Non-uniform Multi-rate Observer Based Event-Triggered Control: Computation Saving

Angel Cuenca, Minghui Zheng, Masayoshi Tomizuka

I. INTRODUCTION AND STRATEGY

As observed in [1], if we focus on increasing computation saving by means of Periodic Event-Triggeed Sampling (PETS) strategies, control performance can be worsened. The consideration of a non-uniform multi-rate estimator (similar to the one used in [2]) can enable to improve control performance so as to reach the desired one (i.e. that obtained by the Time-Triggered Sampling -TTS- strategy) while maintaining computation saving. The main reasons of using this kind of estimator are:

- measurements can arrive at the controller at slow rate following a non-uniform pattern (due to PETS).
- the controller is required to generate a fast-rate control signal in order to reach the desired control performance.

The non-uniform multi-rate estimator can compute $h$-step ahead state predictions considering a time-varying gain. These are the main steps of the proposal:

- When the event is triggered at the sensor device, a new output $y_k$ is received by the controller. Then, the controller calculates the current multiplicity $N$ as the amount of (fast) control periods between the last received output $\bar{y}_{k-1}$ and the new one $y_k$. From the time-varying $N$, the time-varying observer gain $K$ is computed.
- From $y_k$ and $K$, the correction of the estimator is carried out yielding the estimate of the state $\hat{x}_k$.
- From $\hat{x}_k$, an estimated output $\hat{y}_k$ can be obtained, and then, a new (estimated) control action $\hat{u}_k$.
- From $\hat{u}_k$ and the system model, the next $\hat{x}_{k+1}, \hat{y}_{k+1}, \hat{u}_{k+1}$ can be computed, and so on up to $h$-step ahead predictions. Therefore, a set of future, estimated control actions $\hat{u}_{k+1}, ..., \hat{u}_{k+h}$ are obtained following this cascade prediction. Note, the amount of computation at the controller in each iteration could be increased due to the prediction stage.
- If the event is triggered at the controller, the current control action $\hat{u}_k$ and the $h$ future ones $\hat{u}_{k+1}, ..., \hat{u}_{k+h}$ are sent to the actuator in a packet. Therefore, the actuator injects the current control action $\hat{u}_k$ at instant $k$, and the future ones $\hat{u}_{k+1}, ..., \hat{u}_{k+h}$ at instants $k+1, ..., k+h$ while no new control action is received for the $h$ future instants. Note, this fact implies the
actuator is working every time (its utilization is 100%, as in the TTS strategy), and hence, no possible energy saving could be achieved at this device.

To complete the study, nominal vibration dynamics beyond the Nyquist frequency and sensor noise can be added to the HDD model such as in [2]. The non-uniform multi-rate observer should be able to reduce measurement noise effects (filtering task) and generate vibration estimates which can be used by the multi-rate controller with interlacing to compensate for the actual vibrations (prediction task). This estimation will be followed when the dynamic model of the disturbance is known. Another option to be explored is the inclusion of a Disturbance Observer (DOB) to estimate the disturbance when no dynamic model is known.

II. MATHEMATIC DETAILS

The fast model of the plant is

\[
x_p(i + 1) = A_p x_p(i) + B_p u(i) + B_p d(i) (i = 1, 2, 3, \ldots)
\]
\[
y(j) = C_p x_p(j) + v(j) (j = N, 2N, \ldots)
\]

The fast model of the disturbance is assumed as

\[
x_d(i + 1) = A_d x_d(i) + B_d w(i)
\]
\[
d(i) = C_d x_d(i)
\]

Therefore, the augmented system is

\[
\begin{bmatrix}
  x_p(i + 1) \\
  x_d(i + 1)
\end{bmatrix}
= \begin{bmatrix}
  A_p & B_p C_d \\
  0 & A_d
\end{bmatrix}
\begin{bmatrix}
  x_p(i) \\
  x_d(i)
\end{bmatrix}
+ \begin{bmatrix}
  B_p \\
  0
\end{bmatrix} u(i)
+ \begin{bmatrix}
  0 \\
  B_d
\end{bmatrix} w(i)
\]
\[
y(j) = \begin{bmatrix}
  C_p & 0
\end{bmatrix}
\begin{bmatrix}
  x_p(j) \\
  x_d(j)
\end{bmatrix}
+ v(j)
\]
\[
d(i) = C_d x_d(i)
\]

which is further written in a compact format

\[
x(i + 1) = Ax(i) + Bu(i) + B_u w(i)
\]
\[
y(j) = Cx(j) + v(j)
\]

The event-triggered policies are used at both the sensor device and the control device. The event-triggered policy at the sensor device is

\[
\|\bar{y}_k - y_k\|^2 > \sigma_s \|y_k\|^2 + \delta_s
\]

where $\bar{y}_k$ is the latest output value transmitted to the controller, $y_k$ is the current sampled output, $\sigma_s$ and $\delta_s$ are positive constants. The idea of this kind of sampling strategy is: when the measurement is changing fast, a new control action is calculated; when the measurement
is changing slowly, no control action is calculated (computation saving) and no packet is sent (energy saving).

Similarly, the event-triggered policy at the controller device is
\[ \| \bar{u}_{k-1} - u_k \|^2 > \sigma_c \| u_k \|^2 + \delta_c \]  
where \( \bar{u}_{k-1} \) is the latest control action transmitted to the actuator, \( u_k \) is the current control action, \( \sigma_c \) and \( \delta_c \) are positive constants.

Based on the event-triggered policy, the non-uniform multi-rate observer includes the prediction and the correction stages. The mathematic details are provided as follows:

1. Prediction \((1 \leq m \leq N)\).
   \[
   \hat{x}(Nk + m|Nk) = A^m \hat{x}(Nk|Nk) + \sum_{c=0}^{m-1} A^{m-1-c} Bu(kN + c)
   \]  

2. Correction.
   \[
   \hat{x}(Nk + N|Nk + N) = \hat{x}(Nk + N|Nk) + K(N)[y(Nk + N) - C\hat{x}(Nk + N|Nk)]
   \]  

where \( N \) is varying. Therefore, the observer gain is dependent on \( N \) and varying, and it can be obtained from the following equations:

\[
K(N) = MC^T[CMC^T + V]^{-1};
\]
\[
M = A^N M(A^N)^T + W_e - A^N M C^T [CMC^T + V]^{-1} CM (A^N)^T
\]  

where \( V \) and \( W_e \) are the covariances of the noise \( v \) and \( w_e \). That is, \( V = Cov(v) \) and

\[
W_e = Cov\left( \sum_{c=0}^{N-1} A^{N-1-c} B_w w(kN + c) \right)
\]
\[
= E\left\{ \left[ \sum_{c=0}^{N-1} A^{N-1-c} B_w w(kN + c) \right] \left[ \sum_{c=0}^{N-1} A^{N-1-c} B_w w(kN + c) \right]^T \right\}
\]
\[
= \left( \sum_{c=0}^{N-1} A^{N-1-c} B_w \right) W_e \left( \sum_{c=0}^{N-1} A^{N-1-c} B_w \right)^T
\]  

The overall control scheme is provided in Figure 1.
III. SIMULATION RESULTS VIA TRUETIME

A. Case 1: PETS scenario (wider thresholds) with no noise, no disturbance, but estimator

In this first case, the non-uniform multi-rate estimator has been implemented in a PETS scenario reaching the expected results, that is, the control system is able to achieve the desired performance despite decreasing controller utilization (i.e. increasing computation saving). The study uses the optimal values detected for the thresholds in [1] when giving preference to computation saving, that is, $\delta_s = \sigma_s = 1 \cdot 10^{-9}$ and $\delta_c = \sigma_c = 1.33 \cdot 10^{-2}$. Let us name these thresholds as the wider thresholds.

Figure 2 shows output, control actions, and device utilization for this study, where neither sensor noise nor vibration disturbance are considered. Under this assumption, the number of packets sent from sensor to controller is $N_{sc}^{PETS}=111$, and hence, the controller’s utilization is $U_c=0.24$ (remember that, due to the interlacing technique, each controller usage represents 66% of computation, then the total amount of computation will be 16%). This amount of computation is clearly lower than that obtained when the non-uniform multi-rate observer was not used in [1] (49.2%). Although the number of packets sent from controller to actuator is also lower (now $N_{ca}^{PETS}=46$, and in [1] $N_{ca}^{PETS}=109$, which represents 58% of reduction), remember that the actuator’s utilization is now 100%, since it is periodically working in order to inject not only the current control actions but also the future ones. The iterations in which the actuator injects current control actions are marked by a point at the bottom subplot in Figure 2. Despite the low amount of sent packets and controller’s utilization, the control system is able to achieve the desired performance (which was presented in Figure 2 in [1]).
was not used in [1], the control performance worsened according to $NE=32.5\%$.

B. Case 2: TTS scenario with noise, disturbance (knowing its dynamic model), and estimator. Desired performance

The main aim of this study is establishing the desired control performance when noise and disturbance appear, and the estimator is included in the control system. Figure 3 shows output, control actions, and device utilization. Figure 4 depicts a zoom to better observe the system output and control signal. Figure 5 (top subplot) illustrates a comparison between actual and estimated disturbances, being both of them very similar. The middle subplot presents the instantaneous
value for the multiplicity $N$, that is, the number of control actions injected by the actuator between two consecutive measurements received by the controller. The bottom subplot shows the distribution of $N$. As expected, $N$ is always equal to 1 in a TTS scenario, that is, all the sensed outputs are used by the controller (for this reason, actual and estimated outputs are identical in Figure 4). Finally, Figure 6 shows a zoom of the output and control signal obtained for the same TTS scenario (with the same noise and disturbance) but when no estimator is used. Compared to Figure 4, some degradation of the output is observed due to not considering the estimation of the disturbance signal when computing the control action.
Fig. 4. TTS: Zoom for output and control signal

Fig. 5. TTS: Actual and estimated vibration disturbances. Multiplicity $N$
C. Case 3: PETS scenario (wider thresholds) with noise, disturbance (knowing its dynamic model), and estimator

In this third study (Figure 7), both sensor noise and vibration disturbance are considered in a PETS scenario with the wider thresholds. Under these assumptions, compared to the case 2 (the TTS strategy, where the desired performance is established), the number of packets sent from sensor to controller $N_{sc}^{\text{PETS}}=211$ and from controller to actuator $N_{ca}^{\text{PETS}}=74$ is decreased in around 55% and 85%, respectively. The controller’s utilization is now $U_c=0.45$, which approximately implies 30% of total amount of computation. This amount is 36% lower than that obtained in the case 2. However, the behavior of the estimated output is worsened according to $NE=8.55\%$.

Figure 8 shows a zoom of the steady-state response to better compare the actual and estimated outputs. As depicted, both outputs differ in some instants. This fact can be explained observing the control signal, which is negatively affected by a bad estimation of the vibration disturbance. As shown in Figure 9, the estimates of the vibration disturbance are not accurate enough so as to compensate for the actual disturbance signal, negatively affecting the system output as well. The reason is the consideration of wider thresholds in the event-triggered conditions, which implies higher values for $N$, mainly at the steady-state response (as depicted in the middle plot in Figure 9), and then, worse estimations. Note, $N=0$ means no event is triggered at the sensor device.
D. Case 4: PETS scenario (narrower thresholds) with noise, disturbance (knowing its dynamic model), and estimator

In this fourth study (Figure 10), both sensor noise and vibration disturbance are considered in a PETS scenario with narrower thresholds (concretely, $\delta_s = \sigma_s = 1 \cdot 10^{-11}$ and $\delta_c = \sigma_c = 1 \cdot 10^{-4}$). As expected, compared to the case 3, now the number of packets sent from sensor to controller $N_{PETS}^{PETS} = 351$ and from controller to actuator $N_{CA}^{PETS} = 247$ is increased, but compared to the TTS scenario (case 2), these results represent around 16% of computation saving (now the controller’s utilization is $U_c \approx 0.75$, which approximately implies 50% of total amount of...
Fig. 8. PETS (wider thresholds): Zoom for output and control signal

Fig. 9. PETS (wider thresholds): Actual and estimated vibration disturbances. Multiplicity $N$
Fig. 10. PETS results (narrower thresholds) using non-uniform multi-rate estimator with noise and disturbance computation) and 50% of reduction of packets sent from controller to actuator. These figures are achieved while practically maintaining the desired performance ($NE$ is only worsened around 1%). Figure 11 shows a zoom of the steady-state response to better compare the actual and estimated outputs, observing that both outputs are very similar. The control signal also seems to be very similar to that presented for the TTS strategy in Figure 4. Now, as depicted in Figure 12, the vibration disturbance is accurately estimated, since narrower thresholds are considered in the event-triggered conditions, and hence lower values for $N$ are achieved. Finally, Figure 13 shows a zoom to better compare the consequent distributions obtained for $N$ in both PETS cases.
Fig. 11. PETS (narrower thresholds): Zoom for output and control signal

Fig. 12. PETS (narrower thresholds): Actual and estimated vibration disturbances. Multiplicity $N$
Fig. 13. PETS: Zoom for the distributions of $N$ (comparison from $N=3$)
IV. EXTENSION: WHEN DISTURBANCE DYNAMICS UNKNOWN - DISTURBANCE OBSERVER

Remarks: This section will be rewritten carefully with a better DOB design if necessary. A diffident disturbance file could be generated for further evaluation.

Figure 14 shows a general design structure of the disturbance observer. $G(z)$ is the plant; $C(z)$ is the baseline feedback controller; $G_n(z)$ is the nominal model of the plant; $Q(z)$ is a filter to maintain the causality of the DOB. The reference signal $r$, the output signal $y$, the control signal $u$, the disturbance $d$ and the estimated disturbance $\hat{d}$ are defined in Figure 14. The disturbance observer is the system in the red box, whose input signals are $u$ and $y$, and output signal is $\hat{d}$.

A. Mathematics

![Fig. 14. DOB Basics](image)

The basic idea of DOB is use the inverse of the plant $G_n^{-1}(z)$ to reconstruct the actual disturbance $d$. It is easy to obtain the dynamics from $d$ to $\hat{d}$:

$$\hat{d} = \frac{Q[G_n^{-1}G + GC][1 + GC]^{-1}}{(1 - Q) + Q[G_n^{-1}G + GC][1 + GC]^{-1}}d$$

where $z$ is omitted. It is worthing noting that if $Q = 1$ and $G_n^{-1}G = 1$, then $d = \hat{d}$, and the disturbance can be fully reconstructed from such a DOB. More design details will be included.

B. Simulation results via Truetime

1) Case 5: TTS scenario with noise, disturbance (no dynamic model), and both estimators:

Now, the dynamic model of the disturbance is not known. Therefore, a DOB is added. The control system presents similar results to those obtained in III-B regarding output response and control signal (remember, for brevity, in Figures 3 and 4). However, the disturbance estimation behaves worse at the transient response (Figure 15).
2) Case 6: PETS scenario (wider thresholds) with noise, disturbance (no dynamic model), and both estimators: In this study (Figure 16), the PETS scenario with the wider thresholds is considered, but now with no dynamic model for the disturbance (DOB is added).

Under these assumptions, compared to the case in III-C (when the dynamic model is known), a little higher computation saving is got ($U_c$ is around 1.3% lower) but sending a few more packets from controller to actuator ($N_{PETS}^{PETX}$ is around 8% higher). In addition, a little worse performance is obtained ($NE$ is around 3% higher).

As in III-C, estimated and actual outputs differ in some instants (Figure 17) due to a bad estimation of the vibration disturbance, which negatively affects the control signal (Figure 18). This signal shows sharper peaks with regard to those observed in Figure 8. This fact is related with high values for $N$.

3) Case 7: PETS scenario (narrower thresholds) with noise, disturbance (no dynamic model), and both estimators: This last study (Figure 19) considers the PETS scenario with narrower thresholds and no dynamic model for the disturbance, and hence, DOB is required.

Under these assumptions, compared to the case in III-D (when the dynamic model is known), a little higher computation saving is got ($U_c$ is around 0.5% lower) but sending a few more packets from controller to actuator ($N_{PETS}^{PETX}$ is around 2% higher). In addition, a little better performance is obtained ($NE$ is around 0.5% lower; hence, the desired performance is practically reached).

As in III-D, estimated and actual outputs are very similar (Figure 20) due to a better estimation of the vibration disturbance (Figure 21). However, control action continues presenting sharper peaks with regard to those observed in Figure 11 (although they are softer than in the previous
case with wider thresholds; $N$ is also lower). Finally, Figure 22 shows a zoom to better compare the consequent distributions obtained for $N$ in both PETS cases. The main differences with regard to those obtained in Figure 13 (when the disturbance model is known) are: 1) higher $N$ are reached when wider thresholds are used, 2) the distributions seem to follow an exponential fashion.
Fig. 17. PETS (wider thresholds): Zoom for output and control signal

Fig. 18. PETS (wider thresholds): Actual and estimated vibration disturbances. Multiplicity $N$
Fig. 19. PETS results (narrower thresholds) using both estimators with noise and disturbance.
PETS: $\sigma_s = \delta_s = 1 \cdot 10^{-11}$, $\sigma_c = \delta_c = 1 \cdot 10^{-04}$; NE=0.5%

Fig. 20. PETS (narrower thresholds): Zoom for output and control signal

Fig. 21. PETS (narrower thresholds): Actual and estimated vibration disturbances. Multiplicity $N$
Fig. 22. PETS: Zoom for the distributions of $N$ (comparison from $N=3$)
V. CONCLUSIONS AND FUTURE WORKS

The main benefits of using PETS strategies together with non-uniform multi-rate estimation techniques are: 1) reaching higher computation saving and lower packet traffic, 2) maintaining control performance at about the same level as the TTS solution. As analyzed, some trade-off between performance and saving can be taken into account by conveniently tuning the event-triggered condition thresholds.

Another interesting conclusion is the achievement of similar results when the requirement of known disturbance dynamic model is released, and then, DOB is added to estimate the vibration disturbance.

The proposal might be relevant in rehabilitation systems from two different points of view:

- computation saving: the limited computation power of the mobile controllers used in rehabilitation systems could be smoothed.
- energy saving: just to avoid the previous problem, Wireless Networked Control Systems (WNCS) can be used [3]–[5] (and literature therein). Sending less amount of sensor-to-controller and controller-to-actuator packets can result in considerable reductions in the overall energy usage of battery-powered wireless devices, thereby having positive effects on their battery lives. In addition, as the network load is decreased, more devices could be connected to the network (wider bandwidth is available).

From the second point of view, the proposed control solution could be used in a WNCS because:

- packet dropouts can be treated by the controller in the same way as when event conditions are not satisfied. On one hand, if the sensor condition is not triggered or the output is lost through the sensor-to-controller link, then an estimate of the output can be calculated. On the other hand, if the controller condition is not triggered or the action is lost through the controller-to-actuator link, then an estimated, future control action can be injected (sent in a previous packet).
- the computation of the control signal is delay-free. The set of estimated, future control actions is periodically applied with independence from the (time-varying) delays. If a new delayed set of actions is received by the actuator, the action selected from this set to be applied should be very similar to the one applied from the beginning of the current control period, since the latter is an estimate of the former (from the point of view that the former is computed after correcting the estimated state). Figure 23 tries to illustrate this situation.

For the sake of simplicity only controller and actuator are represented. The set of actions computed in $k$ does not arrive at the actuator, and that one computed in $k + 1$ arrives with a delay $\tau_{k+1}$, which is greater than 2 control periods $T$. At the beginning of $(k + 3)T$, $\hat{u}_{k+3/k-1}$ is injected, that is, $\hat{u}_{k+3}$ calculated in $k - 1$ (the packet computed in $k - 1$ does arrive at the actuator with a delay $\tau_{k-1} < T$). When the new set of actions arrives at the
actuator in $(k + 3)T + \tau_{k+1}$, the suitable action to be applied can be selected from the received packet. In this case, this action is $\hat{u}_{k+3/k+1}$, where $\hat{u}_{k+3/k+1} \simeq \hat{u}_{k+3/k-1}$.

- packet disorder can appear due to the fact that the time-varying delay could be greater than one control period. Implementing time-stamping techniques, out-of-order packets could be detected and discarded (that is, treated as packet dropouts).

REFERENCES


