Virtual Sensors. An overview

P. Albertos[1], I. Peñarrocha[2], R. Sanchis[2]

[1] Departamento de Ingeniería de Sistemas y Automática
Polytechnic University of Valencia, Spain.
pedro@aii.upv.es
[2] Departament d'Enginyeria de Sistemes Industrials i Disseny
Universitat Jaume I de Castelló, Spain.
{ipenarro,rsanchis}@esid.uji.es

Abstract. A virtual sensor is an information-processing based device designed to get information about an internal process variable which is not directly accessible. It may receive information from several sensors as well as the sequence of process inputs, updated following a given pattern. By using a model of the process, the desired variables are estimated at the desired rate. Virtual sensors can be used to monitor the evolution of some internal variables, some quality indices or some parameters in the process. Moreover, the estimated variables can be also used to feed a control system or to provide some sort of redundancy in the measurements. In this way, richer feedback control loops can be implemented or more reliable redundant control systems can be designed. In this plenary talk, some of the main issues in the design of virtual sensors are discussed and their main features, such as robustness or convergence, are analyzed. Some illustrative applications illustrate the main concepts.

1 Introduction

The main goal of a sensing device is to provide reliable, precise, up-to-date and concise information about a process variable, parameter or index, to be used for different monitoring or control purposes. The basic *transducer* device converts a variable into another one which is easier to handle. It is also typical to get an improvement in the sensed variable by correcting this basic measurement by means of related information based on the characteristics of the transducer (non-linear compensation) or the environment (temperature compensation).

Modern computer-based systems allow for more versatile sensors providing such information without the need for extra hardware. Additional features such as device identification, change in the range of operation, availability, storage of previous values and so on are also typical in modern sensors. This leads to the concept of *smart sensors*, that is, sensors devices providing not only the value of the sensed variable (one of the possible optional measurements) but also data about the operating conditions, location of the device and quality of the measurement. This data allow for a better management of the full data acquisition system.

© L. Sánchez, O. Pogrebnyak and E. Rubio (Eds.)
Industrial Informatics.
Research in Computing Science 31, 2007, pp. 1-14

The use of AI techniques also allows for some sort of intelligence in performing these tasks. As a result, these devices are sometimes known as ***intelligent sensors***, being also able to validate, communicate and report the sensed variable and reconfigure its operation based on additional knowledge about environmental conditions.

The common features of advanced sensors include: to convert the sensed variable into a treatable signal (in digital form), to cover a wide range of interest, to provide the best measurements in terms of accuracy, timing (conversion time, delays), disturbances (noise and drifts) or linearity, to prevent maintenance and detect malfunctioning, and to communicate with the user as well as the rest of components.

Most of these devices, originally working in continuous time, are conceived to operate under regular sampling/updating patterns.

The concept of a ***virtual sensor*** is rather different. The estimated value of the variable is computed as a result of the action of the process inputs on a model of the process, combined with additional information taken from the process. In this way, the sensed variable is not directly measured, it may be even not accessible or physically measurable, but there exists a model relating its time evolution with that of the available variables. Virtual sensors may also incorporate some of the features already mentioned for smart or intelligent sensors. Virtual sensors can be used to monitor the evolution of some internal variables, some quality indices or some parameters in the process. Moreover, the estimated variables can be also used to detect faulty conditions, to feed a control system or to provide some sort of redundancy in the measurements. In this way, richer feedback control loops can be implemented or more reliable redundant control systems can be designed.

Moreover, the constraint of regular sampling/updating pattern may be removed in such a way that the measurements are gathered at different time instants. Figure 1 represents a continuous-time process manipulated by several continuous-time inputs ($u(\tau), \tau \in \mathbb{R}$, measurable, and $v_c(\tau)$, unmeasurable) and several output signals ($y(\tau)$) that can be measured with different rates and at different time instants. Some of them could be measured with redundant sensors of different precision, different sampling rates and different associated delay, leading to the discrete-time measurement values $m_{i,k}$, where i refers to the sensor number.

Assume that the values of some internal variables $z(\tau)$, which are not directly measurable but are a function of the measurable ones, are needed for monitoring or control purposes. Moreover, assume they are required at a fixed frequency on predefined time instants. In this scenario, a virtual sensor is defined as the information-processing device that based on the knowledge of the applied inputs and the measurement of the available outputs at arbitrary instants, predicts the values of the desired signals at the required time instants [1].

Among the virtual sensors applications the following cases can be considered: a) the process variable cannot be directly measured, i.e. measuring some performance; b) the physical sensor is too expensive, not enough accurate or too slow, like a chromatograph; c) the sensor placement is not accessible, like the

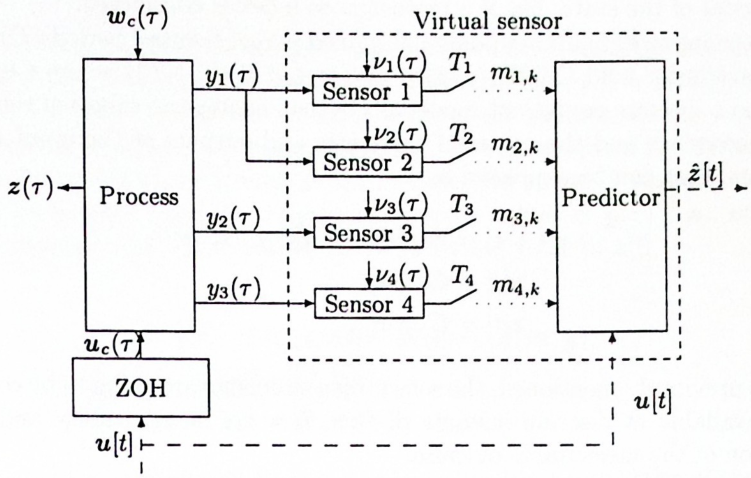


Fig. 1. General setting for internal knowledge estimation in MIMO process with several sensors.

burning temperature in a cement kiln; or d) it should be too far away of the place of interest, like the moisture of the paper web at the headbox exit. Other situations may also require a virtual sensor if, for instance, there is no room for a physical device or it is very expensive to maintain. Some applications of virtual sensors are reported, i.e. in [2] and [3].

Several approaches can be used for this prediction. The simplest one is to interpolate or extrapolate the measurements with any interpolation method. This idea could result in a computationally simple predictor that can be applied if the targeted variables are measurable. However, if the frequency of the measurements is not high enough the predictions could have a large error. In these situations the predictions can be significantly improved if a model of the process is used and the inputs as well as the output measurements are taken into account in the predictor algorithm.

The model of the process can be identified off line or can be estimated online by an adaptive algorithm running on the virtual sensor. In this work the general case of unmeasured output prediction is studied, where several signals that are related to the output are assumed to be measured at different instants (the desired output may be one of them, but not necessarily).

The process is assumed to be a continuous linear time-invariant system modeled by the equations

$$\dot{\boldsymbol{x}}(\tau) = \boldsymbol{A}_c\, \boldsymbol{x}(\tau) + \boldsymbol{B}_c\, \boldsymbol{u}_c(\tau) + \boldsymbol{B}_{wc}\, \boldsymbol{w}_c(\tau), \tag{1a}$$

$$\boldsymbol{y}(\tau) = \boldsymbol{C}_y\, \boldsymbol{x}(\tau), \tag{1b}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ is the input vector, $\boldsymbol{w}_c(\tau) \in \mathbb{R}^{n_w}$ is the continuous time disturbance vector and $\boldsymbol{y} \in \mathbb{R}^{n_y}$ is the measured output vector. The vector of targeted variables can be written as

$$\boldsymbol{z}(\tau) = \boldsymbol{C}_z\, \boldsymbol{x}(\tau), \tag{1c}$$

where $z(\tau) \in \mathbb{R}^{n_z}$. In the general case, $z(\tau)$ can be a nonlinear function or a functional of the state, like if a performance index is considered.

If the input signals are updated at a fixed period (control period) T by means of a zero order hold (ZOH) ($u(\tau) = u[t], \tau \in [tT, tT+T)$) where $t \in \mathbb{N}$, then there is a discrete equivalent model of (1) that relates the values of the discrete input sequence and the values of the states and outputs at the input updating instants, that can be expressed as

$$x[t+1] = A\,x[t] + B\,u[t] + B_w\,w[t], \tag{2a}$$

$$y[t] = C_y\,x[t], \tag{2b}$$

$$z[t] = C_z\,x[t]. \tag{2c}$$

As previously mentioned, the sensor measurements are going to be considered only available at discrete instants of time $\tau = \tau_k$, being a noisy and delayed function of the measurable outputs:

$$m_{i,k} = c_i\,x(\tau_k - \delta_{i,k}) + \nu_{i,k}, \quad i = 1, \ldots, n_m \tag{3}$$

where $m_{i,k}$ is the available measurement of sensor i at the k-th sampling (not all sensor measurements are available at each sampling time), with delay $\delta_{i,k}$ and measurement noise $\nu_{i,k}$, and n_m is the number of sensors. The availability factor $\alpha_{i,k}$ defined as

$$\alpha_{i,k} = \begin{cases} 1, \text{ if } m_{i,k} \text{ is available,} \\ 0, \text{ if } m_{i,k} \text{ is not available,} \end{cases} \tag{4}$$

indicates which are the available measurements at the k-th sampling instant. If some of the sensors have their own non negligible dynamics, the last representation is also valid just by using some of the process states as sensor states (see [4,5]).

If the sensor measurements are assumed to be only available at instants $t = t_k$ (synchronously with the input update) and are affected by a variable time delay, the measurement equation (3) can be written

$$m_{i,k} = c_i\,x[t_k - d_{i,k}] + \nu_{i,k}, \quad i = 1, \ldots, n_m \tag{5}$$

where $d_{i,k}$ is the discrete delay (measured in number of control periods) assigned to sensor i in the k-th sampling.

The rest of this paper is structured as follows. In the next section the basic structure of a virtual sensor is described. The simplest and more useful situations are analyzed and the sensor properties are summarized. For the sake of clarity, all the treatment is done in discrete time. Continuous time and random occurrence of measurements and input updating considerations introduce additional complexity in the treatment [5]. The general case of arbitrary sampling/updating pattern is considered in Section 3. A Kalman filter approach, which is also valid in the previous cases, is generalized in Section 4. Then, a typical application is presented and some conclusions are drafted in the final Section.

2 Virtual sensor structure

In the following, only discrete time systems are considered. Assume that some variable values are required at each input updating, $z[t]$ ($z[t] = z(tT)$), but they are not available. Instead, the values of some of the elements of the vector $y(\tau)$ are measured at different sampling instants. The virtual sensor is a dynamic system similar to (2) whose inputs are the input action at rate T and an *innovation* term $\mathbb{J}$ elaborated from the irregularly sampled values of $y(\tau)$ (i.e., $m_{i,k}$) and their estimates, and whose output is the predicted value of $z[t]$.

$$\hat{x}[t+1] = A\,\hat{x}[t] + B\,u[t] + \mathbb{J}(m, \hat{y}), \tag{6a}$$

$$\hat{y}[t] = C_y\,\hat{x}[t], \tag{6b}$$

$$\hat{z}[t] = C_z\,\hat{x}[t]. \tag{6c}$$

The innovation term should be designed to assure some properties of the predictor. In particular, the sensing error

$$\tilde{z}[t] = z[t] - \hat{z}[t] \tag{7}$$

should be small and converge to zero under nominal and constant operating conditions, regardless the stability condition of the plant (the plant matrix A not necessarily being Hurwitz). Also, the sensitivity of this error with respect to measurement noises and plant disturbances should be bounded.

Depending on the sampling scenario, different prediction algorithms can be used to estimate the state and the targeted variables at the input updating instants.

2.1 Conventional sampling

The simplest case corresponds to a regular sampling/updating of all the signals, with a period T and without any delay in any of the measurement signals. The innovation term is made proportional to the output estimation error, i.e.

$$\mathbb{J}(m, \hat{y}) = L(m - C\hat{x}), \tag{8}$$

where C is the matrix formed with the rows c_i that define each sensor equation (5). As it is well known [6], the error dynamics is characterized by the eigenvalues of $A - LC$.

In this regular case, this innovation sequence can be filtered by a discrete transfer matrix $L(z)$. The virtual sensor can be represented in block diagram as depicted in Figure 2, where it has been assumed that $n_m = n_y$.

As already mentioned, by $L(z) = L$ a constant matrix, the stability of the sensor (like in any Luenberger-type observer) can be ensured, if the pair (A, C) is observable. By the appropriate selection of the *filter* $L(z)$, the sensitivity functions relating the estimated output with respect to noise and plant disturbance can be fine tuned (see, i.e., [4] and the references therein).

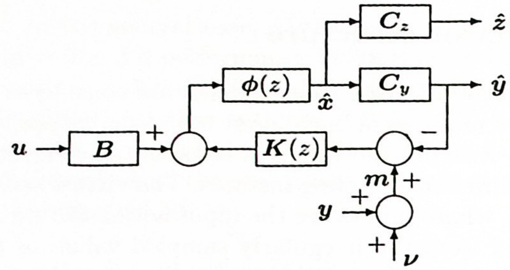


Fig. 2. Innovation in virtual sensor.

Denoting by $\phi(z) = (zI - A)^{-1}$, the noise sensitivity function is

$$T_{\tilde{z}\nu} = C_z \left(\phi(z)^{-1} + L(z)C\right)^{-1} L(z). \tag{9}$$

An analysis of the shape of the frequency response of this function will enlighten the possible options in selecting L.

2.2 Nonconventional regular sampling

Assume now that all the measurements are available without delay periodically at the same instants of time, being N the number of input updates between two consecutive measurement instants ($N = t_k - t_{k-1}$). In this case, the innovation term (8) is only applied at sampling instants, leading to the prediction error dynamics (in the absence of disturbances):

$$\tilde{x}_{k+1} = \mathcal{A}\tilde{x}_k = (I - LC)\, A^N \tilde{x}_k, \tag{10}$$

$$\tilde{z}_k = C_z \tilde{x}_k, \tag{11}$$

The state estimation error is defined as $\tilde{x}_k = x[t_k] - \hat{x}[t_k]$. The necessary and sufficient condition to stabilize the predictor is to take a matrix L such that all eigenvalues of $\mathcal{A}$ are inside the unit circle ([5]).

In this case, the analysis by using matrix transfer functions is rather complex involving $\mathcal{Z}$-modified transform representations ([7]).

3 Nonconventional general sampling

Assume now that measurements are taken synchronously with the input updating instants, but not all sensors are available at every sampling instant. Assume also that the available sensors have a different time-varying delay. Then, a different prediction algorithm must be developed in order to make use of all the available measurements in spite of their availability and delays. The state is first periodically updated running the model (2) in open loop from the last sampling instant t_{k-1}

$$\hat{x}[t|t_{k-1}] = A^{t-t_{k-1}}\,\hat{x}[t_{k-1}] + \sum_{i=1}^{t-t_{k-1}} A^{i-1} B\,u[t-i]. \tag{12a}$$

where $\hat{x}[t|t_{k-1}]$ represents the state estimation at time t with the available measurements until t_{k-1}, and $\hat{x}[t_{k-1}]$ represents the state estimation at t_{k-1} with the available measurements until t_{k-1}, i.e., the best estimation of $x(t_{k-1})$. If a new measurement is available at $t = t_k$ the state is updated as

$$\hat{x}[t_k] = \hat{x}[t_k|t_k-1] + \sum_{i=1}^{n_m} \ell_{i,k}\,(m_{i,k} - c_i\,\hat{x}[t_k - d_{i,k}|t_k-1])\,\alpha_{i,k}. \tag{12b}$$

where $\ell_{i,k}$ is the innovation gain vector used to update the state estimation with the measurement $m_{i,k}$. Equation (12b) uses the delayed state estimation $\hat{x}[t_k - d_{i,k}|t_k - 1]$, that is calculated as

$$\hat{x}[t_k - d_{i,k}|t_{k-1}] = A^{-d_{i,k}}\,\hat{x}[t_k|t_{k-1}] + \sum_{j=1}^{d_{i,k}} A^{j-1-d_{i,k}}\,B\,u[t_k - j]. \tag{12c}$$

To reduce the computational cost on the delayed state estimation, an extended state vector including all the necessary past estate estimations can be used (see [8]). The desired output prediction is obtained by means of the output equation $\hat{z}[t] = C_z\,\hat{x}[t]$.

The features of the predictor are determined by the gains $\ell_{i,k}$ used to update the state estimation upon the sampling of each measurement, that is, the innovation matrix gain

$$L_k = \left[\ell_{1,k}\;\ell_{2,k}\;\cdots\;\ell_{n_m,k}\right] \tag{13}$$

The predictor matrix gain must be designed to assure the predictor stability, robustness to the sporadic data availability, and disturbance and noise attenuation. The predictor gain is, in general, time-varying, but the case of a constant gain can be also feasible. In order to design a predictor, i.e., to determine the predictor matrix gain (13), with those properties several techniques can be used, depending on the nature of the sampling scenario and disturbances. The different design techniques considered are pole placement, LMI based disturbance attenuation, and Kalman filtering. The situations in which each of the techniques are more suitable will be discussed in the following.

The case developed in subsection 2.2 is a particular case in which the measurements were taken regular and periodically and, therefore, the pole placement technic is applicable. If the measurements are taken synchronously with the input update but they have irregular delay and availability, the pole placement technique cannot be applied. In order to derive an LMI based design technique some definitions must be made first.

The measurements $m_{i,k}$ are assumed to be available irregularly at discrete instants $t = t_k$, $t \in \mathbb{N}$, $k \in \mathbb{N}$, being $N_k = t_k - t_{k-1}$ the number of input updates from t_{k-1} to t_k, and, therefore $t_k = \sum_{i=1}^{k} N_i$ represents the instant in which the t-th input update occurs and the k-th sample is available (formed with the values of some of the sensors). The availability matrix at instant t_k is defined as $\boldsymbol{\alpha}_k = \mathrm{diag}\{\alpha_{1,k}, \ldots, \alpha_{n_m,k}\}$. The sampling scenario parameter s_k is defined as the combination of sensor availability, time between samples and sensor delay $(\boldsymbol{\alpha}_k, N_k, d_{i,k})$ that defines a sample, and enumerates all the possible sampling situations as

$$s_k \in \mathcal{S} = \{1, 2, \ldots, n_{\mathcal{S}}\}, \tag{14}$$

where $n_{\mathcal{S}}$ is the number of possible combinations. All the variables that define the sampling scenario can be expressed as a function of this parameter, i.e., $N_k = N(s_k)$, $d_{i,k} = d_i(s_k)$, and $\boldsymbol{\alpha}_k = \boldsymbol{\alpha}(s_k)$. If the updating gain matrix is also defined to have a different value for each possible value of the sampling parameter, that is

$$\boldsymbol{L}_k = \boldsymbol{L}(s_k) \in \boldsymbol{\mathcal{L}} = \{\boldsymbol{L}(1), \boldsymbol{L}(2), \ldots, \boldsymbol{L}(n_{\mathcal{S}})\}, \tag{15}$$

then, the prediction error dynamics is demonstrated to be of the parametric form (see [8])

$$\tilde{\boldsymbol{x}}_k = \boldsymbol{\mathcal{A}}(s_k)\, \tilde{\boldsymbol{x}}_{k-1} + \boldsymbol{\mathcal{B}}(s_k)\, \mathbb{V}_k \tag{16a}$$

$$\tilde{\boldsymbol{z}}_k = \boldsymbol{C}_z\, \tilde{\boldsymbol{x}}_k, \tag{16b}$$

with $\mathbb{V}_k = \begin{bmatrix} \boldsymbol{w}[t_k - 1] \\ \boldsymbol{w}[t_k - 2] \\ \vdots \\ \boldsymbol{w}[t_k - \beta_k] \\ \boldsymbol{\nu}_k \end{bmatrix}$, $\beta = \max\{d_1(s_k), \ldots, d_{n_m}(s_k), N(s_k)\}, \forall s_k \in \mathcal{S}$,

and where

$$\boldsymbol{\mathcal{A}}(s_k) = (\boldsymbol{I} - \boldsymbol{L}(s_k)\, \boldsymbol{\alpha}(s_k)\, \boldsymbol{C}_d(s_k))\, \boldsymbol{A}^{N(s_k)}, \qquad \boldsymbol{C}_d(s_k) = \begin{bmatrix} \boldsymbol{c}_1\, \boldsymbol{A}^{-d_1(s_k)} \\ \vdots \\ \boldsymbol{c}_{n_m}\, \boldsymbol{A}^{-d_{n_m}(s_k)} \end{bmatrix}_{n_m \times n},$$

$$\boldsymbol{\mathcal{B}}(s_k) = \begin{bmatrix} \boldsymbol{\Lambda}(N(s_k)) - \boldsymbol{L}(s_k)\, \boldsymbol{\alpha}(s_k)\, \boldsymbol{\mathcal{C}}_d(s_k) & -\boldsymbol{L}(s_k)\, \boldsymbol{\alpha}(s_k) \end{bmatrix}$$

$$\boldsymbol{\mathcal{C}}_d(s_k) = \begin{bmatrix} \boldsymbol{c}_1\, \boldsymbol{A}^{-d_1(s_k)}\, (\boldsymbol{\Lambda}(N(s_k)) - \boldsymbol{\Lambda}(d_1(s_k))) \\ \vdots \\ \boldsymbol{c}_{n_m}\, \boldsymbol{A}^{-d_{n_m}(s_k)}\, (\boldsymbol{\Lambda}(N(s_k)) - \boldsymbol{\Lambda}(d_{n_m}(s_k))) \end{bmatrix}_{n_m \times \beta n}.$$

$$\boldsymbol{\Lambda}(N(s_k)) = \underbrace{[\overbrace{\boldsymbol{B}_w \;\; \boldsymbol{A}\boldsymbol{B}_w \;\; \boldsymbol{A}^2\boldsymbol{B}_w \;\; \cdots \;\; \boldsymbol{A}^{N(s_k)-1}\boldsymbol{B}_w}^{N(s_k)} \;\; \boldsymbol{0} \;\; \cdots \;\; \boldsymbol{0}]}_{\beta}{}_{n \times \beta n_w}. \tag{17}$$

With the introduction of parameter s_k the error dynamics is a parametrically time-varying linear system, specifically a jump linear system, where the disturbances and noise vectors ($\mathbb{V}_k$) are the inputs, the state estimation error ($\tilde{\boldsymbol{x}}_k$) is the state, and the desired output prediction error ($\tilde{z}_k$) is the output.

The predictor design objective is to find a procedure to calculate the matrix $\boldsymbol{L}(s_k)$ (time varying with s_k, or constant) that leads to an adequate performance in terms of disturbance attenuation and robustness to the time varying delay and availability. The main idea consists of minimizing the norm of the transfer function from input $\mathbb{V}_k$ to output $\tilde{z}_k$. Different norms can be used depending on the known norms that characterize the disturbances. If the ℓ_2 norm of the disturbances is known, the $\mathcal{H}_\infty$ norm can be minimized in order to minimize the ℓ_2 norm of the prediction error. If the ℓ_∞ or RMS norm of the disturbances is known, minimization of $\mathcal{H}_\infty$ implies the minimization of the RMS value of the prediction error. If the disturbances are white noises of known variance, minimization of $\mathcal{H}_2$ norm will imply the minimization of RMS norm of the prediction error.

As an example, the design procedure for one of the transfer function norms is shown. If the ℓ_2 norm of the disturbance and noise signals are known, it is possible to minimize the norm $\|\tilde{z}_k\|_2$ by means of minimizing the sum

$$\sum_{i=1}^{n_w} \gamma_{w_i}^2 \|w_{i,k}\|_2^2 + \sum_{i=1}^{n_m} \gamma_{\nu_i}^2 \|\nu_{i,k}\|_2^2$$

along all variables γ_{w_i}, γ_{ν_i}, $\boldsymbol{P}(s_k) = \boldsymbol{P}(s_k)^\top \in \mathbb{R}^{n\times n}$, $\boldsymbol{Q}(s_k) \in \mathbb{R}^{n\times n}$, $\boldsymbol{X}(s_k) \in \mathbb{R}^{n\times n_m}$, $s_k \in \mathcal{S}$ that satisfy the LMI equation

$$\begin{bmatrix} \boldsymbol{Q}(s_k) + \boldsymbol{Q}(s_k)^\top - \boldsymbol{P}(s_k) & \boldsymbol{M}_A(s_k) & \boldsymbol{M}_B(s_k) \\ \boldsymbol{M}_A(s_k)^\top & \boldsymbol{P}(s_{k-1}) - \boldsymbol{C}_y^\top \boldsymbol{C}_y & \boldsymbol{0} \\ \boldsymbol{M}_B(s_k)^\top & \boldsymbol{0} & \boldsymbol{\Gamma}^2 \end{bmatrix} \succ 0 \tag{18}$$

with

$$\boldsymbol{M}_A(s_k) = (\boldsymbol{Q}(s_k) - \boldsymbol{X}(s_k)\boldsymbol{\alpha}(s_k)\boldsymbol{C}_d(s_k))\, \boldsymbol{A}^{N(s_k)}, \tag{19}$$

$$\boldsymbol{M}_B(s_k) = \left[\boldsymbol{Q}(s_k)\boldsymbol{\Lambda}(N(s_k)) - \boldsymbol{X}(s_k)\boldsymbol{\alpha}(s_k)\boldsymbol{C}_d(s_k)\ \ -\boldsymbol{X}(s_k)\boldsymbol{\alpha}(s_k)\right] \tag{20}$$

$$\boldsymbol{\Gamma} = \mathrm{diag}\{\boldsymbol{\Gamma}_w', \boldsymbol{\Gamma}_\nu\}, \qquad \boldsymbol{\Gamma}_w' = \frac{1}{\beta}\mathrm{diag}\{\boldsymbol{\Gamma}_w, \ldots, \boldsymbol{\Gamma}_w\}_{\beta n\times\beta n},$$

$$\boldsymbol{\Gamma}_w = \mathrm{diag}\{\gamma_{w_1}, \ldots, \gamma_{w_n}\}, \qquad \boldsymbol{\Gamma}_\nu = \mathrm{diag}\{\gamma_{\nu_1}, \ldots, \gamma_{\nu_{nm}}\}.$$

with matrix $\boldsymbol{\Lambda}(N(s_k))$ defined as (17). This problem can be solved with LMI standard solvers. Finally, the matrix gain that minimizes the norm error is $\boldsymbol{L}(s_k) = \boldsymbol{Q}(s_k)^{-1}\boldsymbol{X}(s_k)$. This solution gives a different gain for each sampling scenario, that leads to a scheduled-gain predictor implementation. If some restrictions are made over matrices $\boldsymbol{Q}(s_k)$ and $\boldsymbol{X}(s_k)$, it is possible to obtain different implementations. For example, if $\boldsymbol{Q}(s_k)$ and $\boldsymbol{X}(s_k)$ are chosen to be constant (i.e., $\boldsymbol{Q}(s_k) = \boldsymbol{Q}$ and $\boldsymbol{X}(s_k) = \boldsymbol{X}$) a constant gain $\boldsymbol{L}$ is obtained leading to the predictor with the lowest computational cost (see [9]).

Application to asynchronous sampling. When dealing with asynchronous measurements (taken at arbitrary time instants), the use of a continuous time model is necessary, implying matrix exponentials computation and, therefore, a high computational cost. In order to avoid this, an interpolation is proposed to first estimate the value of the output measurement at synchronous instants with the input update (when $\tau = tT$). If τ_k ($tT < \tau_k < (t+1)T$) is the instant in which an asynchronous measurement $y(\tau_k)$ takes place, the idea is to use an interpolation technique that gives a synchronous measurement m_k corresponding to the instant t or $t+1$, i.e., $m_k = f(y(\tau_k), \tau_k)$. When this technique is applied, a new measurement noise is introduced, whose value depends on the interpolation technique. As the interpolation technique is known, the noise norm due to interpolation errors can be bounded and, therefore, the previous technique is also applicable (see [5]).

4 Kalman filter techniques

Assume that the disturbances $\boldsymbol{w}[t]$ and $\boldsymbol{\nu}_k$ are white noise with variance-covariance matrices $\boldsymbol{W}$ and $\boldsymbol{V}$ respectivelly. ($\mathcal{E}\{\boldsymbol{w}[t]^\top \boldsymbol{w}[\tau]\} = \boldsymbol{W}\delta(t-\tau)$, $\mathcal{E}\{\boldsymbol{\nu}_k^\top \boldsymbol{\nu}_j\} = \boldsymbol{V}\delta(k-j)$). Assume that some of the elements of vector $\boldsymbol{y}(\tau)$ are measured at arbitrary discrete instants t_k, $k \in \mathbb{N}$ (at least one element $m_{i,k}$ is available at instant t_k). Let us define the $\mu_k \times n_m$ matrix $\boldsymbol{M}_k$ formed with the rows of an identity matrix that correspond to the position of the elements $m_{i,k}$ available at instant t_k, thus μ_k is the number of variables measured at instant t_k. N_k is the elapsed time between the last two consecutive sampling instants ($N_k = t_k - t_{k-1}$) and $d_{i,k}$ is the delay of an individual measurement $m_{i,k}$. Under these suppositions, the gain matrix $\boldsymbol{L}_k$ that minimizes the error variance (Kalman filter) of the algorithm (12) is calculated on-line by ([1,5])

$$
\begin{aligned}
\hat{\boldsymbol{Q}}_{k|k-1} &= \boldsymbol{A}^{N_k} \hat{\boldsymbol{Q}}_{k-1} (\boldsymbol{A}^\top)^{N_k} + \sum_{i=0}^{N_k-1} \boldsymbol{A}^i \boldsymbol{B}_w \boldsymbol{W} \boldsymbol{B}_w^\top (\boldsymbol{A}^i)^\top, \\
\boldsymbol{L}_k &= \hat{\boldsymbol{Q}}_{k|k-1} (\boldsymbol{M}_k \boldsymbol{C}_{d,k})^\top \Big(\boldsymbol{M}_k \left(\boldsymbol{C}_{d,k} \hat{\boldsymbol{Q}}_{k|k-1} \boldsymbol{C}_{d,k}^\top + \boldsymbol{V} + \boldsymbol{W}_d \right) \boldsymbol{M}_k^\top \Big)^{-1} \boldsymbol{M}_k \\
\hat{\boldsymbol{Q}}_k &= (\boldsymbol{I} - \boldsymbol{L}_k \boldsymbol{C}_{d,k}) \hat{\boldsymbol{Q}}_{k|k-1}
\end{aligned}
$$

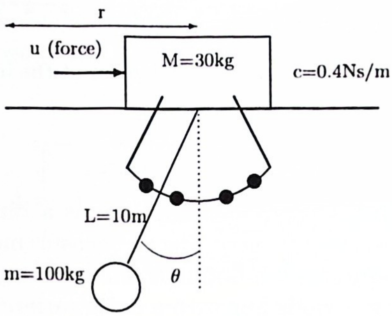
being

$$
\boldsymbol{C}_{d,k} = \begin{bmatrix} c_1 \boldsymbol{A}^{-d_{1,k}} \\ \vdots \\ c_{n_m} \boldsymbol{A}^{-d_{n_m,k}} \end{bmatrix}_{n_m \times n}, \quad \boldsymbol{W}_d = \underbrace{\begin{bmatrix} c_1 \boldsymbol{A}^{-d_{1,k}} \boldsymbol{\Lambda}(d_{1,k}) \\ \vdots \\ c_{n_m} \boldsymbol{A}^{-d_{n_m,k}} \boldsymbol{\Lambda}(d_{n_m,k}) \end{bmatrix}}_{\star} \begin{bmatrix} \boldsymbol{W} & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{W} \end{bmatrix} [\star]^\top
$$

If the measurements are available periodically and the delay in each sensor is constant, an stationary solution of the above equations can be obtained and then, the predictor gains can be calculated off-line and applied as an scheduled gain predictor (as in the solution in the LMI based approach).

5 Application example

Consider the system shown in the figure 3. The input of the system is the force that is assumed to be updated by a controller at a constant rate of $T = 0.5s$. The measured signals are θ and r, but they are assumed to be measured by binary sensors that produce a digital pulse when the signal reaches some predefined values. A low cost encoder produces a pulse every 1 m for the r signal. The θ signal is assumed to produce a pulse at values $\{-0.07\ -0.02\ 0.02\ 0.07\}$. As a consequence there are measurements of both θ and r at random instants that are asynchronous with the input update. The linearized model of this system (see [10]) is defined by the shown matrices, where $\boldsymbol{B_w} = \boldsymbol{B_c}$. The output variable



$$\boldsymbol{A_c} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{c}{M} & -\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{c}{ML} & -\frac{(M+m)g}{ML} & 0 \end{bmatrix}, \quad \boldsymbol{B_c} = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{ML} \end{bmatrix},$$

$$C_y = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,0\,1\,0 \end{bmatrix}^T$$

Fig. 3. Crane with binary sensors.

of interest is assumed to be θ, and hence $C_z = [0\ 0\ 1\ 0]$. The objective of the predictor is then to estimate the value of θ at the synchronous instants (i.e. to obtain $\hat{\theta}[t]$). For the simulation, the input is generated as a band limited white noise of period 1 sec. and power 100, that is contaminated by another band limited white noise of power 2 and period 0.2 sec. The measurements are contaminated by additive random normal noise of variance $0.01^2 rad^2$ and $0.1^2 m^2$ respectively. A Kalman filter is designed taking first the matrices $\boldsymbol{V} = [0.1^2\ 0; 0\ 0.01^2]$ and $\boldsymbol{W} = 4$. In the figure 5 the true and the estimated states at the measurement instants are shown, as well as the true output ($\theta[t]$) and the predicted one ($\hat{\theta}[t]$) at the input updating instants. The average quadratic prediction error is $1.357 \cdot 10^{-5}$.

If the $\boldsymbol{V}$ and $\boldsymbol{W}$ matrices are not exactly known, the behavior is not optimal, but the predictor still works well. As an example, if the matrices used in the predictor equation are $\boldsymbol{V} = [0.1^2\ 0; 0\ 0.02^2]$ and $\boldsymbol{W} = 2$, the quadratic prediction error average is $1.94 \cdot 10^{-5}$, and with $\boldsymbol{V} = [0.1^2\ 0; 0\ 0.005^2]$ and $\boldsymbol{W} = 8$, the quadratic prediction error average is $2.52 \cdot 10^{-5}$.

Consider now the case when the output of the system is measured by 2 sensors with different precision and measurement availability instants. This idea is covered by the general model taking $C_y = [C_z^\top\ C_z^\top]^\top$ and $\boldsymbol{V} = [v_1\ v_{12}\ ;\ v_{12}\ v_2]$.

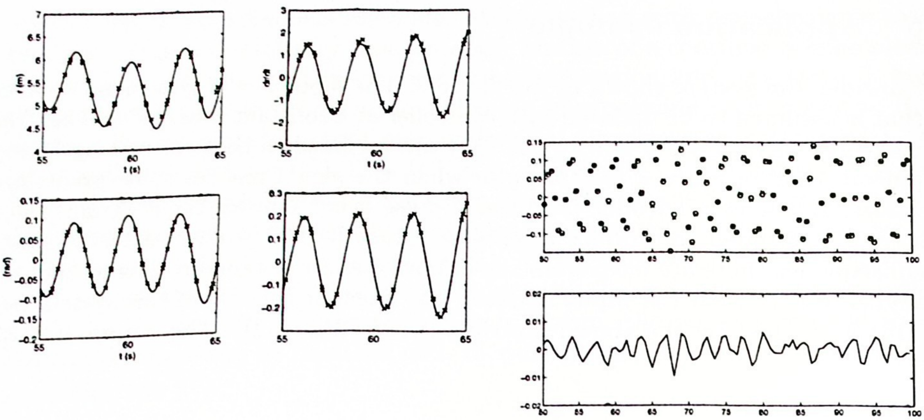


Fig. 4. True (- left, o right) and estimated (x) output, and prediction error at the input updating instants (tT).

Assume now that only θ is measured with 2 sensors. The first one is a binary sensor fixed on position $\theta = 0$ that gives scarce but very precise measurements (null noise variance assumed). The second sensor is a continuous sensor that gives a measurement every $N = 2$ input updating periods but with a noise variance of 0.006^2 (hence $\boldsymbol{V} = [0.006^2\ 0; 0\ 0]$). With the same input conditions as before the average quadratic prediction error is $4.35 \cdot 10^{-6}$. In order to compare the sensor fusion effect of the predictor, the same simulation is carried out assuming that only the second sensor is available, obtaining an error of $10.5 \cdot 10^{-6}$. If only the first sensor is used the resulting error is $10.7 \cdot 10^{-6}$. In the figure 5 the prediction errors are shown.

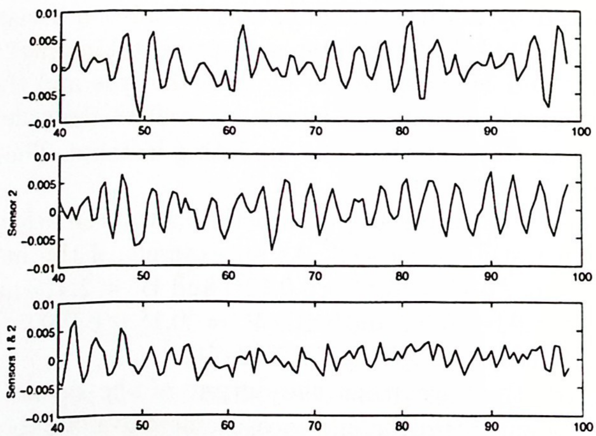


Fig. 5. Error with different sensor measurements.

Consider now a control period of $T = 0.2$ s. Assume that the angle and horizontal position are measured synchronously with the input update every 2 or 4 seconds ($N = \{10, 20\}$), and assume also that the signal transmission takes 0.4 or 0.8 seconds ($d_i = \{2, 4\}$) because they are accessed through a shared network. It is assumed that r and θ are measured with zero mean white noise sensors with typical deviation $\sigma_r = 0.01$ m and $\sigma_\theta = 0.001$ rad. In this scenario, both $\mathcal{H}_\infty$ and *Kalman* predictor are applied in order to implement a closed loop control using the state estimation. A disturbance $v(\tau) = 0.2,\ \tau > 200$s has been applied. Figure 6 show the results with $\mathcal{H}_\infty$ and Kalman filter approaches. Before the constant disturbance, the performance with $\mathcal{H}_\infty$ and Kalman filter approaches is similar. With the disturbance, the $\mathcal{H}_\infty$ approach leads to a prediction error characterized with $\sigma_r = 0.048$, while the Kalman filter leads to $\sigma_r = 0.078$. An explanation to this fact is that the Kalman filter is optimum only if the disturbances are white noise with zero mean. On the other hand, the computational cost of the Kalman filter is 8 times the $\mathcal{H}_\infty$ predictor.

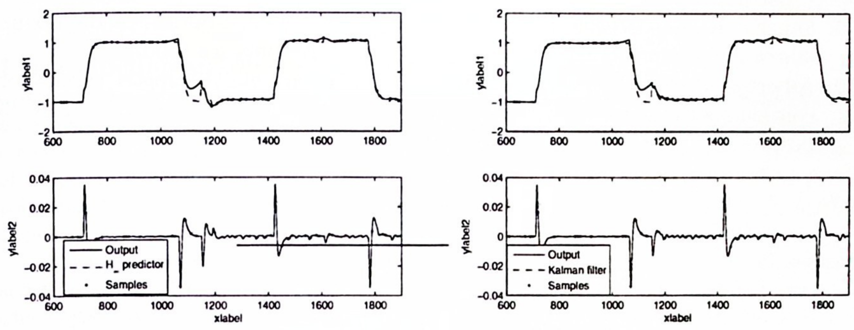


Fig. 6. Crane Control with different virtual sensors.

6 Conclusions

A virtual sensor has been defined as the intelligent device that estimates the desired outputs of a process using the information of the model, the known control inputs, and the accessible outputs measured with different sensors with different availability, noise and delays. Two different estimation algorithms that address sensor availability, scarce measurements and delays as a whole have been proposed (continuous and discrete one). Different techniques have been proposed for different sampling scenarios: pole placement (for periodic sampling), LMI based approach (when the different sampling scenarios are known and finite) and Kalman filter based techniques (for the general case with white noise disturbances). The Kalman filter needs an on-line calculation of the predictor gain, while the other techniques calculate the gain off-line, leading to a lower computational cost predictor.

References

1. Sanchis, R., Albertos, P.: Virtual sensors: model based output prediction. First EurAsia Conference on Advances in Information and Communication Technology (2002)
2. Gustafsson, F., Drevo, M., Forssell, U., Lofgren, M., Persson, N., Quicklund, H.: Virtual sensors of tire pressure and roadfriction. Society of Automotive Engineers **2001-01-0796** (2001)
3. Long, T., Hanzevack, E., Bynum, W.: Sensor fusion and failure detection using virtual sensors. Proceedings of the Arnencan Control Conference (1999)
4. Albertos, P., Goodwin, G.: Virtual sensors for control applications. Annual Reviews in Control **26** (2002) 101–112
5. Peñarrocha, I.: Virtual Sensors for processes with scarce and delayed measurements. PhD Thesis, Polythecnic University of Valencia (2006)
6. Astrom, K., Wittenmark, B.: Computer controlled systems. Prentice Hall (1997)
7. Cuenca, A., Salt, J., Albertos, P.: Implementation of algebraic controllers for non-conventional sampled-data systems. Real-Time Systems **35**(1) (2007) 59–89
8. Peñarrocha, I., Sanchis, R., Albertos, P.: Design of low cost virtual sensors. IEEE - Instrumentation and Measurement Technology Conference (IMTC) (2006)
9. Sanchis, R., Peñarrocha, I., Albertos, P.: Design of robust output predictors under scarce measurements with time-varying delays. Automatica (2007)
10. Albertos, P., Sanchis, R., Sala, A.: Fault detection via parameter estimation in continuous-time systems with random sampling (scarce measurements). IFAC Symposium on Low Cost Automation. Buenos Aires (1995)