Reducing Timing Errors of Polling Mechanism Using Event Timpstamps for Safety-Critical Real-Time Control Systems

Heejin Kim, Sangsoo Park
Dept. of Computer Science and Engineering
Ewha Womans University
Seoul, South Korea
Email: heejin.kim@ewhain.net, sangsoo.park@ewha.ac.kr

Sehun Jeong, Sungdeok Cha
Dept. of Computer Science and Engineering
Korea University
Seoul, South Korea
Email: gifaranga@korea.ac.kr, scha@korea.ac.kr

***Abstract*—Real-time control systems must handle events from external environment in timely manner to achieve their objectives. To detect such events and then to obtain input values, interrupt mechanism has been often used in such systems. While commonly used in practice, predictability of system behavior based on interrupts is difficult to analyze because system behavior might be different depending on the order of interrupts detected by the system. Despite potential weakness of polling approach such as timing errors, recent advances on hardware provide features that can effectively deal with such challenges. For example, as registers maintain timestamps of events, system can provide enhanced predictability even when all the sensor events are periodically polled. In this paper, we describe a case study in which existing interrupt-based design of a real-time software controlling an artificial heart has been modified to the polling-based version. Empirical experiments revealed that revised design is as efficient, when measured in terms of system's external output, as the old design.**

***Keywords*-real-time system, safety-critical systems, software design, predictability;**

I. Introduction

Real-time control systems typically have a set of tasks for accomplishing system objectives while each task should be completed within its deadline. External environment dynamically generates input values, usually by sensors, to the real-time control system, and then the control software embedded in the system computes output values to actuators for accomplishing its objectives. To obtain such input values in timely manner, such real-time control systems often use the interrupt mechanism.

Interrupts are preferred when developing real-time control systems, because they use hardware support to reduce both the latency and overhead of event detection compared to polling mechanism [1] . Predictability indicates amount of analysis efforts to calculate the next state of the target system at any given time or software state [2] and it is especially important when designing real-time software for safety-critical systems, as exhaustive testing is impractical and testing alone can never establish sufficient safety assurance [3]. While commonly used in practice, predictability of system behavior based on interrupts is difficult to analyze because system behavior might be different depending on the order of interrupts detected by the system.

Polling mechanism, on the other hand, obtains the input values periodically, while its principles inherently introduce time gaps between accual event occurrences and polling. Fig. 1 depicts an example of interarrival times of interupts from a magnetic sensor in a real target system, Hybrid Ventricular Assist Device (H-VAD) artifical heart system [4], [5], that is used as a case study in the following sections in this paper. If all the external events are handled by interrupt mechanism, then the control software would be able to obtain the input values immediately within interrupt delays which is often very short. There are time gaps, however, until the control software obtains the input values for polling mechanism. The time gaps or timing errors decrease both monitoring performance and responsiveness. A real-time control system only with polling mechanims can be designed to execute a set of tasks using a predefined time grid without the interventions of interrupts, which will provide valuable time information for developers when they try to analyze and to predict a given system behavior [6]. As shown in Fig. 1, the time gaps can be reduced to some extent by polling more frequently, but, it is inevitable to increase the system resource utilization at the same time that may affect timing properties of some of tasks in the system [7]. Polling period influences both task schedulability and efficiency of the whole real-time system. System resource utilization may go beyond the schedulability threshold, as the polling rate goes too high. Too low polling rate, as a contrary, decreases system response performance and, at worst, may lose some events.

Despite potential weakness of polling mechanism (i.e. timing errors), recent advances on hardware provide features that can deal with such challenges. As special registers maintain timestamps of sensor events, we can effectively minimize the timing errors, potential design vulnerability caused by time gaps between the occurrence of sensor events and event polling, with built-in event capture hardware capabilities, which are common features in modern off-the-shelf embedded processors from major vendors. Major

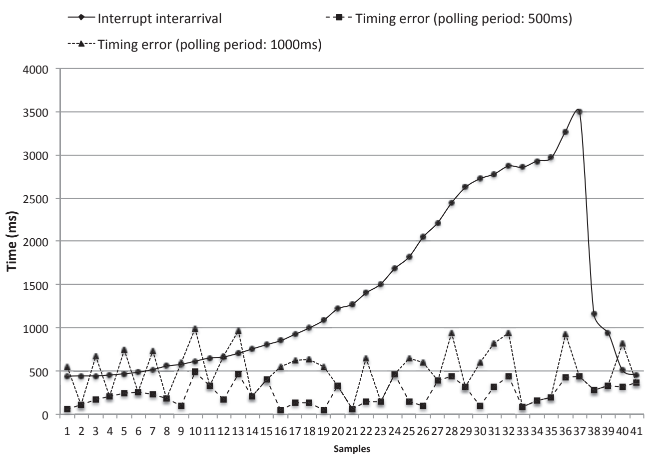

Figure 1. An example of real-time jitters of an interrupt (CAPINT4) in H-VAD artificial heart system.

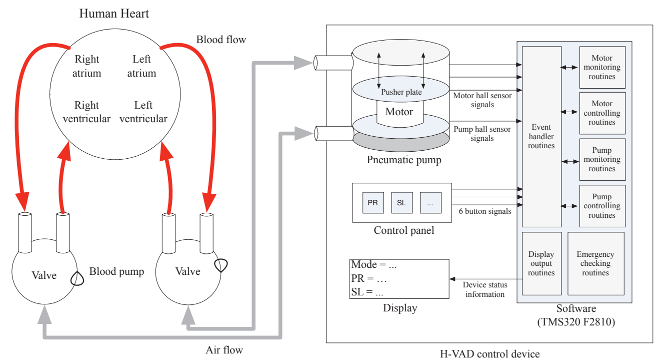

Figure 2. H-VAD artificial heart system.

disadvantage of polling mechanisms is timing error between polling time and actual event arrival time. If the system relies on timestamp of events for monitoring activities, this disadvantage degrades quality of monitored values. But, fortunately, timestamp can be precisely restored due to the built-in features in modern embedded processors that automatically store timestamp in registers. We confirmed that simple calculation could reproduce marginally equal timestamps with original's one

This paper introduces a way to establish safer basement of safety-critical real-time control systems. First, in order to calculate proper density of time grid for polling mechanism, we obtained real-time constraints via profiling interrupts. Second, we effectively removed real-time jitters, with built-in event-timestamping hardware capabilities. Further, we describe a concrete case study in which existing interrupt-based design of a real-time software controlling an artificial heart has been modified to the polling-based design. Benefits of such architectural refactoring include rigorous schedulability guarantee, ease of software quality assurance due to enhanced system predictability. Empirical experiments revealed that revised design is as efficient, when measured in terms of system's external output, as the old design.

II. Case Study: Architectural Refactoring of Real-Time Control Software in Artifical Heart

Artificial hearts, which is a typical safety-critical real-time control systens, become the only practical solution for the patients with terminal heart disease because the lack of heart donors [8], [9], [10] and clinical issues such as suppressing the immune system for implant living organs are still unchallenged problems. Modern real-time motor controllers [11], [12] and embedded software provide benefits to both the patients and developers. Small and power-efficient controllers contributed in developing portable artificial heart systems, which can enhance the patients' quality of life. Developers, also, flexibly customize the controller for objectives, thanks to the controllers' support for sensor handling and high-level programming languages. However, these benefits came along with software quality assurance problems for clinical use. Pumping speed control logic, for example, takes target pumping speed and current pumping speed as inputs and computes a new speed for the next motor movement. Anomaly in embedded software may lead to medical emergencies or even death in the worst case.

H-VAD is a portable artificial heart developed by Korea Artificial Organ Center (KAOC). H-VAD system has successfully operated for more than 180 days in animal testing, and it satisfied the US FDA (Food and Drug Administration) regulations for long-term experiments. As shown in Fig. 2, this device consists of one or two blood pumps and a H-VAD control device, which is similar to an A4 paper in size, and weighs about 2kg. Due to its size and weight, it provides better mobility to the patients comparing to other portable artificial hearts. The patients can adjust the device according to his or her status by pressing 6 buttons, which allows the controlling 2 parameters H-VAD software consists of about 9,000 lines of C language including about 3,800 lines of specific code to KAOC H-VAD running on the micro controller based on TMS320 F2810 embedded processor.

There are 7 interrupt service routines and can be classified by interrupt sources:

- Timer interrupt (T3PINT): This is the only periodic task executed at the interval of 1 millisecond. It determines the motor's next direction and velocity using the current position, current velocity and reference velocity, which is calculated by using the two control parameters. It also monitors other tasks for confirming normal operations of the pump. In addition, the task periodically polls 6 button events, which are not designated as interrupt.
- Pump center hall sensor interrupt (CAPINT1): This is a sporadic interrupt, which is activated when the

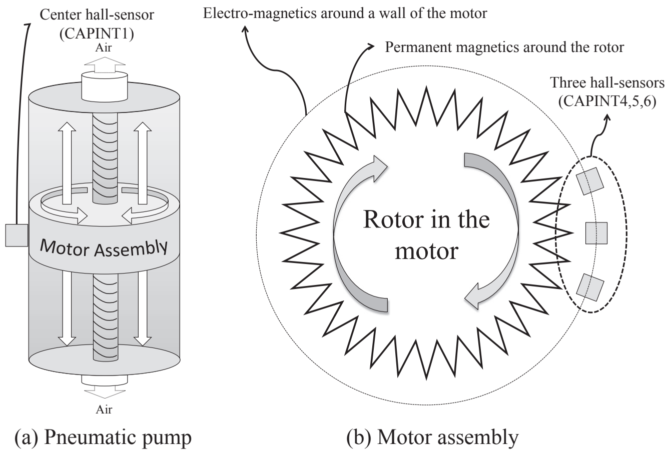

Figure 3. Interrupt sources in H-VAD system.

motor passes through the pump center hall sensor as in Fig. 3(a). This task compares the current center position and the absolute center position of the sensor determined at initial phase, to monitor range of the motor's motion.

- Motor hall sensor interrupt (CAPINT4,5,6): This task is a sporadic interrupt, which is triggered when the motor rotates a fixed angle as in Fig. 3(b). It updates the value of velocity and position. There are three motor hall sensors.
- Button interrupt (CAPINT2,3): This is an aperiodic interrupts, which is triggered when a button is pressed. It updates new values of the control parameters. There are two interrupts multiplexed to 6 buttons.

III. Profiling Timing Properties of Interrupts and a Periodic Task

Refactoring to polling-based software requires an additional event polling task that periodically checks external events if occurred. As summarized in Table I, the augmented timing profiler obtains timing attributes of all tasks in original control software. Based on the collected timing profiles, proper polling period is selected to serve all tasks without skipping. When the motor is set to the highest speed, in the most demanding condition, motor hall sensors trigger interrupts at the rate of about 1.3 milliseconds. In that configuration, the pump would take about 400 milliseconds to complete one round trip of back-and-forth movement. Based on repetitive experiments in various motor settings, we derived information on the execution time of each task as shown in the E column. Due to timer's hardware characteristics, we are unable to measure time intervals shorter than 0.01 millisecond as shown in the P column. Furthermore, software's execution paths were relatively simple and repetitive, observed execution time is a close approximation of each task's worst case execution time (WCET) value.

Table I
Measured timing attributes of H-VAD software.

Name	**Type**	P_{ob} **(msec)**	E_{ob} **(msec)**	**Event source**
CAPINT1	Sporadic	400	< 0.01	Center hall-sensor
CAPINT2	Aperiodic	N/A	< 0.01	Button
CAPINT3	Aperiodic	N/A	< 0.01	Button
T3PINT	Periodic	1	0.23	Internal timer
CAPINT4	Sporadic	1.3	< 0.01	Motor hall-sensor
CAPINT5	Sporadic	1.3	< 0.01	Motor hall-sensor
CAPINT6	Sporadic	1.3	< 0.01	Motor hall-sensor

P: Observed minimum inter-arrival time
E: Observed maximum execution time

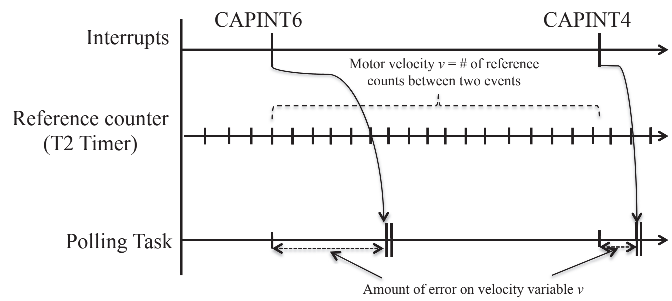

Figure 4. Timing errors in polling mechanism.

IV. Reducing Timing Errors Using Event-Timestamps in Polling Mechanism

The H-VAD control software updates current motor velocity with number of reference counts (T2 timer) between previous two successive motor hall-sensor events. To refactor current interrupt-based software into polling-based one, all of the interrupt handlers were masked and one periodic task whose period is one millisecond handled the external events. As shown in Fig. 4, same motor hall-sensor events, however, are acknowledged at next period of polling task if the events are handled by polling mechanism.

If all events are assumed to have occurred when polling took place, timing gap would introduce subtle deviation from intended motor control patterns shown in the figure. We have to "remember" the precise time when various sensor events had occurred. Such information is not necessary to determine the order the priority among event handers. They are static and never change. However, motor control logic in the H-VAD design uses timestamps of motor hall sensor events to derive the current motor speed and compute the adjustment to be made to the actuator. Special-purpose registers, built in the *TMS320 F2810* motor controller (e.g., *CAP1FLAG*, etc.), were used to preserve execution semantics of the H-VAD design. The hardware event handler stores the timer value in dedicated stack when an expected signal is recognized by sensor or button. In this paper, we polled this counter information from the stack to recover T2 timer information.

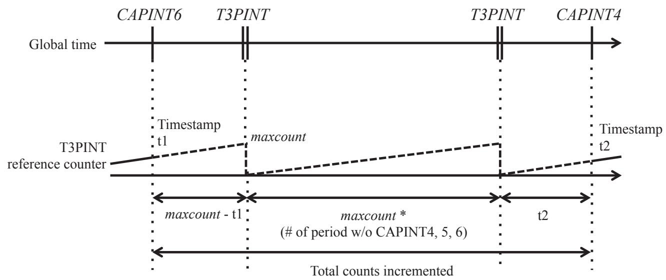

Figure 5. Using event-timestamps to recover reference counts.

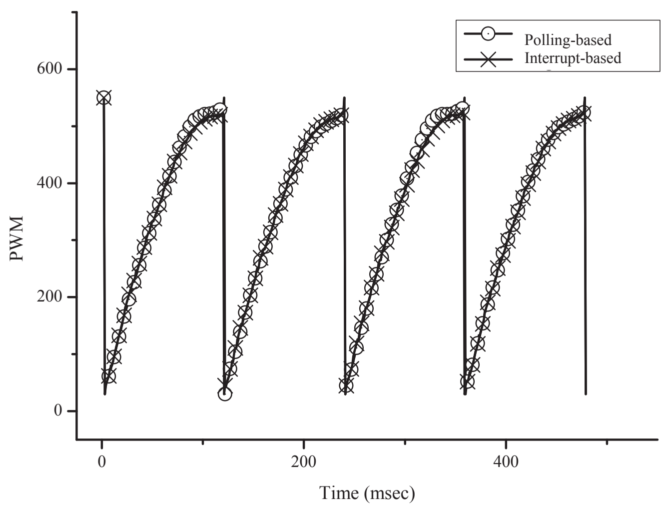

Figure 6. Experimental Results.

Fig. 5 visualizes how the event-timestamps can be used to recover reference counts to calculate precise motor speed even in polling mechanism.

V. Experimental Results

The performace of proposed approach was measured using output variable comparison. H-VAD software periodically outputs PWM (Pulse Width Modulation) [13] value to command speed of the motor in the blood pump. To generalize the comparison, we conducted 3 types of experiments, which have different pumping rate: a) default (pumping rate=50), b) fast (pumping rate=120) and c) slow (pumping rate=30). Each experiment was repeated 5 times and we confirmed expected results from all experiments that showed almost same PWM values in both 2 system architectures. Fig. 6 is the representative overall result when the pumping rate was default value. Besides the PWM values, we checked other important func- tionalities related with the operations (e.g., changing control parameters by pushing buttons) and they were the same as well.

VI. Conclusion

Polling mechanism has known to offer several advantages over interrupt mechanism in safety critical real-time control systems. It comes with real-time guarantee backed up by formal proof, enhanced predictability in system behavior, etc. In this paper, we reported a software design refactoring study in which interrupt-based design of software was converted to polling based design. Modern embedded processors generally offer features (e.g., special-purpose registers to track timestamps) which enable precise and fine control of a system. Experiment results demonstrated that proposed method has advantages in many ways. As a future work, we plan to work on quantitative metric that can measure predictability and maintainability of real-time design approaches.

References

[1] I. Lee, J. Y.-T. Leung, and S. H. Son, *Handbook of Real-Time and Embedded Systems*. CRC Press, 2008.

[2] D. Frund, "Toward a formal definition of timing predectability," October 2009, talk at Workshop on Reconciling Performance with Predictability.

[3] G. Candea, "Predictable software-a shortcut to dependable computing?" *CoRR*, 2004.

[4] G. Jeong, C. Hwang, K. Nam, C. Ahn, J. Lee, J. Choi, and H. Son, "Development of a closed air loop electropenumatic cctuator for driving a penumatic blood pump," *Artificial Organs*, vol. 33, pp. 657–662, 2009.

[5] J. Lee, B. Kim, J. Choi, and C. Ahn, "Optimal pressure regulation of the penumatic ventricular assist device with bellows-type driver," *Artificial Organs*, vol. 33, pp. 627–633, 2009.

[6] M. Short, "Development guidelines for dependable real-time embedded systems," in *Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on*, 31 2008-april 4 2008, pp. 1032 –1039.

[7] H. KOPETZ, "Should responsive systems be event-triggered or time-triggered?" *IEICE Transactions on Information and Systems*, vol. 76, no. D, pp. 1325–1332, 1993.

[8] *Heart disease and stroke statistics 2010 update at-a-glance*, American Heart Association.

[9] *The 2008 annual report of the OPTN and SRTR: heart transplantation*, US Department of Health and Human Services.

[10] *Organ procurement and transplantation network national data*, US Department of Health and Human Services.

[11] "Texas instruments, inc." [Online]. Available: www.ti.com

[12] "Analog devices, inc." [Online]. Available: http://www.analog.com/en/index.html

[13] S. S.K and S. M.G., "Microcontroller based pwm strategies for the real-time control and condition monitoring of ac drives," in *IEEE International Symposium on Industrial Electronics*, June 1998, pp. 219–224.