Scheduling of Accuracy-Constrained Real-Time Systems in Dynamic Environments

Mitra Nasri, *Student Member, IEEE*, Mehdi Kargahi, *Member, IEEE*, and Morteza Mohaqeqi, *Student Member, IEEE*

***Abstract*—Many real-time embedded systems are sensitive to both the accuracy and timeliness of job results. In this letter, two sources of inaccuracy are considered for such systems: 1) input data noise (IDN) due to the environmental transient noises, and 2) age of data (AD) related to the time of capturing data, which may depend on the length of time between capturing and using the input data. Thus, in the presence of one or more jobs in the system, some tradeoffs are needed among capturing data with an appropriate IDN when the environment is less noisy, reducing AD, and respecting the timeliness of jobs. Our emphasis in the current study is to model firm real-time jobs having some thresholds for the inaccuracy and handle the aforementioned tradeoff by the system scheduler. An online accuracy-aware real-time scheduler is also proposed and evaluated.**

***Index Terms*—Accuracy-constrained scheduling, age of data (AD), input data noise (IDN), Kalman filter, real-time embedded system.**

I. Introduction

REAL-TIME embedded systems are often in direct interaction with dynamic and harsh environments. In such time-sensitive and likely mission-critical systems [1], rather than temporal correctness, the accuracy of results is very important in the fulfillment of the mission.

Environmental noises, like the electromagnetic interferences (EMI) [2], can variously affect the input data, and thus, the accuracy of results in embedded systems. Some of these systems are equipped with noise compensation techniques like Kalman filters [1], [3], [4], some are not. In both cases, accuracy of the results will be improved if the system tries to capture data when the environment is less noisy. Noise characteristics can be identified via hardware- or software-based techniques. For example, a hardware technique via a radiometer has been introduced in [5] for detecting correlation and total power of different noises in a receiver device. Also, according to the formulation proposed in [6], one may calculate the signal-to-noise ratio (SNR) using device temperature.

There is another factor, referred to as age of data (AD) in this letter, which affects the accuracy of results. AD is defined as the possible delays of capturing data with respect to when expected, or between capturing and processing data. For example, in multi-sensor real-time systems, delays occurring because of scheduler decisions and context switches would affect the accuracy of the data-fusion estimation process [3], [4]. Current solutions for this problem, employed inside the applications, are compensation techniques based on the estimation of the amount of delay [3], [4]. However, these reactive techniques are not appropriate enough as the application has limited information about the delays. Besides, these techniques can typically tolerate only small amounts of delay [4], while in multi-task systems, when a high priority job acquires the processor, it might enforce delays equal to its execution-time to the existing low priority jobs, possibly causing considerable deficiencies in their accuracy. Thus, the role of the scheduler becomes more important in such systems. In this letter, we propose proactive scheduler-level techniques to control the delays and show that the job accuracy can be moderated if the scheduler is aware of the AD costs.

Previously, there have been attempts to apply scheduling algorithms to control either input data noise (IDN) [7] or AD [8] in real-time systems. In [7], job execution value is specified by an instant value function (IVF) per job, so as the proposed offline preemptive scheduler can schedule jobs at their best (i.e., less noisy) moments. However, in this model, the value function is fixed on job arrival and cannot reflect the effects of dynamic noises of the environment. Besides, this model is not sensitive to AD. On the other hand, AD is studied in [8], where each job in a distributed firm real-time system has an accuracy constraint. In the study, job accuracy is defined as a function of the age of input data stored in a directory service. However, the study considers neither IDN nor the delays happening in the middle of job execution, namely between capturing and processing data. There are also other studies on real-time scheduling trying to reduce job execution time by avoiding context-switch overheads [9], [10], which are not concerned about accuracy and IDN, may indirectly reduce AD.

In summary, this letter introduces a new firm real-time job model with an emphasis on job accuracy. The accuracy is affected by both the IDN and AD, and each job has a threshold for the amount of inaccuracy it can tolerate. If this threshold is violated, the job will be missed (similar to [8]). In fact, both accuracy and timing violations of a job have the same adverse effect on the system or on the success of the mission (see the example inspired by [1] in Section II). An online accuracy-aware real-time scheduling algorithm is also proposed for reducing the miss ratio. The effectiveness of the algorithm is examined through simulation experiments.

The remainder of this letter is organized as follows. In Section II, we present an example to formulate job inaccuracy

Manuscript received January 24, 2012; accepted April 06, 2012. Date of publication April 18, 2012; date of current version September 14, 2012. This manuscript was recommended for publication by S. Ramesh.

M. Nasri and M. Mohaqeqi are with the School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 14399-57131, Iran (e-mail: mitra.nasri@ut.ac.ir; m.mohaqeqi@ut.ac.ir).

M. Kargahi is with the School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 14399-57131, Iran. He is also with the School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran 19395-5746, Iran (e-mail: kargahi@ut.ac.ir).

Digital Object Identifier 10.1109/LES.2012.2195294

as a function of IDN and AD. System and accuracy models are described in Section III. An accuracy-aware real-time scheduling algorithm is presented in Section IV, followed by the experimental results in Section V. Finally the letter is concluded in Section VI.

II. Input Data Noise and Age of Data

In this section, we present a motivating example and formulate job inaccuracy as a function of IDN and AD using the Kalman filter formulations [11] in a multi-sensor system.

In a real-time guidance navigation and control (GNC) system of unmanned aerial vehicles (UAV) [1], there are different types of sensors like low cost inertial measurement unit (IMU) and global positioning system (GPS). They are usually applied together to reduce the error of vehicle position and velocity estimations using a Kalman filter. In the GNC system, position estimation is the first step in a sequence of tasks that finally lead to the actuator commands. In the case of low accuracy, such a command may be ignored. Therefore, the accuracy of the estimation process plays a vital role in the effectiveness of the guidance loop. As long as the estimation process is fed by GPS and IMU data, variation of the IDN and delays between capturing these two data have significant impacts on the accuracy of the GNC outputs.

In general, Kalman filters can be used to estimate the state of a process that is governed by a stochastic difference equation [11] and is subjected to process and measurement noises. The state equation is $x(k+1) = f_k(x(k), u(k)) + w(k)$, where $x(k) \in \mathbb{R}^N$ is a vector of size N, representing the state of the process at time t_k, and $u(k)$ and $w(k)$, respectively, are the system control command and the process noise (with covariance Q_k) at the same time [4] (k is taken as a short notation for t_k). Also, the system has a number of sensors (indexed by $i \in \mathbb{N}$) giving discrete measurements as $y_i(k) = h_i(x(k)) + v_i(k)$, where $h_i(x(k))$ is the observation model and $v_i(k)$ is the measurement noise with covariance matrix $R_i(k)$. A Kalman filter tries to minimize the mean square error of the estimated state with respect to the actual process state through computing the Kalman gain such that P_k, the error covariance, is minimized. See[11] for more details.

Also, a multisensor system is prone to measurement delays caused by context switches and scheduling decisions [3], [4]. These delays cannot be neglected since they might have considerable adverse effects on the results. In fact, this concern is rooted at the AD. To tolerate AD, Nilsson *et al.* [4] have extended the Taylor's series to merge the measurement time delay of sensor i, i.e., $\delta t_i(k)$, into an extended Kalman filter formulation to estimate and compensate AD in the application-level. In fact, the delay will affect $y_i(k)$ in the following way:

$$y_i(k) = h_i(x(k + \delta t_i(k))) + v_i(k), \quad i = 1, 2, \ldots. \tag{1}$$

Substituting (1) into the formulation of the updated estimated value of $x(k)$, i.e., $\hat{x}^+(k)$, and in turn, using the resulting $\hat{x}^+(k)$ into the formulation of P_k^-, one can attain the relation between IDN, AD, and the inaccuracy (trace of matrix P_k^-, namely trace of the predicted value of error covariance). Now, the inaccuracy formula used by every Kalman filter can be rewritten as a function of IDN and AD (see [4] and [11])

$$P_k^- = \hat{F}_k P_{k-1} \hat{F}_k^T + Q_k \tag{2}$$

where $\hat{F}_k$ is the Jocobians of $f_k(\hat{x}^+(k), u(k))$.

In practice, IDN can be obtained through methods such as the ones introduced in [2] and [12]. These methods calculate $R_i(k)$, which is $R_i(k) = VAR(v_i(k))$. Also, the delay identifying AD can be extracted from the scheduler as it knows the I/O instants of the jobs. As mentioned before, we aim to moderate the effects of IDN and AD on inaccuracy in a proactive manner in the level of system scheduler.

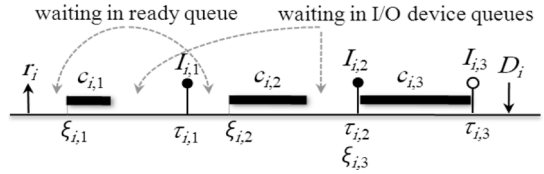

Fig. 1. Job with three segments.

III. System and Accuracy Models

We consider a system with a set of ρ I/O devices $I = \{I_1, \ldots, I_\rho\}$. It is assumed that the system has another hardware device (like the simple one proposed in [5] for EMI measurement) specialized for noise detection. This hardware also implements a method to convert the noise level into the effective variance of noise on the input data (such as in[2]). We call this value IDN, which is represented by $\mathcal{J}_{I_i}(t)$ for I/O device I_i at time t. A noise-free device I_i has $\mathcal{J}_{I_i}(t) = 0$.

The system consists of firm real-time jobs, $J_i, i \in \mathbb{N}$. Job J_i is identified as $J_i : (r_i, D_i, \alpha_i, \gamma_i(\cdot), S_i)$, where r_i and D_i are the job release-time and relative deadline, respectively. $\alpha_i \in [0, 1]$ is the threshold for the maximum tolerable inaccuracy of that job. $\gamma_i(\cdot)$ is the job inaccuracy function, and will be discussed in detail shortly. If $\gamma_i(\cdot) > \alpha_i$, an accuracy violation is occurred. Either on timing violation or on accuracy violation, the job is considered missed and is thrown away. Each job has a sequence of h_i segments defined by $S_i : \langle s_{i,j} \rangle 1 \leq_{j \leq h_i}$. Each segment is specified by two parameters $s_{i,j} : (c_{i,j}, I_{i,j})$, where $c_{i,j}$ is the segment computation time (e.g., to store the sensor data into registers or for some application-specific computations), and $I_{i,j} \in I$ defines the I/O type of the segment (like the I/O operations in the UAV system).

In this system, when a job is released, its first segment is entered into the ready queue which is managed by the system scheduler. When segment $s_{i,j}$ acquires the main processor at time $\xi_{i,j}$, it is executed non-preemptively up to its end, where an I/O operation is triggered. As a result, an I/O request $I_{i,j} \in I$ is sent to the respective I/O device queue, and the scheduler is activated. Except for the final output request, the scheduler is supposed to decide whether to perform the I/O at the moment $(\tau_{i,j})$ or defer it. It is assumed that I/O operations can be performed in an infinitesimal amount of time. When the I/O is done, the next segment of the job (i.e., $s_{i,j+1}$) enters the ready queue (See Fig. 1).

As denoted above, job inaccuracy is computed based on a job-specific inaccuracy function $\gamma_i(\cdot)$. In general, $\gamma_i(\cdot)$ is a function of the job description, moments of I/O operations (denoted by $\tau_{i,j}$ for $I_{i,j}$), IDN of those I/O operations ($\mathcal{J}_{I_{i,j}}(\tau_{i,j})$), and start time of the execution of segments ($\xi_{i,j}$)

$$\gamma_i(\langle\tau_{i,j}\rangle, \langle\mathcal{J}_{I_{i,j}}(\tau_{i,j})\rangle, \langle\xi_{i,j}\rangle) \text{ for } j = 1, \ldots, h_i. \quad (3)$$

$\gamma_i(\cdot)$ is an *a priori* known application-specific function which is determined at the job release time. However, without loss of generality, we consider the inaccuracy of job J_i as a function of its segment inaccuracies, i.e., $\gamma_i(\langle\gamma_{s_{i,j}}\rangle, j = 1, \ldots, h_i$, where $\gamma_{s_{i,j}}$ is the inaccuracy of segment $s_{i,j}$. In turn, $\gamma_{s_{i,j}}$ is defined as a function of IDN and AD of its preceding I/O operations and preceding segment inaccuracies. For example, for a simple job of Kalman filter with one input operation, $\gamma_{s_{i,j}}$ can be computed as the trace of matrix P_k^- using (1) and (2), where $\delta t_{I_{i,1}} = \tau_{i,1} - r_i$ and $v_{I_{i,1}} = \mathcal{J}_{I_{i,1}}(\tau_{i,1})$.

IV. An Accuracy-Aware Scheduling Algorithm

In this section, we introduce accuracy-aware EDF (A-EDF) scheduling algorithm with the objective of decreasing miss ratio, namely the sum of timing and accuracy violation ratios. It is a work conserving algorithm based on EDF [13], and is activated when a job enters an I/O device queue. A-EDF has two steps. First, it selects a candidate list (CL) of jobs from those waiting in I/O device queues considering their IDNs, and second, it picks a job to be executed on the processor among the jobs in the ready queue and CL. In the latter step, it also updates CL, considering possible amounts of AD for the waiting jobs. This updated list constructs the I/O schedule which is used to update the ready queue.

In the first step, A-EDF selects a number of jobs which satisfy the following condition from the I/O device queues and put them in CL: the current value of IDN of each selected job should not be such that an accuracy violation happens for it. To find whether the accuracy is going to be violated or not, A-EDF optimistically calculates the inaccuracy of the job, using the known inaccuracies of the previously executed segments, IDN of the current segment, and perfect accuracy for the future segments of the job. A perfect segment accuracy can be computed considering no delay (in the ready queue and I/O device queues) and no noise ($\text{IDN} = 0$) for the segment.

In the second step, A-EDF selects job J_i with the earliest deadline among the jobs in the ready queue and CL, if it will not be missed if executed. The job will run its segment $s_{i,j}$, namely occupies the processor for $c_{i,j}$ units of time when this step of scheduling is completed, and thus, it will impose an additional AD equal to $c_{i,j}$ to all jobs in CL. Accordingly, A-EDF removes those jobs in CL which, if they face this amount of additional AD, will suffer an accuracy violation. Afterwards, the remaining jobs in CL constitute the final I/O schedule, according to which the ready queue is updated. Finally, the selected job in the second step (J_i), is scheduled on the processor. If the scheduler cannot find a candidate with the above characteristics, it behaves exactly like EDF.

The intuitive idea behind A-EDF is that it has a separate scheduling phase for I/O of the jobs waiting in the I/O device queues, which may change the order of jobs with respect to EDF. Since an arbitrary pattern for noise is assumed, finding an optimal I/O schedule might not be possible, and thus, A-EDF is designed to work in a greedy manner.

Assuming n jobs in the system, $H = \max_{1 \le i \le n} h_i$ and time complexity $O(H)$ for computing inaccuracy function of a job, A-EDF runs in $O(nH + n \log n)$. The reason is that it traverses CL two times each in $O(nH)$, sorts the jobs in the ready queue and CL according to their deadlines, and selects the appropriate job. If $H \ll \log n$, as it is likely in systems with limited number of I/O devices like the example presented in Section II, then the algorithm runs in $O(n \log n)$.

V. Simulation Results

In this section, we compare A-EDF with the deferred preemption (DP) [10], EDF [13], and gEDF [9] algorithms, customized for the job model of this letter. DP is an optimistic extension of EDF which tries to reduce the number of preemptions. It defers the execution of a job with the earliest deadline if it has enough slack, to let the running job continue its execution. Unlike DP and EDF, gEDF is an algorithm specialized for non-preemptive systems. It creates groups of jobs with roughly the same absolute deadline, and selects the job with the shortest remaining execution time. The simulation framework has been developed in the DEVS-suite discrete event simulator [14].

In the experiments, a system with four types of I/O devices $I = \{I_1, \ldots, I_4\}$ has been considered. Three of them are influenced by Gaussian noise with parameters $\mathcal{N}(0.01, 0.01)$, $\mathcal{N}(0.05, 0.05)$, and $\mathcal{N}(0.10, 0.10)$, which is similar to an example in [11]. The fourth I/O device is noise-free with $\text{IDN} = 0$. We have examined the system using 10 randomly generated periodic tasks and, for each experiment setup, 1000 runs with different task sets. The processor utilization ranges from 0.1 to 1. For each job, h_i is selected randomly from the set $\{1, \ldots, 7\}$ and segments have uniformly distributed execution times in a way that the system utilization is guaranteed. The segments are associated with I/O devices $I_{i,j} \in I$ with uniform chances. Each job has one segment representing a Kalman filter (randomly selected between the implementations of examples of [4] and [11]) and $h_i - 1$ segments with simple function $\gamma_{s_{i,j}} = 1 - (1 - \beta_{i,j}(\tau_{i,j} - \tau_{i,j-1} - c_{i,j}))\big(1 - \mathcal{J}_{I_{i,j}}(\tau_{i,j-1})\big)$, where $\beta_{i,j}$ is the segment AD cost and follows $U(0, 2)$. We have scaled $y_i(k)$ of [4] by 10^{-5} to make it comparable with the IDN in our I/O device setup. In fact, we have implemented the Kalman filters based on (2), to perform the simulations with more realistic inaccuracy functions.

The inaccuracy of a job is considered to be $\gamma_i = 1/h_i \sum_{j=1}^{hi} \gamma_{si,j}$, and each job has an inaccuracy threshold of $\alpha_i = 0.15$. When a job is executed, the inaccuracy of its Kalman filter segment is normalized to the maximum and minimum possible values for its state variable $x(\cdot)$. For the other segments, this value is normalized with respect to the maximum possible inaccuracy of the segment obtained by the worst IDN for I/O of the segment and the largest possible delay it may suffer (i.e., the slack time of the job).

Fig. 2 shows the miss ratio (MR) of the aforementioned scheduling algorithms for different processor utilizations. Fig. 3 depicts the performance of EDF, gEDF, DP, and A-EDF, normalized with respect to MR of EDF. For each algorithm in this figure, the average values of MR, accuracy violation ratio

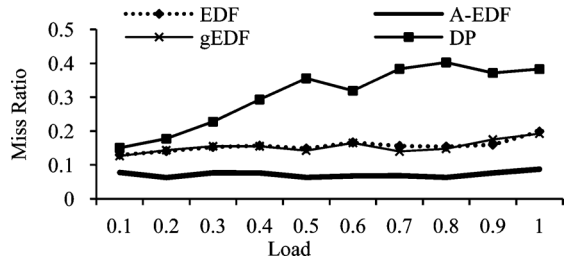

Fig. 2. Miss ratio for DP, EDF, gEDF, and A-EDF algorithms.

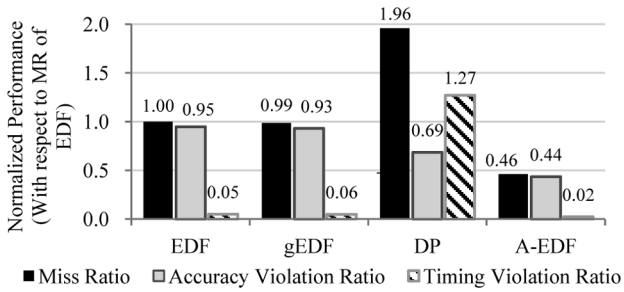

Fig. 3. Normalized performanes with respect to the miss ratio of EDF.

(AVR), and timing violation ratio (TVR) for loads 0.1 to 1.0 with steps of 0.1 have been illustrated.

As can be seen, A-EDF efficiently decreases AVR and TVR compared to the other algorithms. This reveals the importance of scheduling I/O operations and managing the amount of AD for the jobs. DP is an optimistic algorithm which may generate overload situations leading to high TVRs, although it has lower AVR than EDF (see Fig. 3) because of the chance it gives to the running jobs to continue their executions to suffer less AD. According to Figs. 2 and 3, there are no considerable differences between gEDF and EDF in MR. The reason is that the scheduler is activated at the end of each segment (rather than each job) and it allows the newly arrived jobs, with earlier deadlines, to be executed in the middle of the previously executing job. This can be regarded as some kind of preemption, making gEDF and EDF work similarly. It is also worth noting that TVR of all the EDF-based algorithms is negligible with respect to MR. Beside the experiment setup, one major reason is that, especially when the system in not overloaded, EDF is a proper algorithm for reducing TVR.

VI. Conclusion

In this letter, we have demonstrated the possibility of applying scheduling algorithms for improving the accuracy of jobs in real-time embedded systems working in dynamic environments. Here, the accuracy is regarded as a function of IDN and AD. An online accuracy-aware real-time scheduling algorithm is also proposed and its appropriateness is shown through simulations. This algorithm efficiently reduces AVR with almost no effect on TVR with respect to EDF. Further studies could focus on a scheduling algorithm to make some tradeoffs between TVR and AVR through guaranteeing some possible upper bounds for each.

References

[1] J. H. Kim, S. Sukkarieh, and S. Wishart, "Real-time navigation, guidance, and control of a UAV using low-cost sensors," *Field and Service Robotics: Springer Tracts in Advanced Robotics*, vol. 24, pp. 299–309, 2006.

[2] M. F. Abdel-Hafez, "The autocovariance least-squares technique for GPS measurement noise estimation," *IEEE Trans. Vehic. Technol.*, vol. 59, no. 2, pp. 574–588, 2010.

[3] M. Choi, J. Choi, J. Park, and W. K. Chung, "State estimation with delayed measurements considering uncertainty of time delay," in *Proc. Int. Conf. Robot. Automat.*, 2009, pp. 3987–3992.

[4] J. O. Nilsson, I. Skog, and P. Handel, "Joint state and measurement time-delay estimation of nonlinear state space systems," in *Proc. Int. Conf. Inform. Sci. Signal Process. Appl.*, 2010, pp. 324–328.

[5] I. Corbella, F. Torres, A. Camps, N. Duffo, M. Vall-llossera, K. Rautiainen, M. Martin-Neira, and A. Colliander, "Analysis of correlation and total power radiometer front-ends using noise waves," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 11, pp. 2452–2459, 2005.

[6] R. B. Langley, "GPS receiver system noise," *GPS World*, vol. 8, no. 6, pp. 40–45, 1997.

[7] L. Farzinvash and M. Kargahi, "A scheduling algorithm for execution-instant sensitive real-time systems," in *Proc. IEEE Int. Conf. Embedded Real-Time Comput. Syst. Appl.*, 2009, pp. 511–518.

[8] Q. Han and N. Venkatasubramanian, "Timeliness-accuracy balanced collection of dynamic context data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 2, pp. 158–171, 2007.

[9] W. Li, K. Kavi, and R. Akl, "A non-preemptive scheduling algorithm for soft real-time systems," *Comput. Elect. Eng.*, vol. 33, no. 1, pp. 12–29, 2007.

[10] R. J. Bril, J. J. Lukkein, and W. F. J. Verhaegh, "Worst-case response time analysis of real-time tasks under fixed-priority scheduling with deferred preemption revisited," in *Proc. Euromicro. Conf. Real-Time Syst.*, 2008, pp. 269–279.

[11] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Dept. of Computer Sci., University North Carolina, Chapel Hill, NC, Tech. Rep., 2000.

[12] M. T. Ozludemir, "The stochastic modeling of GPS observables," *Turk. J. Eng. Environ. Sci.*, vol. 28, pp. 223–231, 2004.

[13] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. ACM*, vol. 20, no. 1, pp. 46–61, 1973.

[14] Discrete-Event System Simulator (DEVS-suite), Arizona Center of Integrative Modeling and Simulation, Arizona State University [Online]. Available: http://acims.asu.edu/software/devs-suite.