Experimental validation of an event-triggered policy for remote sensing and control with performance guarantees

Bas van Eekelen, Nitish Rao, B. Asadi Khashooei, D. J. Antunes and W. P. M. H. Heemels

***Abstract*—There has been a surge of interest in event-triggered control in recent years, and many event-triggered control methods are now available in the literature. As the theory matures, there is a need to experimentally validate and test these methods in applications of interest. In this paper, we extend and experimentally validate an event-triggered control strategy presented in [1] for the remote point stabilization problem for a ground robot. This strategy specifies when transmissions should occur in both sensor-controller and controller-actuator channels, and guarantees a bound on performance measured by a finite-horizon quadratic cost. The experimental results are coherent with the simulation results and reveal that event-triggered control leads to a tremendous data transmission reduction (up to 90%) with respect to period control, with a minor performance loss.**

I. Introduction

Event-triggered control (ETC) has been proposed in recent years and pertains to closing a control loop, for example by triggering transmissions between sensors and actuators or triggering control computations, only when required in order to guarantee the control specifications such as stability or closed-loop performance. This may represent a paradigm shift in digital, embedded and networked control, where periodic control schemes are prevalent, with impact on many applications where reducing data transmissions and computations in control loops is desirable. These include resource constrained application such as networked control with bandwidth limitations, wireless sensor networks with strict battery life requirements and embedded control with computational constraints.

Several event-triggered control methods are currently available in the literature. Some of the early works [2]–[7] proposed threshold policies by which transmissions are only triggered if the error between previously sent and current state and control variables exceeds given thresholds, either constant or proportional to state or control variables. Research on event-triggered control is now being pursued in several directions, such as non-linear [8]–[10], optimal [11], [12] and approximate event-triggered control [13]–[15]. However, while the event-triggered control theory is maturing at a fast pace, there are limited experimental validations of the event-triggered control methods proposed in the literature. Some experimental validations can be found in [16]–[20] but apart from these and a few others in the literature this appears to not be the main concern of the community.

The authors are with with the Control Systems Technology Group, Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands Emails: {b.v.eekelen, n.rao}@student.tue.nl, {B.Asadi.Khashooei, D.Antunes, W.P.M.H.Heemels}@tue.nl.

The purpose of the present paper is to experimentally validate the event-triggered control method proposed in [1]. This strategy specifies when transmissions should occur in both sensor-controller and controller-actuator channels, and guarantees a bound on performance measured by an infinite-horizon quadratic cost. We extend [1], by considering a performance index in terms of a finite horizon quadratic cost, as opposed to the average quadratic cost criterion considered in [1]. As in [1] we can establish that this policy provides a performance bound with respect to a periodic control scheme, where transmissions are triggered at the maximum allowable rate, while reducing the number of transmissions. The rationale behind testing the policy in [1] in an experimental setting is that while it assures a bound on the performance, it can be written in a simple form that connects well with previous thresholds policies proposed in the literature. In fact, such a policy specifies that transmissions in the controller-actuator channels should only be triggered when the error between previously sent and current control inputs exceeds a threshold. Moreover, transmissions in the sensor-controller channel should only be triggered when a function proportional to the covariance matrix of the state estimation, obtained by a time-varying Kalman filter, exceeds a threshold. Transmissions in the sensor-actuator channel can be determined a priori while transmissions in the actuator-controller channel must be determined online as these depend on the disturbance and noise realizations.

The experimental set-up consist of a robot, a wired camera, a control unit, and a wireless network. The camera is used to obtain images of the robot in its work space. The camera takes an image at designated time instances, which are defined by the controller, and sends the images to the controller. The controller receives the images from the camera and processes the images to get the estimated position of the robot with respect to a certain reference frame. Based on the estimated position of the robot the controller is able to compute new control actions. These computed control actions are sent through a wireless network to the actuators of the robot at designated time instances, which are defined by the controller.

The experiments validate the usefulness of the ETC policy derived in [1] and are coherent with simulation results also presented in the present paper. The benefits in terms of communication reduction are tremendous. In fact, the

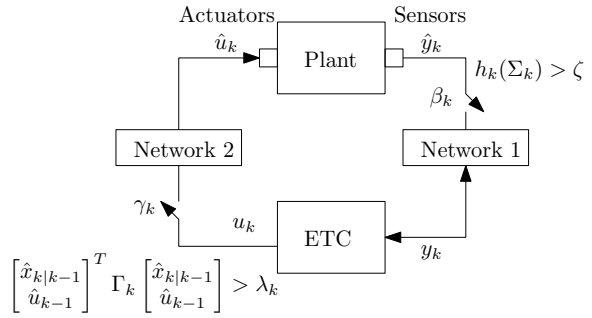

Fig. 1. Overall setup and proposed policy: sensor query depends only on the Kalman filter covariance matrices Σ_k, while the control update transmissions are scheduled based on the Kalman filter state estimate $\hat{x}_{k|k-1}$, and the previously sent control input $\hat{u}_{k-1}$.

communication is reduced by 80% and 90% for the sensor and the actuator network, respectively, while guaranteeing the performance bounds on the cost with respect to the all-time transmission policy.

The remainder of the paper is organized as follows. Section II formulates the problem. In Section III we state the control policy validated in this paper. In Section IV the physical motion system on which the control policy is implemented is discussed. Section V presents numerical simulations of the ETC policy based on the model of the motion system. Section VI presents experimental results of the ETC policy implemented on the actual motion system. In Section VII the results obtained by numerical simulations are compared to the experimental results. Section VIII provides concluding remarks and directions for future work.

A. Nomenclature

The trace of a square matrix $A \in \mathbb{R}^{n\times n}$ is denoted by $\mathrm{Tr}(A)$ and the expected value and the covariance matrix of a random vector $\eta \in \mathbb{R}^n$ are denoted by $\mathbb{E}[\eta]$ and $\mathrm{Cov}[\eta]$ respectively. For a symmetric matrix $Z \in \mathbb{R}^{n\times n}$, we write $Z \succ 0$ ($Z \succeq 0$) to denote that Z is positive definite (semi-definite). The identity map is denoted by id and $\circ$ denotes composition operator. The finite set $\mathbb{S} \in \{0, 1, \ldots, N-1\}$, $N \in Z_{\succ 0}$ is defined.

II. Problem formulation

Fig. 1 shows the block diagram of the considered system. We assume that the plant operates as a linear discrete-time system, described by

$$\begin{aligned} x_{k+1} &= Ax_k + B\hat{u}_k + v_k \\ \hat{y}_k &= Cx_k + r_k, \end{aligned} \tag{1}$$

for $k \in \mathbb{S}$ where $x_k \in \mathbb{R}^{n_x}$, $\hat{u}_k \in \mathbb{R}^{n_u}$ and $\hat{y}_k \in \mathbb{R}^{n_y}$ denote the state, the input, and the output, respectively and v_k and r_k denote the state disturbance and measurement noise at time $k \in \mathbb{S}$. We assume that the disturbance and noise processes are independent processes consisting of sequences of independent zero-mean Gaussian random vectors with covariances Φ_v and Φ_r, respectively. The initial state is assumed to be either a Gaussian random variable with mean $\hat{x}_0$ and covariance Θ_0 or known in which case it equals $\hat{x}_0$ and $\Theta_0 = 0$.

To measure performance, we consider the following finite horizon cost to be minimized

$$\mathbb{E}[\sum_{k=0}^{N-1} (x_k^T Q x_k + \hat{u}_k^T R \hat{u}_k) + x_N^T Q_N x_N], \tag{2}$$

where $Q, Q_N, R \succ 0$ are proper (positive definite) weighting matrices. By considering this performance index, the problem formulation differs form the one introduced in [1]. In fact, in [1] a discounted and an average cost were considered mainly for convenience since for such costs we were able to provide a time-invariant policy for the triggering laws and control inputs. However, in practical applications with a certain objective to be met at a certain time finite-horizon costs may be more suitable, and are therefore considered here. In particular by picking an appropriately large terminal cost we can enforce that the desired terminal conditions are (approximately) met. As shown later in this section, the scheduling and control input policy becomes in this case time-varying.

The network behavior can be modeled using scheduling vector $\sigma_k = (\beta_k, \gamma_k) \in \{0, 1\}^2, k \in \mathbb{S}$, in which $\beta_k = 1$ (or $\gamma_k = 1$) indicates the occurrence of a transmission through the network from sensor to controller (or controller to actuator) at time k and $\beta_k = 0$ (or $\gamma_k = 0$) otherwise. u_k denotes the transmitted value of the control action at time $k \in \mathbb{S}$ and y_k represents the received value of sensor data at time $k \in \mathbb{S}$. When there is no transmissions we write $u_k = \emptyset$ and $y_k = \emptyset$ to denote that they can have any arbitrary values. At the actuator side, we consider a standard zero-order hold (ZOH) device as the control-input generator (CIG) to actuate the plant with the most recent received control action.

The ETC unit in Fig. 1 controls the transmission decisions in both networks, i.e. network 1 between the sensors and the controller and network 2 between the controller and the actuators. Our solution will entail that for network 2, the controller needs only to send data when desired. However, for the network 1, the controller must first query the sensors and then receive measurement data. We assume that the delay introduced by this process is negligible and there are no packet drops in both networks.

Beside scheduling decisions, the ETC unit computes the control actions at transmission instants i.e. $\gamma_k = 1$ based on available information I_{k-1} at time k which is defined as

$$I_k := (I_{k-1}, y_k, u_k, \sigma_k)$$

for $k \in \mathbb{S}$ and $I_{-1} := (\hat{x}_0, \ \Theta_0)$. A policy $\pi := (\mu_0, \mu_1, \ldots, \mu_{N-1})$ is defined as a sequence of functions $\mu_k := (\mu_k^u, \mu_k^\sigma)$ that map the available information vector I_{k-1} into control actions u_k and scheduling decisions σ_k

$$(u_k, \sigma_k) = \mu_k(I_{k-1}). \tag{3}$$

for all $k \in \mathbb{S}$.

We denote by $J_\pi(I_{-1})$ the costs (2), when policy π is applied by the controller. Similar to [1], we are interested in a

policy that reduces the number of transmissions compared to the all-time transmission policy while keeping the performance within a desired bound of the performance of the all-time transmission policy. The all-time transmission policy is defined as $\sigma_k = (1, 1)$ for every time step $k \in \mathbb{S}$, and an associated optimal policy for the control input. We recall that such an optimal control input policy is given by

$$\mu^u_{all,k}(I_{k-1}) = L_k \hat{x}_{k|k-1} \tag{4}$$

where $\hat{x}_{k|k-1} = \mathbb{E}[x_k|I_{k-1}]$ can be obtained by a time varying Kalman filter and

$$L_k = -(R + B^T K_{k+1} B)^{-1} B^T K_{k+1} A$$

where K_{k+1} denotes the solution of the discrete-time dynamic Ricatti equation (DDRE)

$$\begin{aligned} K_N &= Q_N \\ K_k &= Q + A^T K_{k+1} A - P_k \\ P_k &= A^T K_{k+1} B (R + B^T K_{k+1} B)^{-1} B^T K_{k+1} A, \end{aligned} \tag{5}$$

for all $k \in \mathbb{S}$ The cost-to-go function of policy (4) denoted by π_{all} is

$$J_{\pi_{all}}(I_{k-1}) = \mathbb{E}\{x_k^T K_k x_k | I_{k-1}\} + \sum_{s=k+1}^{N} \mathrm{Tr}(K_s \Phi_v) + \sum_{s=k}^{N-1} \mathrm{Tr}(P_s \Sigma_s), \tag{6}$$

where $\Sigma_k = \mathrm{Cov}[x_k|I_{k-1}]$ denotes the conditional covariance matrix of the estimation error that can be expressed as

$$\Sigma_s = \mathrm{Ric}^{\,s}(\Theta_0), \quad s \in \mathbb{S} \tag{7}$$

where

$$\mathrm{Ric}^{\,1}(\Sigma) = A\Sigma A^T + \Phi_v - A\Sigma C^T (C\Sigma C^T + \Phi_r)^{-1} C\Sigma A^T$$

and $\mathrm{Ric}^s = \mathrm{Ric}^{s-1} \circ \mathrm{Ric}^1$, $\mathrm{Ric}^0 = \mathrm{id}$.

Remark 1: By setting the penalty Q_N for the final state x_N equal to $\overline{K}$, where $\overline{K}$ is the steady state solution of DDRE (5), the DDRE becomes the discrete time algebraic Ricatti equation (DARE) and the LQR-gain in (4) is time-invariant. Moreover, if the horizon is sufficiently large, the control policy for the first instances of time approaches this time-invariant policy corresponding to the infinite horizon solution of the discrete time algebraic Ricatti equation. Note that given (A, B) controllable and Q and R are positive definite, the solution of the DDRE converges to that of the DARE as $N \to \infty$.

III. Control policy

The control policy validated in this paper is parameterized by two non-negative scalars ζ, θ and defined by

$$(u_k, \gamma_k) = \begin{cases} (L_k \hat{x}_{k|k-1}, 1), & \text{if } \eta_k^T \Gamma_k \eta_k > \lambda_k \\ (\emptyset, 0), & \text{otherwise} \end{cases} \tag{8}$$

$$\beta_k = \begin{cases} 1, & \text{if } h_k(\Sigma_k) > \zeta \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where

$$\begin{aligned} \eta_k &= \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{u}_{k-1} \end{bmatrix} \\ \Gamma_k &= \begin{bmatrix} (1+\theta)P_k - \theta Q & (1+\theta)A^T K_k B \\ (1+\theta)B^T K_k A & R + (1+\theta)B^T K_k B \end{bmatrix} \\ \lambda_k &= \theta\, \mathrm{Tr}(Q\Sigma_k) \\ \hat{x}_{k|k-1} &= \mathbb{E}[x_k|I_{k-1}] \\ \Sigma_k &= \mathrm{Cov}[x_k|I_{k-1}] \end{aligned} \tag{10}$$

and

$$h_k(\Sigma_k) = \sum_{s=k+1}^{N} \mathrm{Tr}\Big(P_s\big(\mathrm{Ric}^{\,s}(A\Sigma_k A^T + \Phi_v) - \mathrm{Ric}^{\,s+1}(\Sigma_k)\big)\Big). \tag{11}$$

Theorem 1: Consider system (1) with policy π defined by (8)-(11). Then

$$J_\pi(I_{-1}) \le (1+\theta)(J_{\pi_{all}}(I_{-1}) + N\zeta), \tag{12}$$

for every I_{-1}. Let J_π refer to the cost (2) for the control policy π given by (8)-(11) and $J_{\pi_{all}}$ refer to the cost for the all-time transmission control policy given by (6).

Remark 2: Scalars θ and ζ can be used to balance the trade-off between guaranteed performance in terms of (2) and the number of transmissions. Clearly increasing ζ in (9) not only increases the guaranteed bound (12) but will also make the sensor query triggering condition (9) less stringent which results in less transmissions from sensors to the controller. A similar reasoning can be applied to the parameter θ for the controller-actuator network. Note that since $\Sigma \succeq 0$ and $h_k(\Sigma) \ge 0$, choosing $\zeta = 0$ results in a transmission in the sensor-controller network at every time step $k \in \mathbb{S}$. Moreover, note that for $\theta = 0$, $\Gamma_k \succeq 0$ $k \in \mathbb{S}$ and $\lambda_k = 0$, which results in a transmission in the controller-actuator network at every time step $k \in \mathbb{S}$. If $\zeta = 0$ and $\theta = 0$, we recover the all-time transmission control policy π_{all} and (12) holds with equality. Furthermore, the bound on the finite horizon cost does not only depend on the parameters θ and ζ but also on the length of the horizon N. Therefore, the horizon length can be seen as another tuning parameter to control the bound on the cost.

We show in the state estimate subsection of the current section that $\hat{x}_{k|k-1} = \mathbb{E}[x_k|I_{k-1}]$ and $\Sigma_k = \mathrm{Cov}[x_k|I_{k-1}]$ can be obtained by the controller by running the time-varying Kalman filter. As we shall see $\Sigma_k = \mathrm{Cov}[x_k|I_{k-1}]$ can be determined a priori, which entails that the scheduling sequence for sensor queries, triggered by condition (9) can be determined offline. In turn, the state estimate $\hat{x}_{k|k-1} = \mathbb{E}[x_k|I_{k-1}]$ depends on the noise realizations and therefore must be determined online. Consequently, the scheduling decisions, triggered by condition (8), must be determined by the controller online.

Remark 3: As mentioned in Remark 1, if we set $Q_N = \overline{K}$, where $\overline{K}$ is the steady state solution of DDRE (5), the matrices

Γ_k and scalars λ_k become time-invariant which makes the proposed policy time-invariant similar to the policy in [1].

State estimate

The conditional distribution of x_k given I_{k-1} is Gaussian (the proof of this fact can be concluded from a similar proof in [21]) and therefore $\hat{x}_{k|k-1} = \mathbb{E}[x_k|I_{k-1}]$ and $\Sigma_k = \mathrm{Cov}[x_k|I_{k-1}]$ can be obtained by running the time-varying Kalman filter

$$\begin{aligned} \hat{x}_{k+1|k} &= A\hat{x}_{k|k-1} + B\hat{u}_k + \beta_k G_k\big(y_k - C\hat{x}_{k|k-1}\big) \\ G_k &= A\Sigma_k C^T(C\Sigma_k C^T + \Phi_r)^{-1} \\ \Sigma_{k+1} &= \overline{\mathrm{Ric}}(\Sigma_k, \beta_k), \end{aligned} \tag{13}$$

where

$$\begin{aligned} \overline{\mathrm{Ric}}(\Sigma, j) = A\Sigma A^T + \Phi_v - \\ jA\Sigma C^T(C\Sigma C^T + \Phi_r)^{-1}C\Sigma A^T, \end{aligned} \tag{14}$$

β_k is determined via (9) and $\hat{u}_k$ is the input to the plant which is known to the controller

$$\hat{u}_k = \gamma_k L_k \hat{x}_{k|k-1} + (1-\gamma_k)\hat{u}_{k-1}. \tag{15}$$

Remark 4: Based on a similar reasoning to the one presented in [1], the scheduling sequence of sensor transmission $\{\beta_k\}_{k\in\mathbb{S}}$ can be determined offline. However, the scheduling decision for the network connecting controller to the actuator γ_k requires the knowledge of the state estimation $\hat{x}_{k|k-1}$ at each time step $k \in \mathbb{S}$ which itself depends on I_{k-1} and therefore should be computed online.

IV. Experimental set-up

The objective of the experiment is to control a robot in one direction (longitudinal). In particular we, we consider a regulator problem by which the position of the robot is driven towards the zero position. The set-up for the experiments is depicted in Fig.2. The set-up consists of a robot, a wired camera, a control unit which is the laptop and a wireless network. The camera is used to obtain an image of the robot in its field of view. The camera takes an image at designated time instances, which are defined by the controller, and sends the images to the controller. The controller receives the images from the camera and processes the images to get the estimated position of the robot with respect to a certain reference frame placed in the environment. The camera is initially calibrated based on given features of the environment with known positions with respect to this reference frame. Based on the estimated position of the robot, the controller is able to compute new control actions. These computed control actions are sent through a wireless network to the robot and executed. In the sequel the robot characteristics, the camera module, the controller and the communication network will be described.

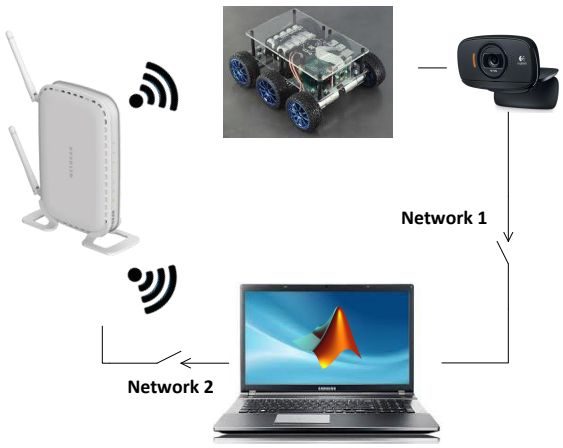

Fig. 2. A simple block diagram of the experimental.

A. Robot

The physical set-up we consider is a non-holonomic Diddy-borg robot with six wheels as depicted in Fig. 3. Each wheel of the robot is driven separately via a DC motor using a PWM-signal designed by the manufacturer. As the objective of the experiment is to control the robot in one dimension the same PWM-signal is used for all motors. Since all the motors are DC and the robot is only controlled in one direction the higher order dynamics of the robot will be neglected. The dynamics of the robot can be modeled by the following continuous-time first order linear system

$$\begin{aligned} \dot{x}(t) &= Bu(t) + v(t) \\ y &= x(t) + r(t), \end{aligned} \tag{16}$$

where $x \in \mathbb{R}$ and $y \in \mathbb{R}$ are the position and the position measurement of the robot with respect to the given reference frame, $u \in [-6V, 6V]$ denotes the voltage applied to the actuators and t denotes time. The state disturbance and measurement noise are represented by v and r respectively.

Remark 5: The first order system model has been identified using the Matlab identification toolbox, this lead to $B = 0.0023$. Given the sampling time T_s the discrete-time model of the system (16) can be described as

$$\begin{aligned} x_{k+1} &= x_k + T_s B u_k + v_k \\ y_k &= x_k + r_k, \end{aligned} \tag{17}$$

with $v_k, \in \mathcal{N}(0, \sigma_k^2)$ and $r_k, \in \mathcal{N}(0, \sigma_r^2)$ are assumed to be Gaussian zero-mean. The values of σ_k^2 and σ_r^2 are obtained through experiments.

B. Vision system and control unit

For the vision based system the Logitech HD Webcam C525 is used, which is initially calibrated with the help of calibration cubes using the homogeneous transformation approach [22]. A pink marker is placed on the robot. In Fig.4 an HSV image of the robot is shown. On the control unit a Matlab program is running to estimate the position of the robot based on the

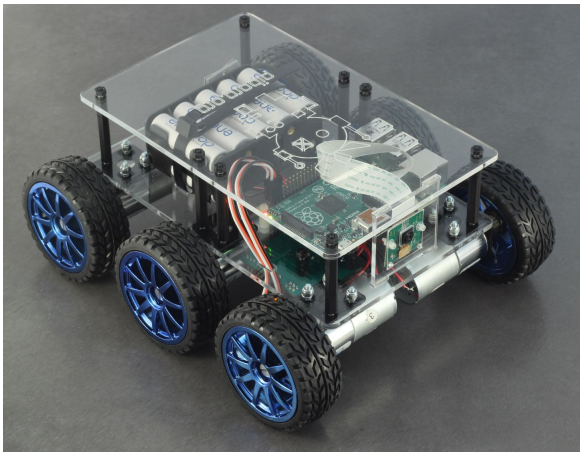

Fig. 3. The Diddyborg robot used for the experiment.

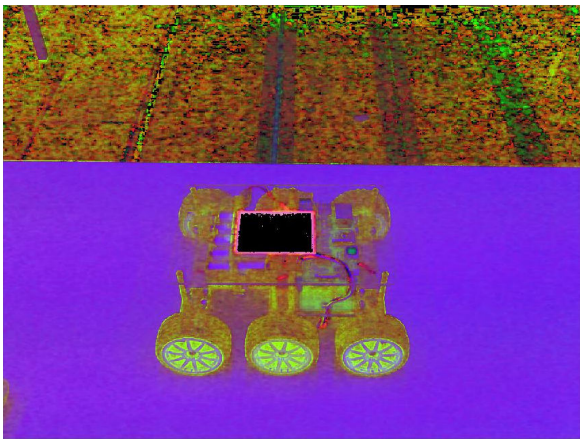

Fig. 4. The pink marker detection in the HSV image.

image of the camera by the Perspective and Point approach (PnP) [22].

Remark 6: The detection algorithm requires computational power and is time consuming. This justifies the use of the discrete-time model of the plant and imposes a lower bound on the sampling time, $T_s > 0.2$ seconds. Also note that this detection phase needs computational resources and therefore it is preferable to reduce the usage of the algorithm to compute the estimate of the state. This further emphasizes the need of a resource allocation algorithm.

C. Wireless communication

The communication between the robot and the controller is established through a wireless network provided by a hot spot via a router. In order to facilitate the transmission of the control input we use the unified datagram protocol (UDP).

Remark 7: UDP is a lightweight simple protocol. We choose this protocol because of its simplicity and its fast execution time. During the experiments we did not encounter any problems regarding packet drops or transmission delays. In fact the delays are negligible w.r.t. our sampling time. So for our experiment the UDP protocol is sufficient.

D. Experiment

In the experiment the motion system is controlled in one direction (longitudinal). The robot is regulated to the zero position, from an approximate initial position of 0.7 m. A horizon of $N = 60$ time steps is considered since for both control policies this horizon is sufficiently large to regulate the robot to the zero position. The horizon of $N = 60$ time steps corresponds to 18 seconds since the sampling time T_s is set to 0.3 seconds. The values of the weighting matrices Q, Q_N and R are tuned to obtain a smooth performance in robot movement.

$$Q = 10 \quad Q_N = 200, \quad R = 0.002.$$

We compare two cases, one with $\theta = \zeta = 0$ which corresponds to an all-time transmission policy, π_{all}, and the other with $\theta = 0.5$ and $\zeta = 0.004$ implementing the policy, π, (8)-(9). The state estimation and related covariances are computed using a time-varying Kalman filter, see (13).

V. Numerical results

In this section, we present the results of numerical simulations based on the obtained model (17). By applying the ETC scheme, the sensor and actuator network usage are reduced by 80% and 90.39% respectively. Fig. 5 shows the estimated running cost $\mathbb{E}[x_k^T Q x_k + \hat{u}_k^T R \hat{u}_k]$ based on Monte Carlo simulations for both cases. The covariances of the state disturbance and measurement noise are set to σ_k^2 and σ_r^2, see Section IV-A. The cost of the ETC policy obtained via Monte Carlo simulations satisfies the performance bound (12)

$$113.15 = J_\pi^{sim} \leq 1.5(J_{\pi_{all}}^{sim} + 60 \cdot 0.004) = 158.85.$$

where $J_{\pi_{all}}^{sim} = 103.66$. Notice that the cost of the ETC policy is much less than the theoretical performance bound and close to the cost of the all-time transmission policy $J_{\pi_{all}}^{sim}$. This means that the bound is conservative. To see the reduction in communications, the actuator signals of one realization of both cases with the same noise is shown in Fig. 6, the value of γ_k for the ETC policy is also shown. As can be seen, the actuator holds its value more often (i.e. less transmissions occur) when applying the ETC method. The evolution of the trajectory $h_k(\Sigma_k)$ is shown in Fig.7. At the time instances where $h_k(\Sigma_k)$ exceeds the threshold ζ, the control unit acquires a new image from the camera through the sensor network. The evolution of the trajectory of $h_k(\Sigma_k)$ exhibits , once every five time steps $h_k(\Sigma_k)$ exceeds ζ and β_k becomes 1.

VI. Experimental results

In this section, we present the experimental results. The settings for the experiments are already discussed in Section IV-D. For both control policies the experiment is carried out ten times with similar initial conditions. Since the settings for the experiment are the same as for the numerical simulations, the behavior of the sensor network is the same for both the simulations and the experiment. The a priori calculated evaluation of $h_k(\Sigma_k)$ for the ETC policy is shown in Fig. 7. The communication in the sensor network is reduced by 80%

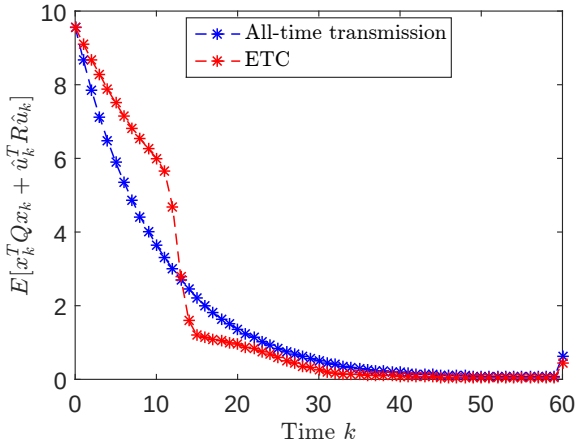

Fig. 5. $\mathbb{E}[x_k^T Q x_k + \hat{u}_k^T R \hat{u}_k]$ estimated via 100 Monte Carlo simulations in two cases $\theta = \zeta = 0$ and $\theta = 0.5$, $\zeta = 0.004$.

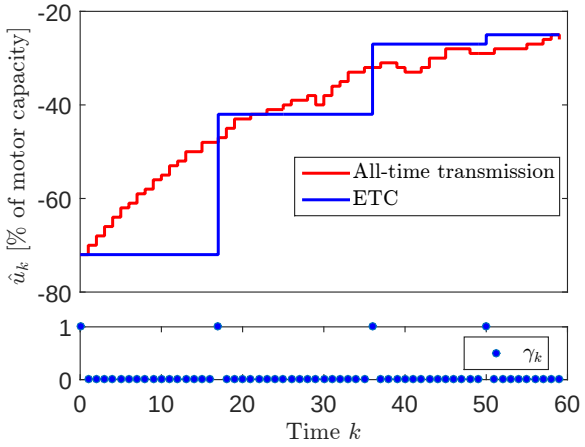

Fig. 6. Trajectory of actuation signal, $\hat{u}_k$, for two cases with $\theta = \zeta = 0$ and $\theta = 0.5, \zeta = 0.004$ obtained by simulation. For $\gamma_k = 1$ a new control input $\hat{u}_k$ for the ETC policy is computed.

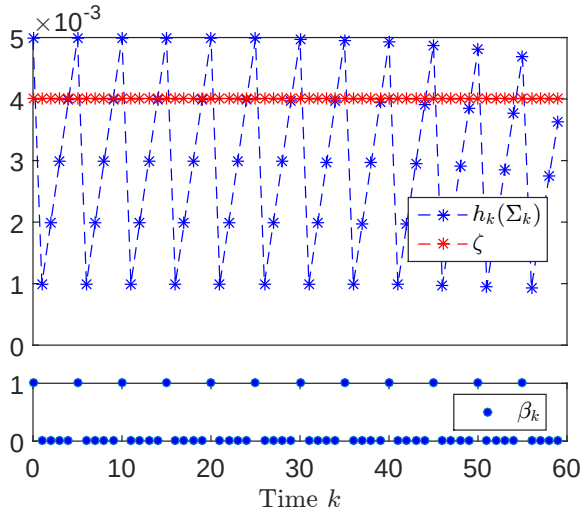

Fig. 7. The offline computed evaluation of $h_k(\Sigma_k)$ for the case $\theta = 0.5$ and $\zeta = 0.004$. For $\beta_k = 1$ a new image is sent from the camera over the network to the control unit.

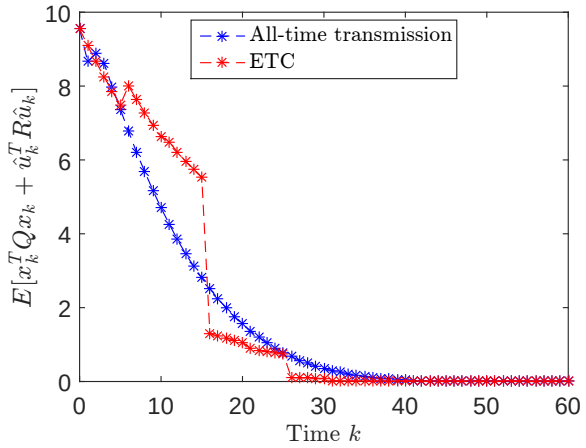

Fig. 8. The experimental expected running cost, $\mathbb{E}[x_k^T Q x_k + \hat{u}_k^T R \hat{u}_k]$, for two cases with $\theta = \zeta = 0$ and $\theta = 0.5, \zeta = 0.004$, over ten experiments.

compared to the all-time transmission control policy. For the experiments carried out the communication in the actuator network is on average reduced by 89.80%. In Fig. 8 the average running cost $\hat{x}_k^T Q \hat{x}_k + \hat{u}_k^T R \hat{u}_k$ for both the all-time transmission and the ETC policy are shown (the average is taken over all ten experiments)f. The total expected cost for the ETC policy satisfies the performance bound on the cost (12)

$$127.85 = J_\pi^{exp} \leq 1.5(J_{\pi_{all}}^{exp} + 60 \cdot 0.004) = 175.06.$$

where $J_{\pi_{all}}^{exp} = 116.47$ is the expected cost for running the all-time transmission policy. Notice that the cost of the ETC method is much less than the performance bound and close to the cost of the all-time transmission policy $J_{\pi_{all}}^{exp}$. Again we see that the bound is conservative. In Fig. 9 the control inputs $\hat{u}_k$ for both the periodic control and the ETC policy are shown for a single experiment, also the value of γ_k for the ETC policy is shown. For the event-triggered control policy, a control input is only sent five times, this is clearly visible in Fig. 9. This result shows the advantage of the applied event-triggered control policy, only if necessary the actuator network is used and otherwise the resources are saved. In Fig. 10 the estimated position of the robot, $\hat{x}_k$ is shown for both the periodic control policy and the ETC policy. In the considered horizon of $N = 60$ time steps the robot is for both policies regulated from its initial position to the zero position. The estimated state for the ETC policy, in Fig. 10, shows a jump every time $\beta_k = 1$ when the state is not yet regulated to zero. The cause of this jump is a mismatch between the model of the robot and the actual dynamics of the robot. The estimated state for the all-time transmission policy does not have such large jumps since the estimated state is updated each time step with a new image from the camera.

VII. Numerical versus experimental results

In this section the numerical results will be compared with the experimental results. Some interesting observations can be pointed out. The expected cost obtained by Monte Carlo simulations is relatively close to the expected cost obtained

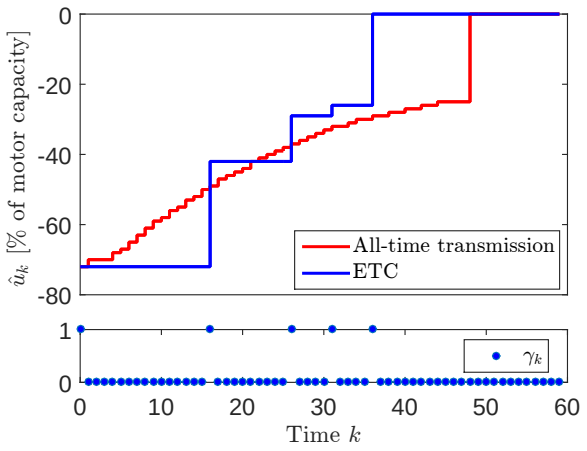

Fig. 9. Trajectory of the actuation signal in the experimental setup, $\hat{u}_k$, for a single experiment for two cases with $\theta = \zeta = 0$ and $\theta = 0.5, \zeta = 0.004$. For $\gamma_k = 1$ a new control input $\hat{u}_k$ is computed.

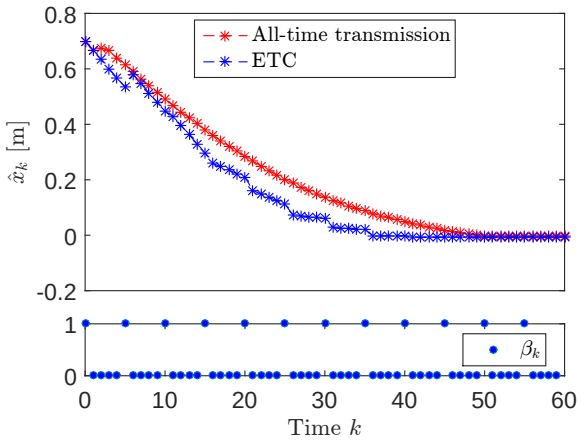

Fig. 10. The estimated position, $\hat{x}_k$, of the real motion system for two cases with $\theta = \zeta = 0$ and $\theta = 0.5, \zeta = 0.004$.

in the experiments for both control policies, the costs in the simulations are slightly lower. In the control policy the disturbances are assumed to be Gaussian zero-mean, this does not perfectly reflect the reality. Which can cause a slightly higher cost in the experiments compared to the cost in the simulations. The behavior of the expected running cost is comparable for the simulations and the experiments, see Fig. 5 and Fig. 8. In the expected running cost for the ETC policy for the experiments, Fig. 8, there is a jump visible at $k = 5$. This jump is due to a mismatch between the model of the robot and the actual dynamics of the robot, as already explained in end of Section VI.

The network utilization shows similar results for the simulations and the experiments. The behavior of the sensor network is computed offline and therefore the same for both cases. However in the actuator network the scheduling mechanism is online and interestingly we see similar results. In the simulations the communication is reduced by 90.3% reduction and in the experiments the communication is reduced by 89.8%. This further validates the effectiveness of the proposed control policy.

VIII. Conclusion

In this paper we validated an event-triggered control strategy for linear discrete-time systems derived in the recent work [1]. The result of this recent work is a control strategy with performance guarantees compared to the all-time transmission control policy, while reducing the overall communication load. The performance of the control strategy is measured by a quadratic cost. In the previous work the cost was defined for infinite horizon whereas in this paper a finite horizon cost is considered, which is a better representation of the actual experiment. To validate the control policy, the policy is implemented on a real motion system. The experiment validated that the proposed event-triggered control policy significantly reduces the communication in the sensor and the actuator network while guaranteeing performance bounds on the cost. From the validation we have learned that for a one dimensional movement, communication can be reduced by 80% and 90% for the sensor network and the actuator network respectively, compared with the all-time transmission control policy. Small differences between the simulations and the experiments can be explained by the fact that in the control policy the disturbances are assumed to be Gaussian zero-mean which does not perfectly reflect the reality.

The validated policy can not deal with packet drops and transmission delays, therefore these are neglected in the validation. An interesting observation of the policy is that the guarantees concerning the performance bound are conservative since the cost for the ETC policy is relatively close to the cost for the all-time transmission policy. Future steps will be to extend the event-triggered policy towards general independent and identically distributed noise including network imperfections such as packet drops and transmission delays. Another future step is to implement the event-triggered policy on a more challenging robot for event-triggered path planning.

References

[1] B. A. Khashooei, D. J. Antunes, and W. P. M. H. Heemels, "An event-triggered policy for remote sensing and control with performance guarantees," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 4830–4835.

[2] J. K. Yook, D. M. Tilbury, and N. R. Soparkar, "Trading computation for bandwidth: reducing communication in distributed control systems using state estimators," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 4, pp. 503–518, Jul 2002.

[3] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *Automatic Control, IEEE Transactions on*, vol. 52, no. 9, pp. 1680–1685, Sept 2007.

[4] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, "Periodic event-triggered control for linear systems," *Automatic Control, IEEE Transactions on*, vol. 58, no. 4, pp. 847–861, April 2013.

[5] D. Lehmann and J. Lunze, "Event-based output-feedback control," in *Control Automation (MED), 2011 19th Mediterranean Conference on*, June 2011, pp. 982–987.

[6] K. J. Astrom and B. M. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in *Decision and Control (CDC), 2002 41st IEEE Conference on*, vol. 2, dec. 2002, pp. 2011 – 2016 vol.2.

[7] K.-E. Arzén, "A simple event-based pid controller," 1999.

[8] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Event-triggered control for discrete-time systems," in *American Control Conference (ACC), 2010*, June 2010, pp. 4719–4724.

[9] M. Abdelrahim, R. Postoyan, J. Daafouz, and D. Nesic, "Stabilization of nonlinear systems using event-triggered output feedback controllers," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2015.

[10] A. Girard, "Dynamic triggering mechanisms for event-triggered control," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1992–1997, July 2015.

[11] A. Molin and S. Hirche, "Structural characterization of optimal event-based controllers for linear stochastic systems," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, Dec 2010, pp. 3227–3233.

[12] C. Ramesh, H. Sandberg, L. Bao, and K. H. Johansson, "On the dual effect in state-based scheduling of networked control systems," in *American Control Conference (ACC), 2011*, June 2011, pp. 2216–2221.

[13] W. Wu, S. Reimann, D. Gorges, and S. Liu, "Suboptimal event-triggered control for time-delayed linear systems," *Automatic Control, IEEE Transactions on*, vol. 60, no. 5, pp. 1386–1391, May 2015.

[14] J. Araujo, A. Teixeira, E. Henriksson, and K. H. Johansson, "A down-sampled controller to reduce network usage with guaranteed closed-loop performance," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, Dec 2014, pp. 6849–6856.

[15] R. Cogill, S. Lall, and J. P. Hespanha, "A constant factor approximation algorithm for event-based sampling," in *American Control Conference*, july 2007, pp. 305 –311.

[16] J. Araujo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson, "System architectures, protocols and algorithms for aperiodic wireless control systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 175–184, Feb 2014.

[17] T. Blevins, M. Nixon, and W. Wojsznis, "Event based control applied to wireless throttling valves," in *Event-based Control, Communication, and Signal Processing (EBCCSP), 2015 International Conference on*, June 2015, pp. 1–6.

[18] J. Snchez, M. n. Guarnes, and S. Dormido, "On the application of different event-based sampling strategies to the control of a simple industrial process," *Sensors*, vol. 9, no. 9, p. 6795, 2009. [Online]. Available: http://www.mdpi.com/1424-8220/9/9/6795

[19] B. Boisseau, S. Durand, J. J. Martinez-Molina, T. Raharijaona, and N. Marchand, "Attitude control of a gyroscope actuator using event-based discrete-time approach," in *Event-based Control, Communication, and Signal Processing (EBCCSP), 2015 International Conference on*, June 2015, pp. 1–6.

[20] M. Sigurani, C. Stcker, L. Grne, and J. Lunze, "Experimental evaluation of two complementary decentralized event-based control methods," *Control Engineering Practice*, vol. 35, pp. 22 – 34, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066114002378

[21] W. Wu and A. Arapostathis, "Optimal sensor querying: General markovian and LQG models with controlled observations," *Automatic Control, IEEE Transactions on*, vol. 53, no. 6, pp. 1392–1405, July 2008.

[22] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer, 2011, vol. 73.