Fidelity-Aware Utilization Control for Cyber-Physical Surveillance Systems

Jinzhu Chen¹, Rui Tan² Guoliang Xing¹; Xiaorui Wang³; Xing Fu³
¹Michigan State University, USA; ²City University of Hong Kong; ³University of Tennessee, USA

Abstract—Recent years have seen the growing deployments of Cyber-Physical Systems (CPSs) in many mission-critical applications such as security, civil infrastructure, and transportation. These applications often impose stringent requirements on system sensing fidelity and timeliness. However, existing approaches treat these two concerns in isolation and hence are not suitable for CPSs where system fidelity and timeliness are dependent of each other due to the tight integration of computational and physical resources. In this paper, we propose a holistic approach called Fidelity-Aware Utilization Controller (FAUC) for wireless cyber-physical surveillance (WCS) systems that combine low-end sensors with cameras for large-scale ad hoc surveillance in unplanned environments. By integrating data fusion with feedback control, FAUC can enforce a CPU utilization upper bound to ensure the system’s real-time schedulability although CPU workloads vary significantly at runtime due to stochastic detection results. At the same time, FAUC optimizes system fidelity and adjusts the control objective of CPU utilization adaptively in the presence of variations of target/noise characteristics. We have implemented FAUC on a small-scale WCS testbed consisting of TelosB/Iris motes and cameras. Our extensive experiments on light and acoustic target detection show that FAUC can achieve robust fidelity and real-time guarantees in dynamic environments.

I. INTRODUCTION

Cyber-physical system (CPS) is a new class of embedded systems that tightly integrate computational and physical resources. Recent years have seen the growing deployments of CPSs in many mission-critical applications such as security, civil infrastructure, and transportation. These applications often impose stringent performance requirements including sensing fidelity and timeliness. In this work, we define fidelity as a system’s capability of reaching correct conclusions even when the sensing results from the dynamic physical environment are noisy. In addition to fidelity, timeliness is another fundamental requirement as many computational tasks in a CPS must complete within tight deadlines in order to avoid undesirable or even catastrophic consequences.

In this work, we investigate the problem of addressing both fidelity and timeliness requirements of wireless cyber-physical surveillance (WCS) systems. A typical WCS system consists of battery-powered cameras, sensors, and embedded computers that communicate through wireless networks. Without the reliance on wired power/communication infrastructure, WCS systems can be rapidly deployed in an ad hoc manner for large-scale surveillance in unplanned environments. This is a key advantage for many critical domains such as security, transportation, and natural/physical hazard monitoring. In 2008, a number of wirelessly connected cameras were deployed for real-time and high-fidelity surveillance over 26-mile course of the Boston Marathon which attracted over 20,000 runners and more than one million spectators. In other scenarios like border security, WCS systems need to provide surveillance and intruder detection during an extended period of time up to several years. Due to the tight budget on power resources and network bandwidth, WCS systems often operate in an on-demand fashion where low-end (e.g., acoustic/infra-red/magnetic) sensors serve as “sentinels” that wake up high-quality but power consuming sensors (e.g., pan-tilt-zoom cameras) once a possible target is detected. High-quality sensing results (e.g., images) are then transmitted to an embedded computer for high-fidelity object detection and recognition.

Both fidelity and timeliness are essential requirements of the WCS systems described above. As an example, users may require any target of interest to be detected “at high fidelity (both missing and false alarm rates lower than 1%) and in real time (delay within five seconds)”. However, a key challenge is that the timeliness and fidelity of a WCS system are tightly dependent of each other. First, the performance of low-end sensors is extremely sensitive to dynamics in the physical environment. It is shown in [10] that individual dual-axis magnetometers on Mica2 motes [6] can exhibit up to 60% false alarm and missing rates. As low-end sensors trigger image capture and processing, their poor fidelity can significantly affect the workload and real-time performance of the system. For instance, the false alarms from low-end sensors not only lead to energy waste of cameras but also generate extra computation workload for image processing. On the other hand, reducing CPU workload and camera activity unnecessarily may lead to the increased target missing rate.

In this paper, we argue that the fidelity and real-time concerns of WCS systems must be jointly addressed due to the tight integration of system computational and physical components. Numerous real-time scheduling algorithms have been proposed to achieve real-time guarantees for computing systems. However, many of them require detailed knowledge of CPU workload while WCS systems are subject to stochastic workload due to the impact of physical dynamics. Several recent approaches [20] [15] [27] can handle variable system workload. However, they are incognizant of system fidelity requirements. On the other hand, although sensor calibration [24] and signal processing [23] techniques are available to improve the fidelity of a sensing system, they do not account for the impact on system timeliness. For instance, minimizing target missing
rate often leads to a high false alarm rate [23], which in turn poses undesirable CPU workload for a WCS system as discussed earlier.

In this work, we propose a novel approach to holistically addressing the fidelity and timeliness requirements of WCS systems. Our approach integrates multi-sensor data fusion [23] with feedback control to achieve adaptive fidelity and real-time guarantees for WCS systems operating in dynamic environments. Specifically, we make the following major contributions in this paper.

1) We propose a novel problem formulation for the fidelity-aware utilization control problem where a given upper bound on the CPU utilization is enforced while system detection error rate is minimized. Utilization control can ensure the system’s real-time schedulability (e.g., based on the Liu and Layland bound for RMS [13]) despite significant uncertainties in system workloads. Moreover, it can also enhance system survivability by providing overload protection against workload fluctuation. Our formulation is based on two rigorous performance models that characterize the fusion-based detection performance and the expected CPU utilization induced by processing stochastic detection results.

2) We develop Fidelity-Aware Utilization Controller (FAUC) that adaptively adjusts the data fusion threshold to bound the CPU utilization according to user requirement. At the same time, FAUC minimizes the system detection error rate while ensuring real-time schedulability. We also rigorously establish the conditions for FAUC’s convergence and stability.

3) We have implemented FAUC on a small-scale WCS testbed consisting of TelosB motes, Iris motes, and cameras. Our extensive experiments on light and acoustic target detection show that FAUC can achieve robust fidelity and utilization control in presence of significant physical dynamics.

The rest of the paper is organized as follows. Section II reviews related work. Section III presents the background on sensing and data fusion models. Section IV describes our problem and provides an overview of our approach. Section V and VI model system performance and present the design of FAUC, respectively. Section VII presents the experimental results and Section VIII concludes the paper.

II. RELATED WORK

Fidelity assurance is a fundamental issue in sensing and cyber-physical systems. Early solutions are focused on sensor calibration or noise mitigation through signal processing. In [18], each chemical sensor is carefully calibrated in controlled environments to obtain the mapping from its reading to the true value. Recent studies aim to optimize the overall system sensing fidelity. In [8], the biases of light sensors are estimated by solving the equations that correlate their measurements. Similarly, in [24], the parameters of ranging sensors are estimated based on pair-wise range measurements. The above approaches calibrate sensors according to known ground truth inputs and hence work in an open-loop fashion. In our recent work [22], we develop a feedback-based calibration algorithm that maintains system sensing fidelity in the presence of environmental dynamics.

Data fusion [23] is an effective signal processing technique that improves the sensing fidelity of sensing systems by mitigating the impact of noise. Most previous studies [23] [3] focus on analyzing the optimal fusion strategy of a given sensing system. In our recent work [26] [21], we investigate the impact of data fusion on coverage and timeliness of surveillance sensor networks. However, all the above solutions on sensing fidelity assurance are not concerned with meeting timing constraints. In this paper, we aim to adaptively control the fusion parameters to maintain a desirable CPU utilization bound while maximizing system fidelity in dynamic environments.

Feedback control techniques have shown great promise in providing real-time guarantees for CPSs by adapting to workload variations based on dynamic feedback. For instance, feedback-based CPU utilization control [20] [15] [27] has been demonstrated to be an effective way of meeting the end-to-end deadlines for real-time systems. However, most of these algorithms rely on task rate adaptation and hence cannot handle unpredictable task rate variations that may be caused by low system fidelity. Different from these studies, we aim to jointly address the requirements on system fidelity and CPU utilization of WCS systems.

III. PRELIMINARIES

In this section, we present the preliminaries of our work, which include sensor measurement and data fusion models.

A. Sensor Measurement Model

We assume that sensors measure the energy of received signals for event detection. Let \( s_i \) denote the signal energy received by sensor \( i \), which is affected by several factors and varies for different sensors. First, each sensor may have its hardware bias. Second, the measurement value is stochastic as it inevitably contains environmental noise. Third, the signal path loss between the event and sensor varies with terrain and other environmental factors. Let \( H_0 \) denote the hypothesis that the target is absent and \( H_1 \) denote that the target is present, the measurement of sensor \( i \), denoted by \( y_i \), is given by

\[
\begin{align*}
H_0 : & \quad y_i = n_i \\
H_1 : & \quad y_i = s_i + n_i
\end{align*}
\]

where \( n_i \) is the energy of noise in sensor \( i \)'s measurement. We assume that the noise \( n_i \) at each sensor \( i \) is independent and follows the normal distribution, i.e., \( n_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \), where \( \mu_i \) and \( \sigma_i^2 \) are the mean and variance of \( n_i \), respectively. The sensor measurement model described above has been widely adopted in the literature of event detection [4] [19] and also have been empirically verified [9], [12]. However, many previous studies assume that the parameters of above model, \( s_i, \mu_i, \sigma_i \) are known \textit{a priori}. Unfortunately,
this assumption often does not hold in reality due to the stochastic nature of sensing. In this paper, we assume that these parameters are unknown to the system and design an adaptive control solution, using well-established control theory, to achieve the desired system detection performance based on dynamic feedback.

B. Data Fusion Model

Data fusion [23] has been proposed as an effective signal processing technique to improve the system performance of sensing systems. A system based on data fusion is usually organized into multiple clusters. Each cluster has a cluster head that gathers information from member sensors and makes the system decision regarding the presence of the target. We adopted a simple data fusion model where the system decision is made by comparing the sum of member sensors’ measurements against a threshold $T$, which is referred to as fusion threshold hereafter. This model has been adopted by several previous studies [4] [26] [22]. Suppose there are $N$ sensors in a cluster. The sum of measurements, denoted by $Y$, is given by $Y = \sum_{i=1}^{N} y_i$. Let $H_0$ and $H_1$ represent the detection decisions that the target is absent and present, respectively. Denote $S = \sum_{i=1}^{N} s_i$, $\mu = \sum_{i=1}^{N} \mu_i$ and $\sigma^2 = \sum_{i=1}^{N} \sigma^2_i$. Depending on whether the target is present, the sum of sensor measurements in a cluster can be expressed as:

$$Y|H_0 = \sum_{i=1}^{N} \sim N(\mu, \sigma^2)$$  \hspace{1cm} (1)

$$Y|H_1 = \sum_{i=1}^{N} \sim N(\mu + S, \sigma^2)$$  \hspace{1cm} (2)

We adopt the following decision rule [4] [26] [22]: a target is detected only if the sum of sensor measurements is greater than threshold $T$, i.e., $Y > T$. The detection of a target is inherently stochastic due to the random noises in sensor measurements. The system detection performance is characterized by two metrics, namely, the false alarm rate (denoted by $P_F$) and missing probability (denoted by $P_M$). $P_F$ is the probability of deciding $H_1$ when no target is present, and $P_M$ is the probability of deciding $H_0$ when a target is present. $P_F$ and $P_M$ are formally given as follows.

$$P_F = Q\left(\frac{T - \mu}{\sigma}\right)$$  \hspace{1cm} (3)

$$P_M = Q\left(\frac{T - \mu - S}{\sigma}\right)$$  \hspace{1cm} (4)

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} \exp\left(-\frac{t^2}{2}\right)dt$, is the complementary cumulative probability distribution function of normal distribution.

IV. PROBLEM STATEMENT

In this section, we describe the problem of fidelity-aware utilization control. We first discuss the system model in Section IV-A. We formulate our problem and provide a brief overview of our approach in Section IV-B.

A. System Model

We assume that a wireless cyber-physical surveillance (WCS) system consists of a base station and multiple sensor clusters. Each cluster is composed of low-end and high-quality sensors. The low-end sensors (e.g., acoustic and infrared sensors) usually have a low manufacturing cost and low energy consumption. As a result, their sensing capability is often limited. As we discussed in Section III-B, the system performance can be improved by employing data fusion on the measurements of low-end sensors. The high-quality sensors (e.g., pan-tilt-zoom cameras [25] and active radars [7]) can provide high-accuracy sensing and detection at the price of higher manufacturing cost and energy consumption. In the following, we assume that there is only one high-quality sensor in each cluster. However, our approach can be easily extended to the case of multiple high-quality sensors.

In accordance with the heterogeneous system architecture, we adopt a two-phase target detection process. Initially, the low-end sensors periodically sense the environment while the cameras remain asleep because their power consumption is typically several orders of magnitude higher than low-end sensors [5]. The cluster head fuses data from low-end sensors and makes a decision according to the threshold $T$. If it is a positive decision, the cluster head then activates the camera to capture the image of the surveillance region, and sends the image to the base station. Finally, a target recognition algorithm is executed by the base station to process the images and detect whether a target of interest is present. The key advantage of the two-phase target detection scheme described above is that the system power consumption can be significantly reduced without sacrificing the detection performance. In particular, most false alarms can be filtered out by the data fusion of low-end sensors and hence the high-quality sensors (i.e., cameras) can sleep for most of the time and be switched on only when the probability of target presence is high. As a result, the system can achieve high-fidelity surveillance for extended lifetime in unplanned environments without wired power infrastructure.

B. Approach Overview

Our objective is to achieve satisfactory timeliness and fidelity of WCS systems. We now describe the formulation of our problem and provide a brief overview of our approach.

To guarantee the end-to-end timeliness required by a WCS system, the delay of each stage of the entire process of sensing, communication, and computing must be carefully considered. In previous studies [11], achieving real-time sensor sampling, data fusion, and wireless communications in surveillance systems has been extensively studied. In this paper, we focus on providing the real-time guarantee on target detection in base station that must run computation-intensive tasks to process high-quality sensor data such as images. Specifically, we control the CPU utilization to enforce appropriate schedulable utilization bounds (e.g., the Liu and Layland bound for RMS [13]) despite significant uncertainties in system workloads. In the meantime, uti-
lization control can also enhance system survivability by providing overload protection against workload fluctuation [14]. Our approach can be integrated with previous solutions [11] to ensure the end-to-end timeliness of a WCS system. For instance, the deadline of target detection can be ensured by enforcing sub-deadlines for sensing, communication, and computing separately. We note that the delay of computation is often significant when complex sensing data such images need to be processed.

Several challenges must be addressed to satisfy both timeliness and fidelity requirements simultaneously for WCS systems. First, the timeliness and fidelity performance of a system are highly dependent of each other. For instance, although the false alarms of low-end sensors can be dealt with by turning on the camera more frequently, it inevitably increases CPU workload and impedes system timeliness. On the other hand, reducing CPU workload and camera activity aggressively may lead to an increased target missing rate. Second, system CPU workloads are highly variable due to several factors such as uncertain image processing time and stochastic low-end sensor detection performance. The probability that the camera is activated and an image needs to be processed is highly dependent on the data fusion results, which in turn are affected by time-varying noise and target characteristics in dynamic environments.

In this paper, we propose a control-theoretic solution called Fidelity-Aware Utilization Controller (FAUC) to address these challenges. FAUC employs a feedback controller to enforce the specified upper bound on CPU utilization of base station while minimizing the overall system detection error rate. By taking advantage of the adaptivity of the controller, FAUC allows a WCS system to achieve robust assurance on timeliness and fidelity in dynamic environments. We can formally formulate our problem as follows.

**Problem 1** (Fidelity-Aware Utilization Control). To find a stable and converging control algorithm for the fusion threshold $T$ at the cluster head based on the feedback of the base station, such that the expected CPU utilization $E[u]$ is upper bounded by $u_s$ while the detection error rate $P_e$ is minimized, where $u_s$ is a constant that ensures system’s real-time schedulability.

In the above formulation, we focus on the utilization control of only one cluster. When there exist multiple clusters in the system, their CPU utilization bounds can be determined by schedulability analysis [13] and then separately enforced by multiple FAUC controllers. The detection error rate $P_e$ in Problem 1 is defined as the sum of false alarm and missing rates, which is widely adopted to characterize the performance of detection systems [23]. We choose fusion threshold $T$ as the control input as it impacts both the system detection performance and timeliness. When $T$ is lower, the missing rate is lower while more false alarms may be triggered by noise leading to higher system workload. On the other hand, a higher $T$ reduces both false alarm rate and system workload while a target is more likely to be missed.

Fig. 1 illustrates the architecture of FAUC controller and a wireless cyber-physical surveillance system.

**V. PERFORMANCE MODELING**

In this section, we formally model the performance of WCS systems. The results provide a foundation for the design of FAUC controller in Section VI. We first model the system detection error rate in Section V-A. The impact of communication packet loss is analyzed in Section V-B. Finally, we model system CPU utilization in Section V-C.

**A. Modeling System Detection Performance**

We now formally model the system detection performance. We make the following assumptions. First, the probability that a target is present at any time instance is $P_n$ which is unknown but can be estimated from detection history. Second, the false alarm rate and missing probability of the high-quality sensor, denoted by $P_{FH}$ and $P_{MHI}$, are
known. $P_{FH}$ and $P_{MH}$ can often be measured via offline experiments. Due to the high accuracy of the high-quality sensor, both $P_{FH}$ and $P_{MH}$ are close to zero. We use the system average error rate, denoted by $P_e$, to qualify the system detection performance. This metric has been widely adopted in the literature of sensing systems [23]. According to the above assumptions, $P_e$ can be derived as the weighted sum of system false alarm and missing probabilities under two cases where both low-end sensors and high-quality sensor make wrong decisions, and the high-quality sensor misses the target while low-end sensors make the positive decision, respectively.

$$P_e = (1 - P_a)P_{FL}P_{FH} + P_a[P_{ML} + (1 - P_{ML})P_{MH}]$$  \hspace{1cm} (5)

We now discuss how to achieve the minimal $P_e$. Let $P_{ij}$ denote the probability that the system makes a decision of $j$ while the true hypothesis is $H_i$ ($i, j \in \{0, 1\}$). For instance, $P_{1,0}$ represents the probability that the final decision is zero while the target is present. Note that there only exist two decisions under either hypothesis, so we have $P_{0,0} + P_{1,1} = 1$ and $P_{0,0} + P_{1,1} = 1$. We can express $P_{FL}$ and $P_{ML}$ by $P_{0,1}$ and $P_{1,0}$ in Eqn. (5), respectively.

$$P_e = (1 - P_a)P_{0,1}P_{FH} + P_a[P_{1,0} + (1 - P_{1,0})P_{MH}]$$

$$= (1 - P_a)P_{0,1}P_{FH} + P_a + (1 - P_{1,0})P_a P_{1,1}$$

According to Eqn. (3) and (4), $P_{ij}$ can be expressed as a function of $Q(x)$, which is in turn a function of $T$. To find the condition for the minimum $P_e$, let $\frac{dP_e}{dT} = 0$ and we have,

$$P_{FH}(1 - P_a)\frac{d}{dT}Q\left(\frac{T - u_j}{\sigma}\right) + P_a(1 - P_{MH})\frac{d}{dT}Q\left(\frac{T - u_j}{\sigma}\right) = 0$$ \hspace{1cm} (6)

By solving Eqn. (6), we have the optimal threshold $T_{opt}$ that minimizes $P_e$:

$$T_{opt} = \frac{\delta \sigma^2}{2S} + \mu + \frac{S}{2}.$$ \hspace{1cm} (7)

$$\delta = 2 \ln \left( \frac{1 - P_a}{P_a} \cdot \frac{P_{FH}}{1 - P_{MH}} \right)$$ \hspace{1cm} (8)

**B. Impact of Packet Loss**

Packet loss caused by unreliable wireless communication can lead to deviation of system detection performance of low-end sensors from the theoretical results derived in Section V-A. We propose to address the impact of packet loss by exploiting temporal sampling, where each low-end sensor samples a number of measurements and transmits them to the cluster head to mitigate the impact of packet loss. Suppose sensor $i$ samples $m$ times in a detection process. Let $y_{ij}$ denote the $j^{th}$ noisy measurement of sensor $i$ in a detection, $p_i$ denote the end-to-end packet reception rate (PRR) of the path from sensor $i$ to the cluster head, and $u_{ij} \in \{0, 1\}$ denote the packet delivery state of measurement $y_{ij}$. Hence, $u_{ij}$ is a Bernoulli random variable with success probability of $p_i$. The cluster head fuses all measurement received during a detection process to make a decision, where the fusion statistic $Y$ is given by $Y = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij} \cdot y_{ij}$.

In the absence of target, the mean and variance of $Y$ are given by:

$$\mathbb{E}[Y|H_0] = \sum_{i=1}^{N} \sum_{j=1}^{m} \mathbb{E}[u_{ij} \cdot y_{ij}|H_0]$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{m} \mathbb{E}[u_{ij}] \cdot \mathbb{E}[y_{ij}|H_0] = m \cdot \sum_{i=1}^{N} p_i \cdot \mu_i$$

$$\text{Var}[Y|H_0] = \sum_{i=1}^{N} \sum_{j=1}^{m} \text{Var}[u_{ij} \cdot y_{ij}|H_0]$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{m} \mathbb{E}[u_{ij}^2 \cdot y_{ij}^2|H_0] - (\mathbb{E}[u_{ij} \cdot y_{ij}|H_0])^2$$

$$= m \cdot \sum_{i=1}^{N} \left[ \mathbb{E}[u_{ij}^2] \cdot \mathbb{E}[y_{ij}^2|H_0] - p_i^2 \cdot \mu_i^2 \right]$$

$$= m \cdot \sum_{i=1}^{N} p_i \cdot \sigma_i^2 + p_i^2 \cdot (p_i - p_i^2)$$

Note that $\mathbb{E}[\mu_i^2] = p_i$ and $\mathbb{E}[\sigma_i^2|H_0] = \sigma_i^2 + p_i^2$. Similarly, in the presence of target, the mean and variance of $Y|H_1$ are given by:

$$\mathbb{E}[Y|H_1] = m \cdot \sum_{i=1}^{N} p_i \cdot (s_i + \mu_i)$$ \hspace{1cm} (9)

$$\text{Var}[Y|H_1] = m \cdot \sum_{i=1}^{N} p_i \cdot (\sigma_i^2 + (s_i + \mu_i)^2) - p_i^2 \cdot (s_i + \mu_i)^2$$

$$= m \cdot \sum_{i=1}^{N} p_i \cdot \sigma_i^2 + (s_i + \mu_i)^2 \cdot (p_i - p_i^2)$$ \hspace{1cm} (10)

Note that $\mathbb{E}[\sigma_i^2|H_1] = \sigma_i^2 + (s_i + \mu_i)^2$. As $Y$ is the sum of $N \cdot m$ independent random variables, according to the central limit theorem (CLT), $Y$ follows the normal distribution when $N \cdot m$ is large enough. Although $N$ is often limited in practice, we can increase the number of samplings during a detection to satisfy the condition of CLT. Denote $\mu' = m \cdot \sum_{i=1}^{N} p_i \cdot \mu_i$, $S' = m \cdot \sum_{i=1}^{N} p_i \cdot s_i$, $\sigma_{H_0} = \sqrt{\text{Var}[Y|H_0]}$ and $\sigma_{H_1} = \sqrt{\text{Var}[Y|H_1]}$. The system false alarm rate and missing probability of low-end sensors are given by:

$$P_{FL} = Q\left(\frac{T - \mu'}{\sigma_{H_0}}\right)$$ \hspace{1cm} (11)

$$P_{ML} = Q\left(\frac{T - \mu' - S'}{\sigma_{H_1}}\right)$$ \hspace{1cm} (12)

**C. Modeling CPU Utilization**

To guarantee the real-time schedulability (e.g., by rate-monotonic scheduling [13]), the CPU utilization of each task at the base station shall be maintained at a certain level. In this section, we derive the CPU utilization model. The CPU workload of the base station is mainly generated by processing the images captured by camera. As the camera is activated by stochastic decisions of data fusion, the CPU workload is hence subject to change over time. We define that a control cycle consists of $m$ detections. In each detection, low-end sensors send their measurements to the
cluster head for data fusion. We now derive the expected CPU utilization in \( m \) detections of a control cycle, denoted by \( \mathbb{E}[u] \), by accounting for the workload generated by both correct decisions and false alarms.

We define the following notation subject to a control cycle: 1) \( n_{f1} \) and \( n_{d1} \) are the numbers of false alarms and correct detections made by the cluster head, respectively. Note that they are unknown to the system; 2) \( n_{f2} \) and \( n_{d2} \) are the numbers of positive decisions made by the cluster head but regarded as false alarms and correct detections by the high-quality sensor, respectively. These two numbers can be counted by the base station after processing the images of camera. We have the following relationships between \( n_{f1}, n_{d1}, n_{f2}, \) and \( n_{d2} \):

\[
\begin{align*}
n_{f1} + n_{d1} &= n_{f2} + n_{d2} \quad (13) \\
n_{f2} &= n_{f1}(1 - P_{FH}) + n_{d1}P_{MH} \quad (14) \\
n_{d2} &= n_{f1}P_{FH} + n_{d1}(1 - P_{MH}) \quad (15)
\end{align*}
\]

Eqn. (13) holds because either a correct decision or false alarm from data fusion would trigger a control cycle at the base station. The result of image processing can then again be classified as correct decision or false alarm. In Eqn. (14), \( n_{f1}(1 - P_{FH}) \) represents the number of false alarms that are correctly identified by the high-quality sensor, and \( n_{d1}P_{MH} \) represents the number of correct detections that are incorrectly decided as false alarms. In (15), \( n_{f1}P_{FH} \) represents the number of false alarms that are incorrectly decided as correct detections, and \( n_{d1}(1 - P_{MH}) \) represents the number of detections that are correctly identified. From (14) and (15), the unknown \( n_{f1} \) and \( n_{d1} \) can be estimated as

\[
\frac{n_{f1}}{n_{d1}} = \frac{n_{f2}P_{MH} - n_{d2}P_{MH}}{1 - P_{FH} - P_{MH}},
\]

and

\[
\frac{n_{d1}}{n_{f1}} = \frac{n_{f2}P_{FH} - n_{d2}P_{FH}}{1 - P_{FH} - P_{MH}}.
\]

Therefore, the estimates of \( P_{FL} \) and \( P_{ML} \), denoted by \( \tilde{P}_{FL} \) and \( \tilde{P}_{ML} \), respectively, are given by

\[
\tilde{P}_{FL} = \frac{n_{f1}}{m - m \cdot P_{a}} \quad \text{and} \quad \tilde{P}_{ML} = \frac{m \cdot P_{a} - n_{d1}}{m \cdot P_{a}}.
\]

The high-quality sensor sends the data to the base station for image processing, which consumes the CPU resource. The CPU execution time for image processing is stochastic with the expected value \( e \). The expected CPU utilization of the base station, denoted by \( \mathbb{E}[u] \), is given by

\[
\mathbb{E}[u] = \frac{(n_{f1} + n_{d1}) \cdot e }{T_d} + \frac{(m - n_{f1} - n_{d1}) \cdot e' }{T_d},
\]

where \( e \) represents the expected execution time for processing an image, \( e' \) represents the expected execution time without processing an image, \( u \) represents the CPU utilization and \( T_d \) represents the duration of a control cycle. Note that \( e' \) may equal zero if no processing is required when the data fusion low-end sensor produces a negative result. Replacing the \( n_{f1}, n_{d1} \) in Eqn. (19) with Eqn. (18) and using Eqn. (3) and Eqn. (4) for \( P_{FL} \) and \( P_{ML} \), we can express the expected CPU utilization as a function of \( T \)

\[
\mathbb{E}[u] = \frac{m \cdot e' }{T_d} \left[ (1 - P_{a})P_{FL} + P_{a}(1 - P_{ML}) \right] \quad (20)
\]

\[
= K_1 + K_2 \left[ (1 - P_{a})Q \left( \frac{T - \mu}{\sigma} \right) + P_{a}Q \left( \frac{T - \mu - S}{\sigma} \right) \right] \quad (21)
\]

where \( K_1 = \frac{m \cdot e'}{T_d} \) and \( K_2 = \frac{m \cdot (e - e')}{T_d} \). From this equation, we can see that \( \mathbb{E}[u] \) is a decreasing function of \( T \) because both \( P_{FL} \) and \( 1 - P_{ML} \) are decreasing function of \( T \).

VI. FIDELITY-AWARE CPU CONTROLLER

In this section, we first design the fidelity-aware CPU controller and discuss its stability and convergency in section VI-A and VI-B. We discuss the estimation of plant parameters in section VI-C and discuss the approach to optimizing the detection error rate in section VI-D.

A. CPU Controller Design

The objective of Problem 1 is to ensure \( \mathbb{E}[u] \leq u_s \) while minimizing system detection error rate \( P_e \), where \( \mathbb{E}[u] \) is a function of the fusion threshold \( T \) given by Eqn. (21) and \( u_s \) is a user-specified utilization bound. As the threshold \( T \) is calibrated for every control cycle, Problem 1 is a typical discrete-time control problem, in which \( u_s \) is the reference, \( T \) is control input and \( \mathbb{E}[u] \) is the controlled variable. In the following, we present the design of Fideliity-aware Utilization Feedback Controller (FAUC). We first discuss how FAUC ensures utilization bound, i.e., \( \mathbb{E}[u] \leq u_s \). In Section VI-D, we discuss how system detection error rate \( P_e \) is minimized under the given utilization bound.

Fig. 2. The closed-loop system to control the fusion threshold according to the CPU utilization feedback.

The block diagram of the FAUC feedback control loop is shown in Fig. 2. A challenge to derive the expression of \( G_p(z) \) in the controller is that \( Q(x) \) in Eqn. (21) is a nonlinear function. There are some mathematical approximations of the \( Q(x) \) function, such as [2], however, they typically have a nonlinear form. In order to design the controller with acceptable accuracy, we now derive a linear approximation of \( Q(x) \). By the fundamental theorem of calculus,

\[
\frac{dQ}{dx} = -\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right) \quad (22)
\]

Then the first order Taylor approximation of \( Q(x) \) is:

\[
Q(x) \approx Q(x_0) + \frac{dQ}{dx}(x_0)(x - x_0) \quad (23)
\]

where \( x_0 \) is called the operating point. If \( x \) is close to \( x_0 \), the approximation is accurate. If the control input \( T \) operates in
a narrow range, we only need to choose a single operating point $T_0$, otherwise we need choose a set of operating points $T_i$ for better accuracy, where $i = 0, 1, 2, ..., N$. For each $T_i$, there is an corresponding approximation:

$$Q(x) \approx Q(x_i) + \frac{dQ}{dx}(x_i) \cdot (x - x_i)$$

where $x_i = \frac{T_i - \mu}{\sigma}$ or $x_i = \frac{\mu - S}{\sigma}$ in our problem. We now derive the approximation for Eqn. (21). We calculate the first order derivative of $\mathbb{E}[u]$ with respect to variable $T$. According to the linear combination rules, we can have:

$$\frac{d\mathbb{E}[u]}{dT} = K_2 \left[ (1 - P_a) \frac{d[Q(T - u)]}{dT} + P_a \frac{d[Q(T - u - S)]}{dT} \right]$$

According to Eqn. (22),

$$\frac{d\mathbb{E}[u]}{dT} = -\frac{K_2}{\sigma \sqrt{2\pi}} \left[ (1 - P_a) \exp \left( -\frac{(T - u)^2}{2} \right) + P_a \exp \left( -\frac{(T - u - S)^2}{2} \right) \right]$$

Letting $F$ denotes $\frac{d\mathbb{E}[u]}{dT}$, then, we can derive linearized approximation of Eqn. (21) as,

$$\mathbb{E}(T) \approx \mathbb{E}(T_i) + F(T_i) \cdot (T - T_i) \quad (24)$$

where, $i = 0, 1, 2, ..., N$. When $T$ only has a single operating point, we have $i = 0$. Based on the above linear approximation, we now derive the expressions of $G_p(z)$ and $H(z)$, and design $G_c(z)$ to solve Problem 1. By taking $z$-transform to Eqn. (24), we have $G_p(z) = \frac{\mathbb{E}[u]}{\mathbb{E}[u]}(T_i)$, where $T_i$ is obtained every monitoring cycle by solving Eqn. (24) at the setpoint $u_s$. We can estimate $\mathbb{E}[u]$ based on the samples of CPU utilization in a control cycle. This estimation will then be fed back to compare with setpoint $u_s$. So $H(z) = z^{-1}$. $G_p(z)$ is a zero-order system to be controlled. A first-order controller is sufficient to achieve the stability and convergence of the closed-loop system. Therefore, we let $G_c(z)$ be

$$G_c(z) = \frac{a}{1 - b \cdot z^{-1}} \quad (25)$$

where $a > 0$ and $b > 0$. The coefficients $a$ and $b$ should be chosen to ensure the system stability and convergence.

B. Stability and Convergence

We first analyze the system stability. The closed-loop transfer function, denoted by $T_c(z)$ is given by

$$T_c(z) = \frac{G_c(z)G_p(z)}{1 + G_c(z)G_p(z)H(z)} = \frac{a \cdot F(T_i) z}{z - b - a \cdot F(T_i)} \quad (26)$$

The closed-loop system has a pole at $z = b - a \cdot F(T_i)$. From control theory [17], if the pole is within the unit circle centered at the origin, i.e., $|b - a \cdot F(T_i)| < 1$, the system is stable. Therefore, considering the fact that $F(T_i) < 0$, the sufficient condition for stability is $\frac{2}{b + 1} < a < \frac{2}{b + 1}$. For analyzing steady-state error of the system, the open-loop transfer function, denoted by $T_0(z)$ is given by

$$T_c(z) = G_c(z)G_p(z)H(z) = \frac{a \cdot F(T_i)}{z - b}$$

By letting $b = 1$, the system is a type I system [17], in which the controlled variable $\mathbb{E}[u]$ can converge to the reference $u_s$ provided that the system is stable. Therefore, by replacing $b$ with 1, the condition for both stability and convergence is $\frac{2}{F(T_i)} < a < 0$.

According to Fig. 2, we have $G_c(z) = \frac{\mathbb{E}[u]}{u_s - H(z)\mathbb{E}[u]}$. As we defined $G_c(z)$ in Eqn. (25) and $H(s) = z^{-1}$, we have the $z$-domain equation $T(z) = b \cdot T(z) z^{-1} + a \cdot (u_s - z^{-1} \mathbb{E}[u] z)$. The $z$-domain equation in the expression of $T(z) = b \cdot T[n - 1] + a \cdot [u_s - \mathbb{E}[u](n - 1)]$. For the expected execution time $e$, it is calculated every control cycle by averaging the monitored execution times of imaging processing and target recognition tasks. To estimate the parameters of target and noise profiles $\mu, \sigma, S$, we employ the K-means [16] classification algorithm to construct two different clusters of sensor measurements which correspond to the cases of target absence and presence, respectively. The objective of the K-means algorithm is to divide $N$ data elements into two clusters while the following square error is minimized.

$$I = \sum_{j=0}^{N_i} \sum_{i=0}^{N_j} |x_i - c_j|^2 \quad (27)$$

where $c_j$ is the centroid of each cluster and $x_i$ represents data elements. $N_0$ and $N_1$ are the numbers of data elements in two clusters. Typically several iterations are needed to reassign the data elements for each cluster before the objective function Eqn. (27) is minimized. The clustering process terminates when $I$ cannot be further reduced. In our solution, the cluster head node executes a simple heuristic algorithm as follows. Initially, it chooses two sensor measurements arbitrarily as the centroids of the two clusters that correspond to measurements of noise and target, respectively. When a new sensor measurement is received, it will be assigned to the cluster that leads to the smaller $I$. Moreover, the corresponding $\mu, S, \sigma$ will be estimated as follows. The mean of Gaussian noise $\mu$ is estimated by averaging the measurements in the cluster representing the noise while target energy $S$ can be calculated by subtracting $\mu$ from the average of measurements in the cluster representing the target. The variance $\sigma^2$ is estimated by averaging the variances from two clusters.
D. Optimizing Detection Error Rate

Our discussion so far is only concerned with controlling the utilization bound while the impact on detection error rate is not considered. Although such a solution can meet the deadline once a target is detected, it may lead to low system fidelity such as excessive target misses. In this section, we discuss how to optimize system detection performance without violating the utilization bound.

According to our fusion model (Eqn. (7)), the detection performance is optimized if the fusion threshold is set to \( T_{\text{opt}} \). However, we cannot simply adjust the current threshold to \( T_{\text{opt}} \) without accounting for the impact on CPU utilization. FAUC addresses this issues by implementing a dual-cycle control strategy. In each control cycle, the control algorithm described earlier is used to enforce the current utilization bound that is initially set to user-specified constant \( u_s \). Each optimization cycle consists of multiple control cycles. The plant parameters are estimated each optimization cycle (as discussed in Section VI-C) and then used to update \( T_{\text{opt}} \) and compute the expected utilization \( u^* \) according to our utilization model (Eqn. (21)). If the estimated utilization is lower than the initial utilization bound, i.e., \( u^* < u_s \), it will be set as the new control objective for the following control cycles until the start of next optimization cycle. It can be seen that the optimization process opportunistically lowers the utilization bound if system detection performance can be optimized. In other words, the CPU utilization never exceeds the initial value specified by user. Therefore, the real-time schedulability is always satisfied.

VII. EXPERIMENTATION

To evaluate the performance of our approach to fidelity-aware CPU utilization control, we have conducted two testbed experiments for detecting light and acoustic targets. The results allow us to evaluate our approach for different sensor modalities. Section VII-A discusses the experimental methodology we employed for the evaluation. Section VII-B and VII-C discuss the experimental settings and results, respectively.

A. Experimental Methodology

We implement two baseline approaches for performance comparison: (1) The open-loop approach in which the noise and signal profiles are measured in an offline experiment and a constant fusion threshold is chosen according to Eqn. (7). As the threshold remains constant, this approach is not expected to handle significant dynamics such as variations of noise and target profiles. (2) The fixed-step heuristic closed-loop approach, the expected CPU utilization \( E[u] \) is fed back to compare with the setpoint \( u_s \). As \( E[u] \) is a decreasing function of threshold \( T \); if \( E[u] > u_s \), the new threshold \( T[n] \) is calculated by adding a fixed-step \( \Delta_T \) to the previous threshold \( T[n-1] \); otherwise \( \Delta_T \) is subtracted from previous \( T[n-1] \) to calculate new threshold \( T[n] \) if \( E[u] < u_s \). However, this design approach does not consider system stability and convergence. We employ different \( \Delta_T \) for this approach to evaluate the impact of step size on system performance.

Our evaluation is focused on two performance metrics: utilization error and average system detection rate. The utilization error is computed as the absolute error between the measured CPU utilization and the utilization setpoint in each control cycle. For both light and acoustic target detection, we record the ground truth information about target appearance and compute system false alarm and missing rates in each control cycle. The overall system detection rate is the sum of two rates.

B. Light Target Detection

In the light target detection experiment, four TelosB motes are attached against the LCD screen of a desktop computer to detect a light spot displayed on the screen (see Fig. 3). The light spot is controlled by a computer program, simulating the random presence of the target at the probability of \( P_a = 50% \) in each one-second time slot. The similar method has been adopted by previous work [22]. We vary the light intensity of each pixel on the LCD screen to simulate the changeable characteristics of noise and target. To create noise on sensor measurements, each pixel is assigned a small random light intensity value \( I_N \) with the mean of \( \mu \). \( I_N \) varies over time to simulate the changing environmental noise. To create the target, a constant light intensity value \( I_T \) is added to the noise for each pixel. Similarly, \( I_T \) can vary to simulate the changes of target profile. The base station is implemented by Java on a laptop.

Four TelosB motes measure the light intensity every 250 milliseconds via on-board Hamamatsu S1087-01 light sensors [1] and transmit to the cluster head connected to the base station laptop. The cluster head fuses the readings received within every 250 milliseconds and detects the light spot. A webcam is attached against the LCD screen and serve as the high-quality sensor. When the cluster head node makes a positive decision, the webcam will be triggered to take a picture and compare the average intensity over all pixels with a threshold. The false alarm and missing rates of webcam, i.e., \( P_{FH} \) and \( P_{MH} \), are 5.1% and 3.9%, respectively, which are estimated in a separate offline experiment.

To compare the performance of different approaches, we conducted experiments under a variety of settings. Each experiment consists of two phases where the noise or target profiles are different. In the first phase, we set the initial \( I_N \) and \( I_T \) for the noise and target light intensities, respectively. The second phase starts after 3500 control cycles, where we vary \( I_N \) or \( I_T \) to simulate the noise and target profile change.

In Section III, we assume that the sensor measurements follow the normal distribution. We now verify this assumption using real data from a light sensor. Fig. 5 shows that the light sensor measurements fit well with the normal distribution \( N(74.63, 5.79) \). Similar results have been observed for other sensors. KAUC employs the K-means algorithm to estimate the noise and target profiles online and update the parameters of controller. Moreover, the results will also be used to compute the optimal fusion threshold. Therefore,
the accuracy of estimation may impact the performance of FAUC. Fig. 6 shows the CDF of relative errors between ground truth and estimated mean and variance of noise energy and the mean of target energy. We can see that both means of noise and target are estimated accurately. About 40% of the noise variance estimates have a near-zero error and the maximum error is only about 2%.

1) Micro-scale Performance Analysis: We first study the micro-scale behavior of KAUC in dynamic environments. Fig. 9 shows the temporal evolution of the system when noise energy is increased at the 12th control cycle. Each optimization cycle comprise five control cycles. The initial CPU utilization setpoint is 0.62. Based on this setpoint, KAUC computes an initial fusion threshold $T$ based on the estimated target/noise parameters. As $T > T_{opt}$, there hence exists an opportunity
to reduce system false alarms. KAUC thus computes a new utilization bound ($0.38$) based on $T_{\text{opt}}$ and updates the setpoint, which causes the controller to increase $T$. It can be seen from the middle and bottom figures that the measured utilization quickly converges to the new setpoint and system error rate drops to zero.

When the noise energy is increased at the 12th control cycle, the CPU utilization increases accordingly because false alarms from low-end sensors trigger new image processing tasks. KAUC then attempts to drive the utilization lower by increasing $T$. At the next optimization cycle (the 15th control cycle), KAUC estimates system parameters and computes a new $T_{\text{opt}}$ that is lower than $T$. As the corresponding utilization ($\sim 0.6$) is still lower than the initial bound 0.62, KAUC increases utilization setpoint to reduce false alarms. At the next optimization cycle (the 20th control cycle), $T$ has exceeded the current $T_{\text{opt}}$. KAUC then lowers the utilization setpoint again, which frees more CPU resources. The above results demonstrate several salient features of KAUC when it operates in dynamic environments. First, it yields satisfactory control performance as the CPU utilization quickly converges to the setpoint even when false alarms introduce unpredictable system workloads. Second, FAUC can effectively adapts utilization setpoints to minimize system error rate.

2) Performance Comparison: We now compare the performance of different approaches in 7 experiments with a variety of target/noise dynamics. In the first four experiments, we change the mean of noise energy from 20 to 40 at a step of 20. In the last three experiments, we decrease the mean of target energy by 30, 50, and 70, respectively. Fig. 7 shows the average detection error rate in these 7 experiments. We can see that all three approaches can maintain small detection error rate when the noise energy increases. The increasing noise energy will cause more false alarms for low-end sensors. However, when cameras are activated, the false alarms can be effectively filtered out based on accurate image processing. When the target energy decreases, all three approaches yield higher error rates. In particular, the closed-loop heuristic with step size 5 misses about 16% targets. When the step size is 20, it leads to fewer misses because the controller settles faster. FAUC achieves the minimum error rate under all settings.

Fig. 7 shows the CDF of the absolute utilization errors in 7 experiments. We can see that FAUC significantly outperforms both heuristics. In particular, about 80% of errors fall within 10%. On the other hand, the two heuristics do not take advantage of the feedback on utilization measurements and hence yield considerable control errors.

C. Acoustic Target Detection

We further evaluate our approach in another experiment where three Iris motes [1] and a camera detect a moving the radio-controlled toy car (see Fig. 4). The cluster head is attached to the base station laptop. The microphones of MTS300 sensor board [1] on Iris motes sample the environment at 100 Hz to detect the acoustic signals from the car’s electrical motor. Each mote calculates the acoustic signal energy every 50 samples and transmits to the cluster head every 500 ms. The cluster head fuses the acoustic signal energies received from Iris motes and activates the webcam to capture an image if a positive decision is made. The base station compares the image with the stored background image to detect the target. Another webcam attached to another desktop is used to record the ground truth information. In this experiment, the characteristics of both noise and target are subject to significant dynamics. As the toy car moves through the surveillance region, the acoustic signal energy emitted varies with the car’s location and speed. Moreover, there exist constant acoustic noises such as air conditioning in the office environment where this experiment is conducted.

Fig. 10 shows the process of the acoustic testing. The CPU utilization setpoint is set to 0.2. Each control cycle comprises 70 samples and the estimation cycle is four times longer than the control cycle. Initially the threshold is set to a lower level. It should cause the high false alarm, however, it didn’t. One observation from this experiment is that, low end sensor missing probability contributes nearly all the average error rate. The controller adjust threshold to control the CPU utilization around this setpoint. In this acoustic detection scenario, the target energy varies more than the light spot testbed experiment as we expected. Also the target presence probability varies. From the experiment result, the CPU utilization is controlled to the setpoint. From the third to the fifth control cycle, the CPU utilization is increased. This is due to the fact that the target presence increased. In this case, the threshold of the decision is increased by the controller and in the sixth cycle, the utilization drop back to the setpoint. From the sixth to eleventh cycle, we can see another adjustment because of the increasing target presence probability. In this test, we see the average error rate is relatively high. This is caused by missing the target due to the low end sensors. As we can see the the low average target present energy varies a lot in the first chart of Fig. 10. In fact, the variance of the target is also high, which cause that the target cannot be detected.

VIII. Conclusion

In this paper, we propose a holistic approach called Fidelity-Aware Utilization Controller (FAUC) to address both fidelity and timeliness requirements of wireless cyber-physical surveillance (WCS) systems. FAUC integrates data fusion with feedback control to enforce CPU utilization upper bound although system workloads vary significantly at runtime due to stochastic detection results. FAUC also optimizes system fidelity and adjusts the control objective of CPU utilization adaptively in dynamic environments. We have implemented FAUC on a small-scale WCS testbed consisting of TelosB/Iris motes and cameras. Our experiments on light and acoustic target detection show that FAUC can achieve robust fidelity and enforce desired utilization bounds in the presence of significant variations of target/noise characteristics.
REFERENCES