spikes compared to the M-, T- and M/T-method at similar update rates.

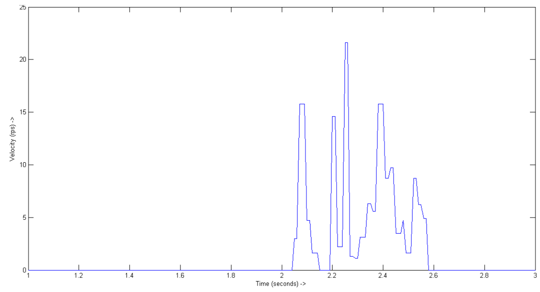

Fig. 10. Velocity estimation at 1 ms update rate (A-method)

References

[1] B. B. Han, J. H Ryu, "An Injecting Method of Physical Damping to Haptic Interfaces Based on FPGA," Proceedings of the International Conference on Control Automation and Systems (ICCAS), Seoul, Korea, 2008.

[2] J.E. Colgate, J.M. Brown, "Factors affecting the Z-width of a haptic display," in 1994 Proc. IEEE Int. Conf. Robot. Automat., pp. 3205-3210, 1994.

[3] W.-H. Zhu, T. Lamarche, "Damping enhancement of haptic devices by using velocities from accelerometers and encoders," Proc. of 48th IEEE Conf. on Decision and Control pp. 7515-7520, Shanghai, China, December 2009.

[4] J. Kalisz, R. Szplet, R. Pelka, A. Pooniecki, "Single-chip interpolating time counter with 200-ps resolution and 43-s range," IEEE Trans. Instrumentation and Measurement, Vol. 46, No. 4, August 1997, pp. 851-85.

[5] J. Wu, Z. Shi, "The 10-ps Wave Union TDC: Improving FPGA TDC Resolution beyond Its Cell Delay," IEEE, Nuclear Science Symposium Conference Record, 2008.

[6] B. B. Han, J. H. Ryu, K. Jung, "FPGA Based Time Domain Passivity Observer and Passivity Controller," International Conf. on Advanced Intelligent Mechatronics, Singapore, 2009.

[7] T. Tsuji, M. Mizuochi, H. Nishi and K. Ohnishi, "A Velocity Measurement Method for Acceleration Control," IEEE Trans. Industrial Electronics, 2005.

[8] J. Kalisz, "Review of methods for time interval measurements with picosecond resolution,"" Metrologia (41), 2004, pp. 17-32.

[9] R. Pelka, J. Kalisz, R. Szplet, "Nonlinearity correction of the integrated time-to-digital converter with direct coding," IEEE Trans. Instrumentation and Measurement, Vol. 46, No. 2, April 1997, pp. 449-452.

[10] J. Wu, Z. Shi, Y. Wang, "Firmware-only implementation of Time-to-Digital Converter (TDC) in Field-Programmable Gate Array (FPGA)," Nuclear Science Syposium, Portland, 2003.

[11] M. D. Fries, J. J. Williams, "High-precision TDC in an FPGA using a 192 MHz quadrature clock," Nuclear Science Symposium Conference Record, IEEE, Vol. 1, 10-16 Nov. 2002, pp. 580-584.

[12] T. Tsuji, M. Mizuochi, K. Ohnishi, "Technical issues on Velocity Measurement for Motion control," Power Electronics and Motion Control Conference, 2006, pp. 326-331.

[13] Y. S. Kim, B. Hannaford, "Some Practical Issues in Time Domain Passivity Control of Haptic Interfaces," Proceedings of Intelligent Robots and Systems, Vol. 3, 2001, pp. 1744-1750.

[14] T. Tsuji, T. Hashimoto, H. Kobayashi, M. Mizuochi and K. Ohnishi, "A Wide-Range Velocity Measurement Method for Motion Control," IEEE Trans. Industrial Electronics, Vol. 56, No. 2, 2009.