Time Delay Compensation in a Wireless Tracking Control System with Previewed Reference

Wenlong Zhang, Masayoshi Tomizuka, Yi-Hung Wei, Quan Leng, Song Han, and Aloysius K. Mok

Abstract—In this paper, a wireless tracking control problem with varying time delay longer than one sampling interval is discussed, and a preview controller is employed for precise motion control. A delay-dependent system model is first introduced and a reference generator is then employed to model the previewed future reference. The system model is augmented with the reference generator and an optimal controller is synthesized to minimize a quadratic cost function of tracking errors and control inputs. A time-varying Kalman filter is designed for state estimation and feedback control. To make the Kalman filter feasible under long time delay, a linear regression model is proposed for delay estimation based on past measurements. A new wireless protocol called RT-WiFi is developed for high-speed and real-time control applications. Using the delay measurement from the RT-WiFi network, simulation studies are conducted to verify the effectiveness of the proposed algorithm.

I. INTRODUCTION

Network control systems (NCSs) have gained significant attention over the past several decades because of its wide use in telerobotics, process control, vehicle systems, and so on. An overview of the networked control system can be found in [1], [2]. A local wireless network, which gets rid of communication cables, is one popular type of network media to transmit sensor and controller signals in a NCS. Compared to traditional wired control systems, wireless NCSs show great advantages such as enhanced mobility, remote operation, and improved safety. However, as is well known, a wireless NCS is inherently less reliable than the traditional wired control system due to time delay, packet loss, packet disorder, and so forth. Among the aforementioned drawbacks, varying time delay happens most frequently and it could significantly degrade the control performance and even destroy the stability of the NCS.

Since time delay is such an essential and critical problem in a NCS, it has been intensively investigated and various controller schemes have been proposed. To list a few, Smith predictor [3], communication disturbance observer [4], [5], sliding mode control [6], robust control [7], and model predictive control [8] have been researched to compensate for delay time. Linear quadratic Gaussian (LQG) control was employed in [9] to deal with random round-trip delay within one sampling interval and that work was extended to the case when round-trip delay is longer than one sampling interval in [10]. However, estimation of time delay, which is essential for implementing the time-varying Kalman filter, has not been fully investigated. It is worth mentioning that most of the approaches presented above focused on stability analysis or stabilizing controller design for NCSs. Networked tracking controller design is still a problem that needs to be further addressed [11].

Our motivation of this research comes from the fact that desired plant trajectories can be previewed in many applications, such as industrial robot manipulators [12] and humanoid robots [13]. The information of future reference can be used to design a tracking controller for improved tracking performance because it can help the system increase its effective bandwidth and respond faster to command signals. In this paper, the preview control technique [14], [15] is applied to a networked tracking control system with varying time delay. Major contributions of this paper include:

- The preview control technique is applied to a NCS for precise tracking under varying time delay.
- A linear regression model is proposed to estimate time delay for real-time implementation of Kalman filter when round-trip delay is longer than one sampling interval.
- A new wireless protocol called RT-WiFi is designed for high-speed and real-time control applications, and a testbed is set up. Delay data from the testbed are used to verify the performance of the proposed algorithm.

The remainder of this paper is organized as follows. In Section II, system modeling with time varying delay is discussed. Section III presents the preview controller design technique as well as the Kalman filter design. The RT-WiFi protocol and time delay measurement are introduced in Section IV. Section V presents the linear regression model for feedforward delay estimation and simulation results. Section VI concludes the paper and discusses some future work.

II. SYSTEM MODEL WITH TIME DELAY

Consider the following continuous-time single-input-single-output (SISO) system model:

\[ \dot{x}(t) = Ax(t) + Bu(t - \tau(t)) + B_ww(t) \]  \hspace{1cm} (1)

\[ y(t) = Cx(t) \]  \hspace{1cm} (2)

\[ y^m(t) = Cx(t) + v(t) \]  \hspace{1cm} (3)

where \( x(t) \in \mathbb{R}^n \) is the state vector of the system, \( u(t) \in \mathbb{R}^m \) is the generated controller signal, \( y(t) \in \mathbb{R} \) is the system
output, and $y^m(t) \in \mathbb{R}$ is the measured output. $w(t) \in \mathbb{R}$ and $v(t) \in \mathbb{R}$ are independent zero-mean Gaussian white input and measurement noises with constant covariances $W$ and $V$, respectively. $A$, $B$ and $C$ are system, input, and output matrices with appropriate dimensions.

To simplify the problem, the following assumptions are made for time delay modeling and controller design:

- Sensors in the system are time-driven, while controllers and actuators are event-driven. The timers of the sensor, controller, and actuator are synchronized.
- Delay in a NCS includes three components, sensor to controller (feedback channel) delay $\tau^c(t)$, computation delay in the controller $\tau^c(t)$, and controller to actuator (feedforward channel) delay $\tau^ca(t)$. Round trip delay $\tau(t)$ is the combination of the three delays above. In this paper, $\tau^c(t)$ is neglected as it is not a network-induced delay and it can be categorized into $\tau^ia(t)$ if necessary.
- The network induced delay is bounded as follows:

$$0 \leq \tau(t) \leq n_0T_s, \quad (4)$$

where $n_0$ is an integer and $T_s$ is the sampling time of the system.
- Input $u(t)$ is piecewise constant in a sampling interval.
- Disturbance and noise are independent of the delay.
- Packet loss and disorder are not considered.
- Initial state of the system is deterministic.
- $E\left(w(i)w^T(j)\right) = W\delta_{ij}$ and $E\left(v(i)v^T(j)\right) = V\delta_{ij}$. $W > 0$ and $V > 0$. $\delta_{ij} = 1$ when $i = j$ and $\delta_{ij} = 0$ when $i \neq j$.

Note that when there is a bounded time delay $\tau(t) \leq n_0T_s$, there are at most $n_0 + 1$ control inputs acting on the actuator during the sampling interval $[kT_s, (k + 1)T_s]$. The changes in $u(t)$ are assumed to occur at random instants $kT_s + t^i_k$. It is shown in Fig. 1 that $kT_s + t^i_k$ is the arrival time of control packet $u_{k-1}$ for $i = 0, 1, \ldots, n_0$. If the control packet $u_{k-1}$ arrives before $kT_s$, $t^i_k = 0$. If the packet $u_{k-1}$ arrives at or after $(k + 1)T_s$, since packet loss and disorder are not considered in this paper, it follows that $t^i_{k-1} \geq t^i_k$. Then the following discrete-time system model is reached:

$$x_{k+1} = A_s x_k + \sum_{i=0}^{n_0} B^i_k u_{k-i} + B_w w_k \quad (5)$$

$$y_k = C x_k \quad (6)$$

$$y^m_k = C x_k + v_k \quad (7)$$

where $A_s = e^{A T_s}$, $B^i_k = \int_{t^i_{k-1}}^{t^i_k} e^{A(T_s-t)} dt B$ with $t^i_{k-1} = T_s$ and $t^i_{k-1} = 0$, $x_k = x(kT_s)$, $y_k = y(kT_s)$, $y^m_k = y^m(kT_s)$, $w_k = \int_{kT_s}^{(k+1)T_s} e^{A(k+1)T_s-t} w(t) dt$, and $v_k = v(kT_s)$. $w_k$ and $v_k$ are still zero-mean Gaussian white noises. The system model indicates that $u_{k-i}$ is applied into the system from $kT_s + t^i_{k-1}$ to $kT_s + t^i_k$, where the former is the arrival time of packet $u_{k-i}$ and the latter is the arrival time of packet $u_{k-i+1}$.

![Fig. 1: Signal transmission in a NCS with a maximum round-trip delay of $n_0$ steps](image)

To achieve improved tracking performance, the following finite horizon linear quadratic cost function is introduced

$$J_N = \mathbb{E} \left[ \sum_{k=0}^{N-1} (e^T_k Q e_k + u^T_k R u_k) + e^T_N Q_N e_N \right], \quad (8)$$

where $Q$ and $Q_N$ are positive semi-definite, and $R$ is positive definite. Tracking error is defined as $e_k = r_k - y_k$ where $r_k \in \mathbb{R}$ is the reference signal at the $k$th time step.

### III. PREVIEW CONTROLLER DESIGN

#### A. Reference Generator and Augmented System Model

If future reference signals over a finite horizon are available, tracking performance can be improved by utilizing them. It is assumed that at time step $k$, only future reference signals up to time step $(k + N_p)$ can be previewed, where $N_p$ stands for the preview horizon. After time step $(k + N_p)$, the reference is modeled as a stochastic signal. Reference signal at the current step is generated by a shaping filter and then augmented into a reference generator to incorporate the future reference up to time step $k + N_p$ as follows:

$$x^d_{k+1} = A_d x^d_k + B_d w^d_{k+N_p} \quad (9)$$

$$r_k = C_d x^d_k \quad (10)$$

where $x^d_k = \begin{bmatrix} r_k & r_{k+1} & \cdots & r_{k+N_p-1} & x^r_{k+N_p} \end{bmatrix}^T$ and $x^r_k \in \mathbb{R}^q$ is the state of the shaping filter. $w^d_{k+N_p} \in \mathbb{R}$ is a Gaussian white noise with mean zero and constant variance $W_r$, and it is independent with $w_k$ and $v_k$. More details of the shaping filter and reference generator are available in [14], [15].

To consider previewed reference signals for controller design, the plant model and reference generator are augmented as follows:

$$x^a_{k+1} = A_a x^a_k + B_a u_k + B_w w^a_k \quad (11)$$

$$e_k = C_a x^a_k \quad (12)$$

where

$$x^a_k = \begin{bmatrix} x^d_k \\
 u_{k-1} \\
 \vdots \\
 u_{k-n_0} \\
 w_k \\
 w_{k+N_p} \end{bmatrix}, \quad w^a_k = \begin{bmatrix} w_k \\
 w_{k+N_p} \end{bmatrix}.$$
Ak = \[
\begin{bmatrix}
A_k & 0 & 0 & \cdots & 0 & 0 \\
0 & A_k & B_k^1 & \cdots & B_k^{n-1} & B_k^n \\
0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 1 \\
\end{bmatrix}
\]

Bk = \[
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
\vdots \\
0 \\
1 \\
\end{bmatrix}
\]

Based on the augmented model (11) and (12), the cost function (8) can be equivalently rewritten as

\[
J_N = \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k + x_N^T Q_N x_N^T
\]  

where

\[
R' = \begin{bmatrix}
R & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

\[
Q_N = C_d^T Q N C_a + 0
\]

\[
Q' = C_d^T Q C_a + 0
\]

It is straightforward to verify that \( Q' \) and \( Q_N \) are positive semi-definite, while \( R' \) is positive definite.

B. Design of an Optimal State Controller

Based on the augmented system model (11), (12) and cost function (13), dynamic programming approach is applied to get the optimal controller gain using the following theorem:

**Theorem 1:** Consider the augmented system (11), (12), and the cost function (13). Optimal controller gain at the \( k \)-th time step is given as

\[
L_k = - R' + \sum_{i=0}^{n_0-1} B_i^k P_{i+1} B_i^k + Q' + \sum_{i=0}^{n_0-1} P_i^k (A_i^k + B_i^k L_k)^T P_{i+1} (A_i^k + B_i^k L_k)
\]

where \( P_k \) is solved backwards recursively using the following equation with an initial value \( P_N = Q_N \):

\[
P_k = L_k^T R' L_k + Q' + \sum_{i=0}^{n_0-1} P_i^k (A_i^k + B_i^k L_k)^T P_{i+1} (A_i^k + B_i^k L_k)
\]

The optimal cost is given by

\[
J^* = b_0 + x_0^T P_0 x_0
\]

where \( P_0 \) is given by (15) and \( b_0 \) is calculated backwards recursively by

\[
b_k = b_{k+1} + \sum_{i=0}^{n_0-1} B_i^k B_i^T P_{i+1} B_i w_k
\]

**Proof:** Proof is similar to that of Theorem 1 in [15] and is omitted here due to the space limit.

C. State Estimator Design

It is often inconvenient or expensive to obtain a full state measurement. Moreover, sometimes the measurement is too noisy to be directly used for feedback control. In this subsection, a time-varying Kalman filter is employed as follows:

\[
\hat{x}_{k+1|k} = A_k \hat{x}_{k|k} + \sum_{i=0}^{n_0-1} B_i^k u_{k-i}
\]

\[
M_{k+1} = A Z_k A^T + B_k W B_k^T
\]

\[
K_{k+1} = M_{k+1} C (C M_{k+1} C^T + V)^{-1}
\]

\[
\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} (Y_{k+1} - C \hat{x}_{k+1|k})
\]

\[
Z_{k+1} = M_{k+1} - K_{k+1} C M_{k+1}
\]

where \( \hat{x}_{k|j} \) is the conditional expectation of \( x_k \) given \( Y_j = \{ \hat{y}_0, \hat{y}_1, \cdots, \hat{y}_j \} \). \( Z_k \) and \( M_{k+1} \) are the a-priori and a-priori state estimation error covariances, respectively.

It should be noted that since time delay in both feedback and feedforward channels might be longer than one sampling interval, it is possible that \( B_k^k \) is unknown to the controller for some \( i \) when the Kalman filter is about to run at the \( k \)-th time step. In such case, (18) becomes infeasible. To solve this problem, a linear regression model is proposed to estimate the feedforward channel delay in Section V.

D. Separation Principle and Stability Analysis

It has been proven in [9], [10] that separation principle holds for the proposed feedback control system with the Kalman filter if \( A_k \) and \( B_k^k \) in (18) are accurately known. Since the previewed reference signals are deterministic and no estimator is required, the separation principle still holds for the preview controller. Combining feedback controller gain (14) and Kalman filter (18) to (22) gives the following controller signal to be implemented:

\[
u_k = L_k \hat{x}_{k|k}
\]

where \( \hat{x}_{k|k} = \begin{bmatrix} a_k^T, \hat{y}_{k-1}, \cdots, \hat{y}_{k-n_0} \end{bmatrix} \). A block diagram of the overall control system is shown in Fig. 2. Since tracking control is achieved by adding reference feedforward control, it will not affect the stability of the feedback control system. A detailed stability analysis is availabe in [10].

IV. TIME DELAY MEASUREMENT IN RT-WiFi NETWORK

A. RT-WiFi Real-time Wireless Protocol

The proposed control algorithm relies on bounded delay for estimating and controlling the state of the plant. Moreover, high sampling rates are usually desirable for precise
motion control. To meet these requirements, a flexible real-time and high-speed wireless communication protocol called RT-WiFi [16] is designed for providing both real-time data delivery and high sampling rates. RT-WiFi is built on top of IEEE 802.11 physical layer and can provide a sampling rate up to 6 kHz. To achieve deterministic timing behavior, time division multiple access (TDMA) mechanism is adopted in its data link layer, and each transmission is configured on a pre-scheduled time slot. RT-WiFi also provides a flexible platform to let control engineers customize the design trade-offs among sampling rate, jitter level, reliability, and co-existence with regular Wi-Fi networks. To the best of our knowledge, RT-WiFi provides the highest sampling rate that can be supported by any wireless real-time communication protocols for flexible control systems.

B. Application Layer Time Delay of RT-WiFi Network

To emulate the network dynamics in the RT-WiFi network, a RT-WiFi network testbed was set up on the 5th floor of the GDC building at the University of Texas at Austin. The configuration of this testbed is shown in Fig. 3. There are one RT-WiFi access point (AP) and one RT-WiFi station in this testbed. Both RT-WiFi nodes were equipped with Atheros AR9285 Wi-Fi chips, and they ran Ubuntu operating system with RT-WiFi software stack. To mimic the behavior of the proposed controller, two pairs of application layer programs were installed and ran on top of the two RT-WiFi nodes for emulating data transmission in the feedback and feedforward channels. The application layer programs were configured to transmit an application message with a payload size of 500 bytes every 1 ms (sampling rate 1 kHz). To observe the time delay in the feedback and feedforward channels, a precise system clock synchronization was established using IEEE 1588 Precision Time Protocol (PTP) [17]. The end-to-end application layer time delay between the two RT-WiFi nodes was measured using the time stamps contained in the application messages, as is shown in Fig 4.

Several interesting phenomena are observed in Fig. 4. First of all, the delay variation between consecutive packets is due to the application layer scheduling delay and the interference of other broadcast messages in the surrounding environment. Since the Ubuntu is not a real-time operating system, packet transmissions of application layer programs might be delayed nondeterministically. In addition, the transmission of RT-WiFi packets can be postponed due to receiving broadcast messages from other APs in the environment. Secondly, it is observed that time delay has a zigzag trend over time. This is because of the timing mismatch between the timer in the RT-WiFi module and the timer in the control application. Therefore, every packet has different queuing delay in the RT-WiFi transmission module. If the queuing delay is larger than the threshold defined in the RT-WiFi protocol, that
As for the preview horizon design, it was proposed in [14] that a good estimate of the maximum effective preview feedforward delays of the control packets. A. Estimation of Time Delay in the Feedforward Channel

As is mentioned in Section III-C, both feedback and feedforward delays of the control packets \( u_k, \ldots, u_{k-n_0} \) need to be available for the Kalman filter to calculate \( B^*_k \) in (18). Although acknowledgements for some packets can be sent back to the controller about the round-trip delay before the Kalman filter starts to run, some acknowledgements may not be available due to time delay. Note that most sensing packets in the feedback channel can be delivered within one sampling interval so feedback channel delay is available, but it takes longer to receive an acknowledgement from the actuator (the actuator needs to receive the packet first and then sends out an acknowledgement with the amount of feedforward delay back to the controller). Therefore, it is very important to estimate the feedforward channel delay. In this subsection, a linear regression model is used to estimate the delay in the feedforward channel based on past delay measurement.

Based on the delay measurement data shown in Fig. 4, given a sampling time of 1000 \( \mu s \), feedback channel delay of the \((k-1)^{th}\) packet \( \tau_{k-1}^{\text{ca}} \) and feedforward channel delay of the \((k-3)^{th}\) packet \( \tau_{k-3}^{\text{ca}} \) are available when the Kalman filter runs at the \(k^{th}\) step. Therefore, those two measurements are chosen as covariates and feedforward channel delay of the current control packet \( \tau_k^{\text{ca}} \) is chosen as the response variable in the regression model. A physical interpretation of the covariate choice is as follows: \( \tau_{k-1}^{\text{sc}} \) is our best knowledge of the current network environment (because it is closer to the current time step) and \( \tau_{k-3}^{\text{ca}} \) is our best knowledge of the current network traffic load in the feedforward channel. Based on least square estimation, the following regression model was built:

\[
\tau_k^{\text{ca}} = 354.7 - 0.0983\tau_{k-1}^{\text{sc}} + 0.5328\tau_{k-3}^{\text{ca}}
\]  

(24)

In comparison, the mean value of the previous feedforward channel delay was used as an estimate of the feedforward channel delay in past literatures such as [9]. As a comparison, root-mean-square (RMS) errors and coefficient of determination \( R^2 \) are shown in Table I, which indicates a 19.5% reduction of RMS error of the current feedforward delay estimation. Moreover, \( R^2 \) also increases so that more variation of the delay can be captured by the proposed regression model.

B. Simulation Setup and Choice of Preview Horizon

In this subsection, the proposed preview control technique is verified by a simulation study. An open-loop stable brushless DC (BLDC) motor was chosen as the plant with a sampling rate of 1 kHz. The delay measurement shown in Fig. 4 was used for simulation study and \( n_0 \) in (5) was chosen as three. In the preview controller design, weighting matrices \( Q, Q_N \), and \( R \) in (8) were chosen as 100, 100, and 1, respectively. This reference generator was designed as a low-pass filter with a cut-off frequency of 5 Hz. The reference trajectory of the motor is shown in Fig. 6.

As for the preview horizon design, it was proposed in [14] that a good estimate of the maximum effective preview

<table>
<thead>
<tr>
<th>Preview horizon</th>
<th>Linear regression model (24)</th>
<th>Mean delay method in [9]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_p = 5 )</td>
<td>0.2548</td>
<td>0.2843</td>
</tr>
<tr>
<td>( N_p = 10 )</td>
<td>0.2250</td>
<td>0.2435</td>
</tr>
<tr>
<td>( N_p = 30 )</td>
<td>0.2045</td>
<td>0.2186</td>
</tr>
<tr>
<td>( N_p = 50 )</td>
<td>0.1831</td>
<td>0.2091</td>
</tr>
<tr>
<td>( N_p = 60 )</td>
<td>0.1813</td>
<td>0.2059</td>
</tr>
</tbody>
</table>

TABLE I: Comparison of the two feedforward delay estimation methods

<table>
<thead>
<tr>
<th></th>
<th>RMS errors</th>
<th>( R^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression model (24)</td>
<td>97.30( \mu s )</td>
<td>0.3522</td>
</tr>
<tr>
<td>Mean delay method in [9]</td>
<td>120.90( \mu s )</td>
<td>0</td>
</tr>
</tbody>
</table>
TABLE III: Values of cost function under different preview horizons and delay estimation methods

<table>
<thead>
<tr>
<th>Preview horizon</th>
<th>Linear regression model (24)</th>
<th>Mean delay method in [9]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_p = 5$</td>
<td>$8.0421 \times 10^4$</td>
<td>$9.9792 \times 10^4$</td>
</tr>
<tr>
<td>$N_p = 10$</td>
<td>$6.2873 \times 10^4$</td>
<td>$7.3378 \times 10^4$</td>
</tr>
<tr>
<td>$N_p = 30$</td>
<td>$5.1573 \times 10^4$</td>
<td>$5.8913 \times 10^4$</td>
</tr>
<tr>
<td>$N_p = 50$</td>
<td>$4.1345 \times 10^4$</td>
<td>$5.3664 \times 10^4$</td>
</tr>
<tr>
<td>$N_p = 60$</td>
<td>$4.0520 \times 10^4$</td>
<td>$5.2225 \times 10^4$</td>
</tr>
</tbody>
</table>

horizon $N_{pc}$ can be three times the longest closed-loop plant time constant, which can be approximated by the settling time with 5% tolerance. Using the procedures given in [15], one could get $N_{pc} = 48.5180 \approx 49$. Note that the system and input matrices are time-varying, so their mean values were used to calculate $N_{pc}$ and controller gain numerically. Tracking performance with different preview horizons and the two ways of estimating feedforward delay will be compared in the next subsection.

C. Simulation Results and Discussions

The estimation accuracy of the Kalman filter is first examined. The regression model (24) was used in the Kalman filter design for feedforward channel prediction. The preview horizon was chosen as $N_p = 50$ considering the $N_{pc}$ calculated before. It is shown in Fig. 5a that position estimate is accurate with noises and long time delay. Estimated velocity is smoother than direct position differentiation, as is shown in Fig. 5b.

Tracking performance of the proposed controller with preview horizon $N_p = 50$ is shown in Fig. 6, which verifies the good tracking performance under time-varying delay in the NCS. To further examine the tracking performance under various preview horizons and delay estimation techniques, RMS tracking errors and values of the cost function are summarized in Table II and Table III. It is shown that both RMS tracking errors and cost functions based on regression model are smaller than those from the mean delay method proposed in [9]. Moreover, RMS tracking errors and cost functions become smaller as the preview horizon increases. When preview horizon goes beyond $N_{pc}$, the RMS tracking errors and cost function decrease slowly with increased preview horizons.

VI. CONCLUSION

In this paper, a preview control technique was proposed to compensate for varying long time delay in a wireless tracking control system. A time-varying Kalman filter considering the time delay was employed to achieve full state estimation for state feedback control. A linear regression model was proposed to estimate delay in the feedforward channel for running Kalman filter in real-time. A new wireless protocol called RT-WiFi was used for high-speed and real-time control applications. Time delay in the RT-WiFi network was measured for both feedback and feedforward channels. Choice of preview horizon was discussed and performance of the proposed algorithm was verified by simulations.

The proposed algorithm can be applied to a wireless rehabilitation system to guarantee that accurate assistive torque is provided to patients even with time delay in the network [18]. As future work, the rehabilitation device in [18] will be integrated with the RT-WiFi network and the method proposed in this paper will be verified by experimental results.

REFERENCES