2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Event-driven Gaussian Process for Object Localization in Wireless Sensor Networks

Jae Hyun Yoo, Woojin Kim, and H. Jin Kim

Abstract— **Object localization using wireless sensor networks (WSN) often requires data from many sensor nodes and different types of sensors for position estimation. This incurs a heavy communication load, which can cause packet loss, communication delay and much energy consumption, deteriorating the performance of object localization. Here we employ an event-driven Gaussian process in order to learn the position of an unknown object using WSN with multiple types of sensors. In the event-driven framework, each sensor node transmits data only when decision criteria are satisfied. We consider the error-bounded algorithm as the decision criteria based on the measurement history of each sensor node. The overall communication between sensor nodes is reduced, thus increasing energy-efficiency of the network and relieving the concentration of communication traffic at the base node. Experiments to track the position of a mobile robot are conducted using a multi-sensor WSN, and the comparison is made between the event-driven framework and the conventional approach in which sensors transmit data at a constant sampling rate. Experimental results demonstrate the efficiency and accuracy of the proposed event-driven Gaussian process approach.**

I. INTRODUCTION

Wireless sensor networks (WSNs) consist of sensor nodes that collect the data using various sensors depending on the type of application [1], and wireless communication through which the data is transmitted between the sensor nodes. Due to their simple deployment, WSNs are employed in various applications such as target detection, recognition, localization [2] [3], health monitoring [4] and navigation [5].

In most cases, the sensor nodes are powered by independent sources such as onboard batteries, so the lifespan of the sensor node is often determined by the amount of processing and communication it carries out. Clearly, reducing the communication between sensor nodes while maintaining the overall performance of the WSN will increase the energy-efficiency of the network. Also, the reduction of communication relieves the excessive concentration of communication traffic at the base node. The issue of excessive traffic at the base node has been shown to dramatically degrade the overall performance of the network [6][7].

Event-driven frameworks can be a solution for reducing communication in WSN. Previously, event-driven algorithms have been studied for medium access control (MAC), and also, applications using event-driven algorithms have been reported for data collection [8], networked control [9], and field estimation [10]. Usually, event-driven algorithms employ a global criterion which decides if a sensor node should transmit or not. Since the criterion is usually analytically motivated, it can be used to guarantee stability or convergency of the algorithms. However, global criteria need constant communication between nodes. To deal with such an issue, a practical heuristic called the *error-bounded algorithm* has been proposed [11]. In this algorithm, each node uses a decision criterion based on its measurement history to decide whether to transmit data or not. Thus, constant communication between the nodes is not needed, reducing the overall communication load.

Previous research [12] on an event-driven method for WSNs uses a global criterion to estimate object position without learning-theoretic concepts. Thus, the true performance in realistic setting will be affected by communication capacity and network topologies. Another previous research [13] employs an event-driven approach to reduce energy consumption in real experiments, but it does not exploit the usefulness of data for a specific purpose such as localization. Our contribution lies in an application of error-bounded algorithm as a simple transmission criterion at each node in order to reduce the communication without sacrificing the accuracy, in combination with a model-based Gaussian process for position estimation.

We consider the problem of tracking the position of a mobile robot using a multi-sensor wireless sensor network. Various measurement values such as received signal strength indicator (RSSI), infrared (IR) sensor and magnetic sensor readings are used to track the mobile robot. Since each type of sensor has different properties, we conduct a sequence of experiments to find suitable parameter values of the error-bounded algorithm for each sensor. We experimentally compare the performance of the event-driven framework with conventional approaches where sensor nodes transmit data at a fixed sampling rate. In doing so, we use the best sampling rate and the number of nodes to maximize the data success rate for the conventional approaches.

For the position estimation of the mobile target, we employ Gaussian process (GP) which is a non-parametric machine learning technique. Due to the flexibility in modeling complex expressions using a small number of parameters, GP has been widely used to approximate a number of nonlinear models such as wireless signal strength maps [14], human motion [15], optimal sensor placement [16]. However, a large amount of data as in a large-scale WSN can pose a great difficulty for GP learning, and the proposed event-driven approach can be an efficient solution.

This work was supported by MEMS Research Center for National Defense funded by Defense Acquisition Program Administration (4000631211).

The authors are with the School of Mechanical and Aerospace Engineering, Seoul National University, Seoul, South Korea {yjh5455,sean84dy,hjinkim}@snu.ac.kr

978-1-61284-456-5/11/$26.00 ©2011 IEEE

The rest of the paper is organized as follows. In Section II, Gaussian process is briefly described. The event-driven framework and the error-bounded algorithm are described in Section III. In Section IV, experimental results are presented using the proposed event-driven Gaussian process method, together with performance comparison with the conventional approach. Also, we describe multi-sensor modeling and parameter selection for the error-bounded algorithm. Finally, Section V is devoted to concluding remarks.

II. GAUSSIAN PROCESS

We are interested in building a regression model, by interpreting the position of the i-th sensor node as an input vector $\mathbf{x}$ of dimension d: $\mathbf{x}_i \in \mathbb{X} \subseteq \mathbb{R}^d$, and the sensor measurement $y_i \in \mathbb{Y} \subseteq \mathbb{R}$ as the output. Based on the input-output data, a regression function $f : \mathbb{X} \to \mathbb{Y}$ can be obtained to predict values at any location in $\mathbb{X}$. A Gaussian process (GP) defines a distribution over a function-space as described in [17]. Let $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a sample set of f obtained from

$$y_i = f(\mathbf{x}_i) + \omega_i, \tag{1}$$

where ω_i is a Gaussian noise variable whose value is chosen independently for each observation i, and assumed to have zero mean and a variance of σ^2, i.e. $\omega_i \sim \mathbb{N}(0, \sigma^2 I)$, where I is a $n \times n$ identity matrix. We can sample the jointly Gaussian data set $y_1, \cdots, y_n$ from the GP distribution:

$$Y \sim \mathbb{N}(0, \Sigma + \sigma^2 I),$$

where

$$Y = [y_1, y_2, \cdots, y_n]^T, \quad X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n],$$

and Σ is a kernel matrix whose (i, j)-th element is

$$\Sigma_{ij} = \mathbb{K}(\mathbf{x}_i, \mathbf{x}_j). \tag{2}$$

In (2), $\mathbb{K}$ is Gaussian kernel function defined as

$$\mathbb{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\theta}\right),$$

where θ is hyper-parameter selected by a sensor range. The wider sensor range is, the larger θ is chosen. We are interested in predicting value of the Gaussian process at an arbitrary position $\mathbf{z}$, where the predicted value has a Gaussian with mean $\mu_{\mathbf{z}}$ and variance $\sigma_{\mathbf{z}}^2$, conditioned on the training data X, Y:

$$f(\mathbf{z}) \sim \mathbb{N}(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}^2),$$

where

$$\begin{aligned}\mu_{\mathbf{z}} &= k_{\mathbf{z}}^T (\Sigma + \sigma^2 I)^{-1} Y,\\ \sigma_{\mathbf{z}}^2 &= \mathbb{K}(\mathbf{z}, \mathbf{z}) - k_{\mathbf{z}}^T (\Sigma + \sigma^2 I)^{-1} k_{\mathbf{z}},\end{aligned}$$

with

$$k_{\mathbf{z}} = [\mathbb{K}(\mathbf{x}_1, \mathbf{z}), \mathbb{K}(\mathbf{x}_2, \mathbf{z}), \cdots, \mathbb{K}(\mathbf{x}_n, \mathbf{z})]^T.$$

The $\mu_{\mathbf{z}}$ is the predictive value at the position $\mathbf{z}$.

```
At time step $k$ compute:
$\hat{y}_i[k] = \alpha y_i[k] + (1-\alpha)\hat{y}_i[k-1]$
in Case 1:
  if $|y_i[k] - \hat{y}_i[k]| > \varepsilon_1$ and $y_i[k] > \varepsilon_2$
      Send $y_i[k]$
  end if
in Case 2:
  if $|y_i[k] - \hat{y}_i[k]| > \varepsilon_1$
      Send $y_i[k]$
  end if
```

Fig. 1: The error-bounded algorithm.

```
Procedure:
  1 : Each node measures $(y_1, y_2, \cdots, y_m)$
  2 : Each node runs the error-bounded algorithm
  3 : Each node decides whether to send $y_i$ or not
  5 : Base station collects $(y_1, y_2, ...y_n)$
  6 : Compute Gaussian process to obtain
      $\Phi$ = $[\mu_{\mathbf{z}_{11}}, \cdots, \mu_{\mathbf{z}_{PQ}}]$ where
      $z_{pq} = (x_p, y_q)$ is the position of the cell $(p, q)$
  7 : $\hat{\mathbf{z}} = \mathbf{z}_{p^*q^*}$ such that $\mu_{\mathbf{z}_{p^*q^*}} = \max_{p,q} \Phi$
Output: Estimated target position $\hat{\mathbf{z}}$
```

Fig. 2: The event-driven Gaussian process.

III. EVENT-DRIVEN GAUSSIAN PROCESS

In an event-driven framework, each node determines whether to transmit data or not using a criteria based on its measurement history. The first intuitive criterion one can think of would be the difference from a nominal value :

$$|y_i[k] - y_n| > \varepsilon, \tag{3}$$

where $y_i[k]$ is the measurement made by node i at time step k, y_n a predetermined nominal value of the measurement, and $\varepsilon > 0$ a threshold value. If a sensor node uses (3) as the criterion, it would transmit data when the difference between the measurement at step k and the predetermined nominal value exceeds the threshold value. This can be suitable when the purpose of the data acquisition is to monitor some variable which should supposedly be at some nominal value. For a broader range of applications such as a problem of tracking the position of a mobile target, however, (3) becomes less useful, as the sensor nodes measure some sense of distance between the node and the target.

In [11], the following criterion was proposed:

$$|y_i[k] - \hat{y}_i[k]| > \varepsilon, \tag{4}$$

where $\hat{y}[k]$ is updated according to

$$\hat{y}_i[k+1] = \alpha y_i[k] + (1-\alpha)\hat{y}_i[k], \quad \alpha \in (0, 1). \tag{5}$$

According to (5), sensor nodes transmit when there is a large change in the measurement data, or, when abnormalities are detected. This can be clear if we look at (5) as a first-order filter where $\hat{y}_i$ is the filtered value of y_i. Then, according to (4), the sensor nodes will transmit if the difference between the measurement y_i and the filtered value $\hat{y}_i$ exceeds ε. The

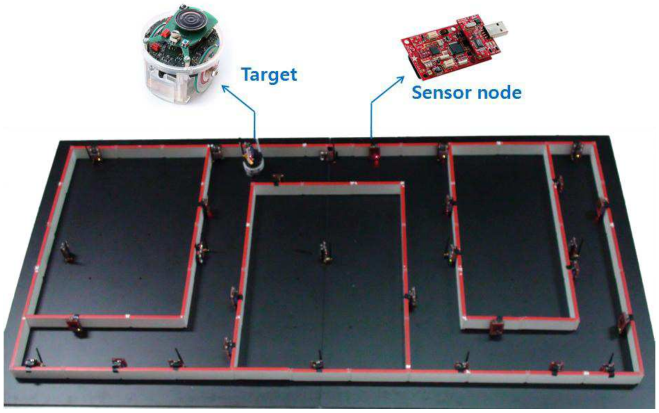


Fig. 3: Experimental setup for the evaluation of the event-driven Gaussian process with 30 sensor nodes and one moving target.

sensitivity to the measurement change can be adjusted by α. With a small value of α, $\hat{y}_i$ will be relatively slow in tracking y_i, thus triggering data transmission more often. On the other hand, with a large value of α the filtering will be relatively fast, insensitive to moderate changes.

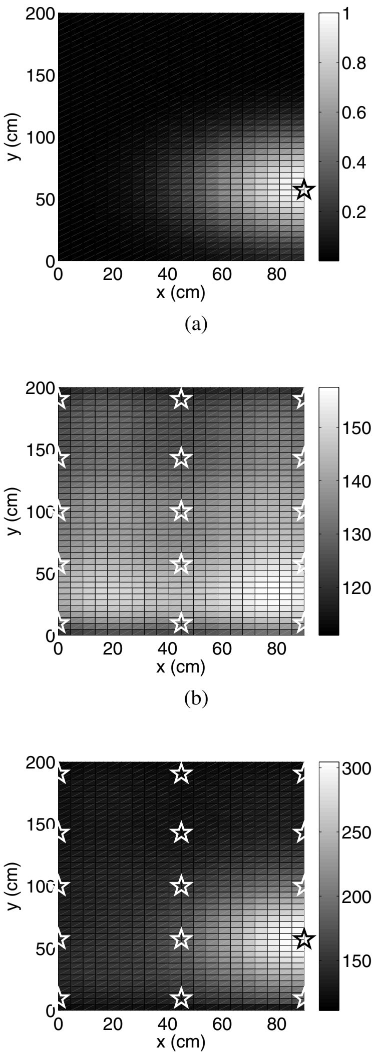
Fig. 1 shows the error-bounded algorithm that uses the criteria (4) with the update law (5). Because we use various sensors that have different properties, we divide the algorithm into `Case 1` for RSSI and `Case 2` for IR and magnetic sensor data. This issue and the determination of parameter values will be further discussed in Section IV. Fig. 2 shows the overall event-driven Gaussian process, where m denotes the total number of nodes, and n the number of nodes that have transmitted data. $\Phi \in \mathbb{R}^{PQ}$ is a vector containing predictive values where z_{pq} is the position (x_p, y_q) of the cell (p,q), $1 < p < P, 1 < q < Q$ in Fig. 4.

IV. Experimental Evaluation

A. Experimental setup

In the experimental setup, the total of thirty sensor nodes (UBee430) are deployed using IEEE 802.15.4 and TinyOS. A mobile robot (E-puck) is used as a target for position estimation, whose size is 8 cm × 8 cm [19]. In a 90 cm × 200 cm workspace, a road is made for the mobile robot to move along. Fig. 3 shows the experiment setup with an E-puck robot and UBee430 sensor nodes.

A mobile robot whose location is unknown to the WSN emits RF signal while moving in the workspace. The RF signals are retrieved by the sensor nodes, and the sensor nodes calculate the RSSI [20]. Because the robot uses IR signals for collision avoidance, IR sensors are also used. Finally magnetic sensors are used to measure the change in the magnetic field induced by the robot's electric motor. A PC is connected to the base station (often referred to as the super node or sink) and estimates the position of the mobile robot by running the GP regression. To evaluate the estimation accuracy, the true position of the mobile robot is measured by the Vicon Motion System [18] composed of four cameras that track reflective markers attached to the mobile robot.



(a)

(b)

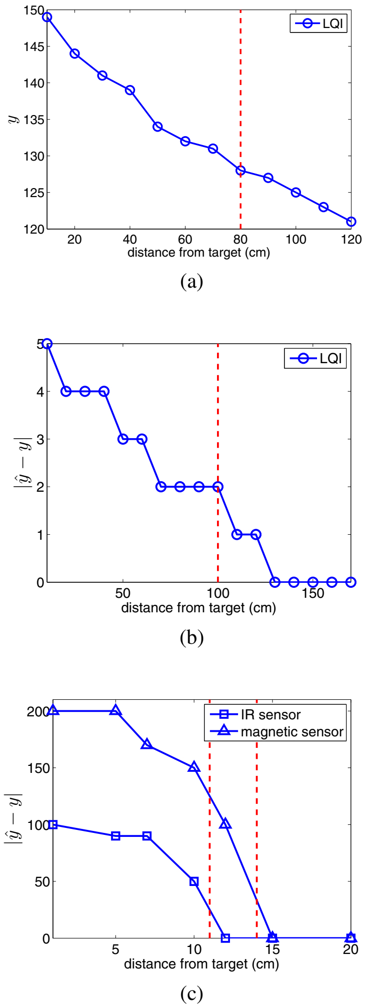
(c)

Fig. 4: Predictive distribution according to (a): the model using sensor data of binary type only with $\theta = 5$, (b): the model using RSSI data only $\theta = 15$, (c): the combined model of RSSI and binary data

B. Multi-sensor modeling for Gaussian process

We can use GP distribution to provide model confidence at any location or area similar to [21]. Because every sensor has a limited range and limited applicability, we use multiple sensors to obtain a more accurate model. As preprocessing prior to GP, we convert the measurement obtained from IR and magnetic sensors to binary values as (6). This is motivated by the fact that IR and magnetic sensors have a short range, and their measurements are very small and useless when the target is out of range.

$$y_i = \begin{cases} 1, & \text{if} \quad y_i \geq \varepsilon \\ 0, & \text{else} \end{cases} \qquad i = 1, \ldots, m \tag{6}$$



(a)

(b)

(c)

Fig. 5: (a): Measurement y of RSSI data, (b): $|\hat{y}-y|$ of RSSI data, (c): $|\hat{y}-y|$ of IR and magnetic sensor data

We determine ε to represent the value of y_i where y_i varies drastically. The corresponding result is shown in Fig. 4(a) where black stars denote the binary nodes. The darkness of each cell represents the possibility of target occupancy, with white denoting highest likelihood. Because the sensing range of RSSI sensors is large enough, all the nodes that obtain RSSI data are always used in our experiment. Fig. 4(b) depicts the model using RSSI data where white stars show that all the sensor nodes have been used. When both RSSI and binary data are used at the same step, the binary data is scaled up to the data range of the RSSI-only model in order to give them the similar contribution. This result is depicted in Fig. 4(c).

C. Parameter selection in error-bounded algorithm

As mentioned in Section III, α is an important parameter in the error-bounded algorithm, as it determines the sensitivity to changing measurements. The value of α is appropriately selected by considering the sampling period of nodes and the speed of the moving target. For example, if α is too large, the convergence of $\hat{y}$ to y in 4 will be quite fast, which can lower the accuracy for a fast moving target. A similar situation can happen with large α for a short sampling period. Also, the threshold values ε_1 and ε_2 clearly affect the operation of the algorithm. In order to select appropriate values for the thresholds, with the selected value of α, we perform an experiment in which the target approaches a sensor node along a straight line.

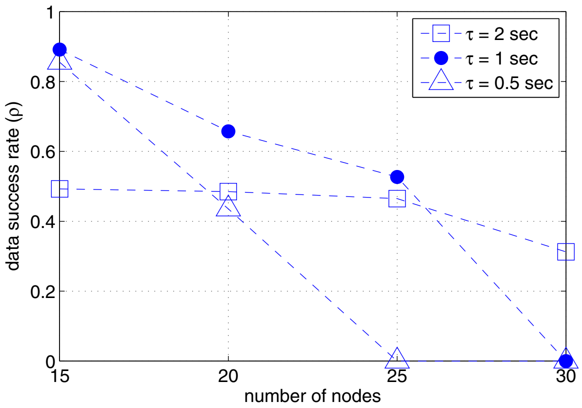


Fig. 6: Analysis of data success rate ρ with variation of number of nodes (m) and sampling period (τ)

Figs. 5(a) and 5(b) show the values of $\hat{y}$ and $|y-\hat{y}|$ for the RSSI data with respect to the distance between the target an a node. As Fig. 5(b) illustrates, $|y-\hat{y}|$ is small regardless of the distance from the target, thus, a sensor node far away from the target can be erroneously triggered by a fault. Thus, we insert one more threshold condition in order to exclude measurements obtained far away from the target. This corresponds to `Case 1` in the error-bounded algorithm (Fig. 1). On the other hand, Fig. 5(c) shows $|y-\hat{y}|$ of the IR sensor data and the magnetic sensor data. Sensor nodes using IR or magnetic sensor correspond to `Case 2` in the error-bounded algorithm.

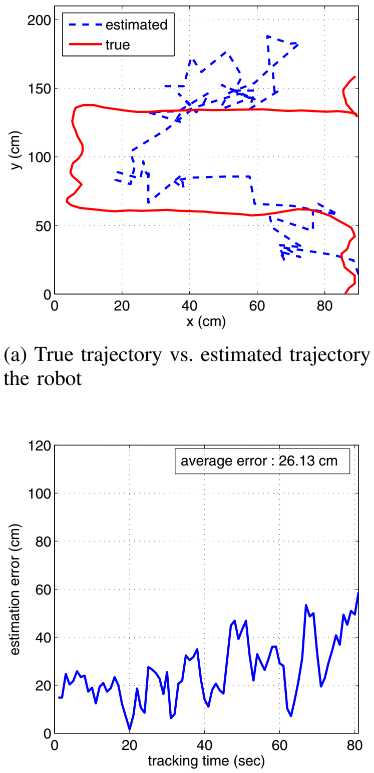
The vertical dashed lines in Fig. 5 denote the maximum distances upto which we want the sensors to react. Based on the experiment results, we set $\varepsilon_1 = 2$ and $\varepsilon_2 = 127$ for the RSSI, $\varepsilon_1 = 20$ for the IR sensor, and $\varepsilon_1 = 50$ for the magnetic sensor.

D. Data success rate

In order to compare the proposed event-driven framework with the (best-case) conventional approach, we search for the optimal sampling period and number of nodes such that the conventional approach maximizes its performance. We evaluate the performance of the conventional approach by using a parameter called *data success rate* defined as:

$$\rho = 1 - \frac{\frac{1}{T}\left(\sum_{k=1}^{T}(m-n[k])\right)}{m},$$

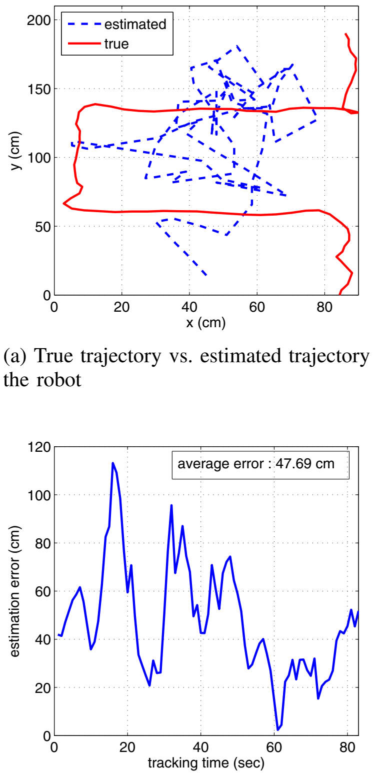
where $n[k]$ denotes the total number of nodes which successively sent data to the base node at time step k, and T is the total number of time steps. For a large value of m, ρ tends to decrease because of excessive traffic concentration



(a) True trajectory vs. estimated trajectory of the robot

(b) Error between the true position and the estimated position

Fig. 7: Experimental results of the conventional approach with $m = 15$ nodes and sampling period $\tau = 1$ sec



(a) True trajectory vs. estimated trajectory of the robot

(b) Error between the true position and the estimated position

Fig. 8: Experimental results of the conventional approach with $m = 30$ nodes and sampling period $\tau = 2$ sec

at the base node. Also, the sampling period τ effects ρ as well, because of the limited processing capability of the base node. In order to survey data success rate at the base node from the sensor nodes, several experiments were conducted using different values of τ and ρ. From Fig. 6, we conclude that $\tau = 1$ and $m = 15$ maximizes the performance of the conventional approach in our experimental setup.

E. Object tracking results

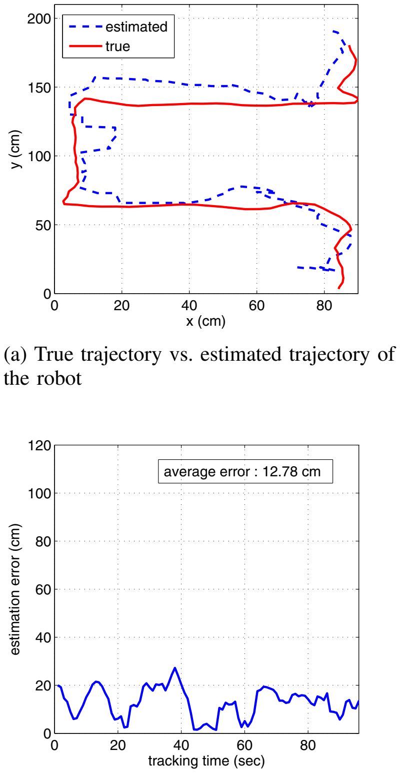
We compare the performance of the proposed method with the conventional methods. Based on the results from Fig. 6, we perform two different conventional experiments. The first uses $m = 15$ and $\tau = 1$. The result is illustrated in Fig. 7. The second uses $m = 30$ and $\tau = 2$ shown in Fig. 8. In Figs. 7(a) and 8(a), the true trajectory of the target is represented by solid line and estimated trajectory of the target is drawn by dash line. Figs. 7(b) and 8(b) show errors per sampling period and average errors. The better tracking performance of the first experiment can be explained by the fact that the data success rate with $m = 15$ and $\tau = 1$ is higher than with $m = 30$ and $\tau = 2$.

We then implement the error-bounded algorithm on all 30 sensor nodes and the result is illustrated in Fig. 9. In comparison with the prior experiments, it shows much better performance in the terms of tracking error. The average error is quiet small considering the size of the mobile robot and Fig. 9(a) depicts that the estimated trajectory is close to the true trajectory.

To see the communication requirements, we compute the communication frequency as the number of nodes communicating to the base node at a time step. Fig. 10 shows average communication frequency of the three experiments during tracking. It confirms that communication of the event-driven approach is less than that of the other conventional approaches.

V. Conclusion

In this paper, we present Gaussian process (GP) with an event-driven algorithm for learning the position of an object using WSNs. Although GP is suitable for representing the complex likelihood models from measurements, it can suffer from large amount of data. Specifically, excessive traffic between nodes in WSN can present a challenge for the GP computation, and time-varying data stream can further degrade its performance. Experimental results confirm that the tracking performance is very low in the conventional approach in which all the sensors transmit data at a fixed sampling rate. In particular, the performance with an increased number of sensor nodes can be worse than the performance with small size of nodes number. This suggests that the essential data for estimation is lost during heavy traffic, which causes deterioration of the performance. The error-



(a) True trajectory vs. estimated trajectory of the robot

(b) Error between the true position and the estimated position

Fig. 9: Experimental result of the event-driven algorithm with $m = 30$ nodes

bounded algorithm offers the benefit of communicating only with sensor nodes near the target without the need for shared global criteria. This leads to more efficient communication without losing meaningful data, which in turn generates more accurate GP regression. By addressing the issues related with communication capability and energy consumption while maintaining the performance, the proposed method is expected to be more effective in larger-size WSNs.

References

[1] F. Lewis, "Wireless Sensor Networks," *Smart environments: technologies, protocols, and applications,* ed. D. J. Cook and S. K. Das, John Wiley, New York, 2004.

[2] S. Zickler and M. Veloso, "RSS-Based Relative Localization and Tethering for Moving Robots in Unknown Environments," In IEEE International Conference on Robotics and Automation (ICRA), pp. 5466-5471 2010.

[3] L. Shi and J. Tan, "Distributive Target Tracking in Sensor Networks with a Markov Random Field Model," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 854-859, 2009.

[4] L. Schwiebert, S. Gupta, and J. Weinmann, "Research Challenges in Wireless Networks of Biomedical Sensors," 7th annual international conference on Mobile computing and networking, pp. 151-165, 2001.

[5] K.J. O'Hara, D.B. Walker, and T.R. Balch, "Physical Path Planning Using a Pervasive Embedded Network," IEEE Transactions on Robotics, vol. 24, no. 3, pp. 741-746, 2008.

[6] H. Zhang, A. Arora, Y. Choi, and M. G. Gouda, "Reliable Bursty Convergecast in Wireless Sensor Networks," In Proc. of the the ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 266.276, 2005.

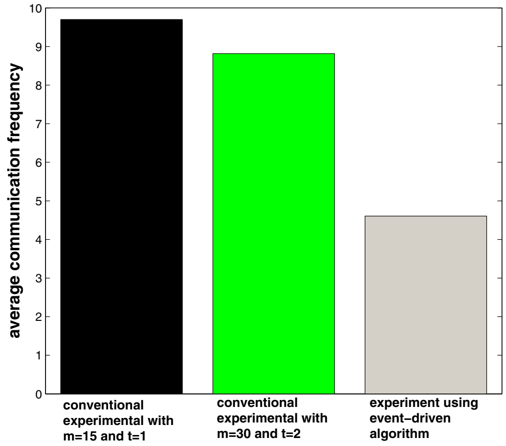


Fig. 10: Average communication frequency of the experiments shown in Figs. 7, 8, and 9

[7] A. Woo, D. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks," In proc. of the ACM International Conforence on Mobile Computing and Networking, pp. 221-235, 2001.

[8] H. Sabbineni and K. Chkrabarty, "Datacollection in Event-Driven Wireless Sensor Networks with Mobile Sinks," International Journal of Distributed Sensor Networks, vol. 2010, Article ID 402680, 2010.

[9] K. H. Johansson, "Control over Wireless Networks," CTS-HYCON Workshop, Paris, France, 2008.

[10] D. O. Cunha, O. C. M. B. Duarte, and G. Pujolle, "Event-driven Field Estimation for Wireless Sensor Networks," IFIP/IEEE International Conference on Mobile and Wireless Communications Networks (MWCN), vol. 211, pp. 89-98, 2006.

[11] D. O. Cunha, R. P. Laufer, I. M. Moraes, M. D. D. Bicudo, P. B. Velloso, and O. C. M. B. Duarte, "Bio-Inspired Field Estimation with Wireless Sensor Networks," Annals of Telecommunications, vol. 60, 2005.

[12] S. Shenoy, J. Tan, "Simultaneous Localization and Mobile Robot Navigation in a Hybrid Sensor Network," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1636-1641, 2005.

[13] Z. Zeng-wei, W. Zhao-hui, L. Huai-zhong, "An Event-Driven Clustering Routing Algorithm for Wireless Sensor Networks," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 2, pp. 1802-1806, 2004.

[14] B. Ferris, D. Fox, N. Lawrence "WiFi- SLAM using Gaussian Process Latent Variable Models," 20th International Joint Conference on Artificial Intelligence (IJCAI), pp. 2480-2485, 2007.

[15] R. Urtasun, D.J. Fleet, and P. Fua, "3D People Tracking with Gaussian Process Dynamical Models," IEEE Int. Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 238-245, June 2006.

[16] A. Krause, A. Singh and C. Guestrin. "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies," Journal of Machine Learning Research, vol. 9, pp. 235-284, 2008.

[17] Christopher M. Bishop, Pattern Recognition and Machine Learning, Spriger Press, 2006.

[18] (2011, Mar.) The VICON website. [Online]. Available:http://www.vicon.com/

[19] (2011, Jan.) The e-puck website. [Online]. Available:http://www.e-puck.org/

[20] B. Ferris, D. H*ä*hnel, and D. Fox. "Gaussian Processes for Signal Strength-Based Location Estimation," Robotics: Science and Systems, 2006.

[21] C. Stachniss, G. Grisetti, N. Roy, and W. Burgard. "Evaluation of Gaussian Proposal Distributions for Mapping with Rao-Blackwellized Particle Filters," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2007.