The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

On-line estimation of time varying capture delay for vision-based vibration control of flexible manipulators deployed in hostile environments

Gregory Dubus

Abstract— **Thanks to recent advances in machine vision technology, vibration control of flexible manipulators using cameras have recently led to very promising results. However, in methods such as on-line sinusoidal regression [1], synchronization between the measured signal and the physical vibrational phenomenon is critical to properly damp vibrations. This paper comes back on the capture delay estimation as done in previous works and highlights the limitations of such a method. Then a new method using a synchronization sensor is proposed. Based on a cross-correlation technique, a novel algorithm for effective time delay estimation is derived. Experimental validation demonstrates that this method yields improved delay estimation which could be beneficial to the vibration damping.**

I. Introduction

In an attempt to push the limits of human knowledge, physicists sometimes give birth to devices or environments that are harmful for human beings. In search for new energy sources, the development of fusion energy is no exception. Indeed, whether in JET or in the upcoming ITER, neutron irradiation does not allow direct human access inside these experimental reactors. Therefore the in-vessel plasma facing components have to be inspected and maintained remotely.

But harsh conditions for humans are hard for the equipment as well. Especially radiations. Many mechanical sensors, particularly accelerometers, are very sensitive to radiation [2]. Their signal level and the noise level are affected. On the one hand, it is well known that the performance of integrated circuits degrades when exposed to low doses of ionizing radiation. On the other hand, exposing just the mechanical part of MEMS sensors to protons and heavy ions causes large changes in the outputs, attributed to charge generated by the ions and trapped in dielectric layers below the moveable mass in the case of accelerometers [3].

Short of costly developments in shielding, these constraints limit the use of dedicated electronics to deal with advanced control issues. However, in such applications, rad-hardened vision devices are inevitably used to provide real-time visual feedback to the operators. Thus the main idea behind our works [1, 4] is to consider these vision processes as full sensors and not only as plain visual feedbacks.

Moreover, on top of being a radioactive environment, a reactor like ITER is a huge and complex structure. Due to the size and the arduous accessibility of the reactor, the robotic arms designed for its maintenance will have to be long-reach arms. The main difficulties when positioning such equipment result from the vibrations due to its high flexibility.

Consequently they need the integration of appropriate compensation schemes to complete the tasks within the requirements whatever stimulates the structural modes: a critical trajectory imposed by the operator, an interaction with the environment, or internal unmodelled dynamics from carried processes. Input shaping techniques [5, 6] are very efficient to avoid critical trajectories by adjusting the actuators input in such a way that the natural modes are not stimulated. Considering the two other origins, the arm vibrational behaviour cannot be foreseen and it needs to be damped as soon as it occurs.

In previous works, vibrations estimation and control using cameras led to very promising results [1, 4, 7–9]. The main advantage of this approach is to sidestep problems related to the use of noisy or biased signals from accelerometers or strain gauges. On the flipside, vision devices have the disadvantage of a long processing time, leading not only to delayed measurements but also to low update rates [10].

The most common method to deal with delays in a control system is to decrease the servo gains to increase the damping, thus making the system more robust in the presence of time delays. However, some approaches are not based on time-delay robust controllers but on the accurate estimation of the delay itself, to reconstruct the quantity to control.

In [7], the controller design is split into two separate problems by using a composite control technique: a fast feedback stabilizes the oscillatory dynamics while a slow feedback ensures that the actual image asymptotically reaches the desired one. However, to implement the fast feedback, a Kalman filter is considered and fed by strain gauges measurements without considering high dynamics data coming from the camera. Obviously the problem of damping out the vibrations does not suffer from either delay or low rate of visual data.

Going further, [8] takes into account camera data in order to improve the capability of the system to damp vibrations. A Kalman filter is used to fuse the measurements coming from the different sensors, to improve the signal to noise ratio. Experimental validation shows that considering both strain gauges and camera yields smaller residual tip vibrations. However the authors do not explain how these desynchronized signals are fused together in the Kalman observer.

In [4, 9], only visual data are used to estimate both

This study follows-up the work carried out in the framework of the PREFIT project, supported by the EC as a Euratom Fusion Training Scheme.

G. Dubus (gregory.dubus@f4e.europa.eu) is now with F4E, Josep Pla 2, Torres Diagonal Litoral - Building B3, 08019 Barcelona, Spain

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

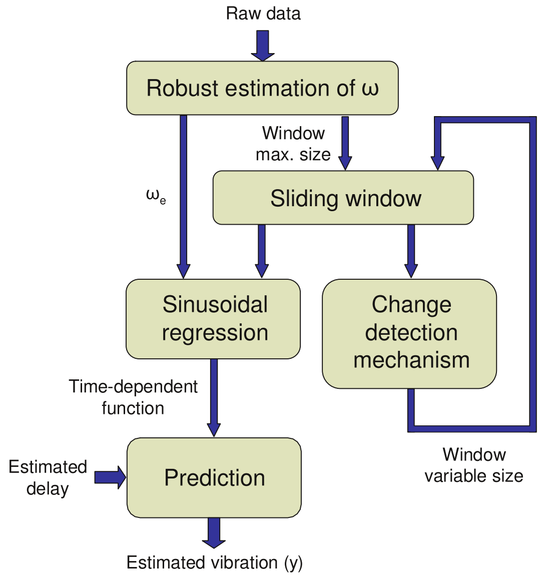


Fig. 1. Principle of the on-line sine regression

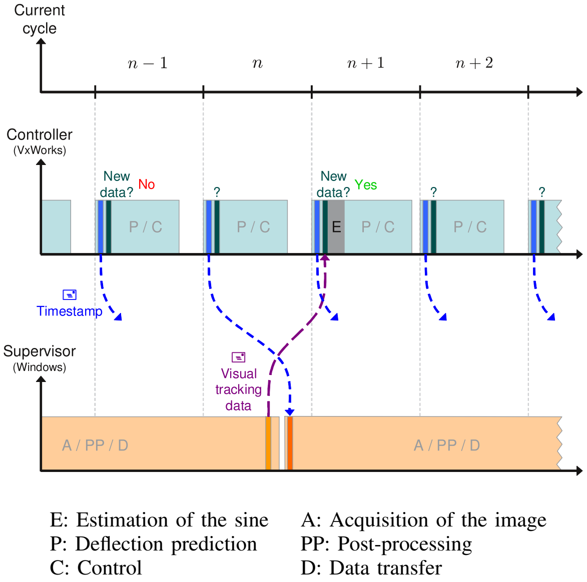


Fig. 2. Time diagram of communication between the real-time manipulator controller and the non real-time supervisor running the camera driver

tip vibration and imposed movement. To that purpose, a modified two-timescale Kalman filter takes into account the delay due to the image processing time. A delay compensator extrapolates the measured output to the present time using past and present estimates. If in [9] the delay is assumed constant, [4] considers a variable delay estimated on the basis of timestamps.

To reconstruct the vibration from visual data regardless its origin, [1] considers on-line sinusoidal regression instead of Kalman filtering. Here again, the current deflection is predicted from delayed visual data using timestamps exchange.

In methods such as in [1, 4], ensuring robustness of the controller towards time delays is not sufficient. An accurate on-line estimation of the delay between physical phenomenon and measured signal is needed to properly reject the vibration. Hitherto the image delay was estimated by exchanging timestamps between real-time high sampling rate controller and the non real-time supervisor whose sampling rate is aligned to the camera frame rate. But, if this method yields quite satisfying results, it does not take into account the uncertainty on the camera exposure time. By doing so, it is not unexpected to improve the vibration reconstruction.

Consequently the alternative method described in this paper consists in using a secondary synchronous - but potentially affected by radiations - sensor to estimate properly the delay and then to enable the correct re-synchronization of the vibration measurement with the physical phenomenon.

In other words this paper proposes, on the one hand, to use clean but delayed visual data to properly estimate the tip displacement and, on the other hand, to use noisy but synchronous inertial data to readjust these visual data in time.

The primary contribution of this work is to propose an on-line delay estimator. It is based on a cross-correlation technique that explicitly computes the time-delay between the two above-cited signals. Since our cross-correlation function can partially be computed recursively, the computational load of the proposed algorithm is limited.

The outline of this paper, which intentionally puts the emphasis on implementation considerations, is organised as follows. Section II comes back on the delay estimation as it is performed in [1] and highlights its main limitation, in the light of a short reminder on visual sensors basics. Then section III describes the proposed method to on-line estimate the capture delay. (It is based on the use of Farrow structures well known to signal processing engineers.) At last, section IV validates the ability of this alternative algorithm to estimate a time-varying capture delay with good accuracy.

II. Limitation of the delay estimation based on timestamps exchange

An accurate delay compensation is required between the physical vibrational phenomenon and the measured signal. Indeed signal synchronization is critical to proper vibration suppression. In this section, a description of the delay estimation as done in [1] is given. Then, in the light of a short reminder on visual sensors basics, the main limitation of this method is highlighted.

A. Delay estimation based on timestamps exchange

In [1] an all-in-one method has been proposed to solve the problem of vibration suppression by using visual features without any markers nor a-priori knowledge on the environment. Thanks to the camera mounted in an eye-in-hand configuration [11], the tip displacement induced by the oscillations is estimated with respect to the static environment. To that purpose the Lucas-Kanade-Tomasi (KLT) feature tracking algorithm [12] is used to extract and track features from the camera images. Then a Tukey M-estimator rejects outliers possibly resulting from the extraction noise and gives a robust estimation of the environment overall displacement seen by the camera. Then one can deduce the speed of the camera in the static environment basis. Figure 1 reminds the principle of the vibration estimator implemented in [1] to obtain a robust prediction of the vibration to be rejected. In this algorithm, an on-line sinusoidal regression is performed over a sliding window to analytically identify the measured vibration. Then, on the basis of the identified function and the estimated capture delay, a prediction step estimates the vibration to reject at the present time.

As most of visual servo control algorithms, this method is implemented astride two systems:

- a controller running a real-time application as fast as needed to perform a stable and quality control (constant refresh rate τ_c in the order of 1 ms);
- a supervisor computer running a non-real-time OS and managing the image acquisition (varying refresh rate τ_s in the order of 14–16 ms)

The two computers can exchange data via UDP.

One simple method to estimate the capture delay relies on timestamps. During operations, timestamps are exchanged between the real-time high sampling rate controller and the non real-time supervisor whose sampling rate is aligned to the camera framerate. This principle is illustrated by Fig.2

Each controller cycle begins by the current timestamp n_c being sent to the supervisor. This one is stored in a buffer and is used only if the application in the supervisor side asks for it. Otherwise the buffer is overwritten at the next cycle. Following this timestamp sending, a second buffer is read to determine if new visual data is available. If it is, a regression is performed. Otherwise the controller directly completes the prediction and control steps.

On the other side, each supervisor cycle begins by reading the timestamp buffer. Consequently one knows at what time the image capture begins with an accuracy in the order of 1 ms. Let us call it initial timestamp n_i. At the end of the supervisor cycle, both visual data and this initial timestamp feeds the data buffer. Therefore, when the controller reads this buffer, both current timestamp n_c and initial timestamp n_i are available. The capture delay is computed as the difference of these two times:

$$\Delta = (n_c - n_i)\ \tau_c \tag{1}$$

B. Basics on camera capture

To properly justify why the above method is not absolutely satisfying, let's first remind some basics on camera capture.

Let's consider either a CDD or a CMOS sensor, that equip most of digital cameras nowadays. An image is recorded in this sensor in three phases: reset of the pixel rows to be exposed, exposure of pixel rows and sensor readout. To do so, several operational modes exist.

In triggered operation, the sensor is on stand-by and exposes one image immediately after the occurrence of a trigger event. Consequently exposure and image readout are performed sequentially, and the achievable framerate directly depends on the exposure time.

In freerun mode, the camera sensor internally exposes one image after another at the set framerate. This time, exposure and readout are performed simultaneously, enabling the maximum camera framerate to be achieved.

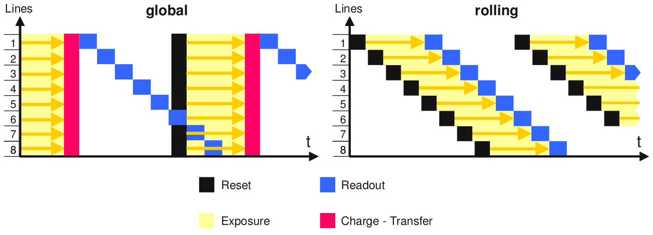


Fig. 3. Schematic of the global & rolling shutter methods in freerun mode

However, the sensor cells must not be exposed during the readout process. As a result camera sensors use mechanical or electronic shutters. Depending on the sensor type, either the rolling or the global shutter method is used.

On a global shutter sensor (left part of Fig.3), all pixel rows are reset and then exposed simultaneously. At the end of the exposure, all rows are synchronously moved to a darkened area of the sensor. The pixels are then read out row by row. Exposing all pixels at the same time has the advantage that fast-moving objects can be captured without geometric distortions.

With the rolling shutter method (see right part of Fig.3), the pixel rows are reset and exposed one after the other. At the end of the exposure, the lines are read out sequentially. Unfortunately it results in a time delay between the exposure of the first and the last rows, and captured images of moving objects are partially distorted. Consequently the only advantage of such sensors is their reduced price.

For the reader information, we chose for our application a CMOS sensor with electronic global shutter and made it run in freemode. This choice enabled framerates up to 60-70fps.

C. On the limitation of such a method

In freerun mode the exposure time is usually set to the reciprocal value of the framerate. As mentioned above, let's consider that our camera achieves a framerate of 60-70fps. The exposure time is close to 15 ms and there is no way of knowing when, along these 15 ms, our image is taken. In other words the capture instant is uncertain in the range of ± 7 controller cycle and cannot be accurately computed on the basis of the initial timestamp, according to $n_i \times \tau_c$.

For most of visual servoing application, this is not a problem as main time constants are far larger. In our case, where the goal is to estimate an oscillation whose frequency is in the order of 2-3Hz, a wrong estimation of this delay in the order of 7 controller samples would induce an error of 11.4% of the signal amplitude. In the worst situation, an error in the delay estimation of 15 controller samples would induce an error of 24.4% of the signal amplitude.

Still, what made the on-line vibration estimation proposed in [1] efficient yet is the fact that the sinusoidal function parameters were identified over a sliding window of size N ($3 \leq N \leq 20$). In the light of what has just been written, it is expected that the vibration damping could benefit from a better estimation of the image capture delay. This could be done using a synchronization sensor for example.

III. Delay estimation using a synchronization sensor and cross-correlation

In this section, an alternative method for estimating the capture delay is described. It consists in using a secondary sensor, synchronous but prone to noise due to radiation, to synchronize the delayed visual data with the physical oscillation. The proposed approach to estimate the capture delay is based on the concept of cross-correlation function.

For two periodic signals $x(n)$ and $y(n)$ having the same period of N samples, the cross-correlation is defined as:

$$\begin{aligned} \mathcal{C}_{xy}(m) &= \frac{1}{N}\sum_{n=1}^{N} x(n)y(n-m) \\ &= \frac{1}{N}\sum_{n=1}^{N} x(n+m)y(n) \end{aligned} \tag{2}$$

This correlation function also has a period of N samples.

Let's consider now a periodic signal $z(n)$ and two derived signals $x(n)$ and $y(n)$. $x(n)$ consists of the signal $z(n)$ plus an additive white gaussian noise $v(n)$, and $y(n)$ corresponds to the signal $z(n)$ delayed by n_0 samples:

$$x(n) = z(n) + v(n) \tag{3}$$

$$y(n) = z(n-n_0) \tag{4}$$

Now let's look at the cross-correlation between $y(n)$ and $x(n)$ during M samples (M is much greater than N):

$$\mathcal{C}_{yx}(m) = \frac{1}{M}\sum_{n=1}^{M} y(n)x(n-m) \tag{5}$$

By replacing the expressions of $x(n)$ and $y(n)$ into (5), we obtain:

$$\mathcal{C}_{yx}(m) = \frac{1}{M}\sum_{n=1}^{M} z(n-n_0)[z(n-m)+v(n-m)] \tag{6}$$

Developing this relation yields

$$\begin{aligned} \mathcal{C}_{yx}(m) =& \frac{1}{M}\sum_{n=1}^{M} z(n-n_0)z(n-m) \\ &+ \frac{1}{M}\sum_{n=1}^{M} z(n-n_0)v(n-m) \end{aligned} \tag{7}$$

which can be re-written

$$\begin{aligned} \mathcal{C}_{yx}(m) =& \frac{1}{M}\sum_{n=1}^{M} z(n-n_0+m)z(n) \\ &+ \frac{1}{M}\sum_{n=1}^{M} z(n-n_0+m)v(n) \end{aligned} \tag{8}$$

Finally, the cross-correlation can be expressed by:

$$\mathcal{C}_{yx}(m) = \mathcal{C}_{zz}(m-n_0) + \mathcal{C}_{zv}(m-n_0) \tag{9}$$

This result shows that the cross-correlation consists of two terms: the auto-correlation $\mathcal{C}_{zz}(m-n_0)$ of the periodic signal shifted in time, and the cross-correlation $\mathcal{C}_{zv}(m-n_0)$ between the periodic signal $z(n)$ and the corrupting noise $v(n)$, also shifted in time. On the one hand, due to the random nature of noise and the independence of the signal and noise, $\mathcal{C}_{zv}(m-n_0)$ is usually rather small. On the other hand, $\mathcal{C}_{zz}(m-n_0)$ is larger. It is also periodic and has peaks at $m = n_0, N+n_0, 2N+n_0 ....$ Thus, by examining $\mathcal{C}_{yx}(m)$, we can estimate very easily the delay n_0.

Consequently, in order to estimate the capture delay, such a cross-correlation computation can be employed over a fixed-size sliding window on the signals coming from the noisy inertial sensor and the delayed visual data. The size of the window must be chosen large enough to include at least one period of the cross-correlation.

Since no new information is received between two set of visual data (i.e. for a duration of τ_s), the delay Δ is assumed to be constant during this period, and the same value of Δ can be used to predict the vibration until the next data reception from the supervisor. However, one can take advantage of this time to refine the estimation of Δ.

Nevertheless, a last difficulty must be overcome: the capture delay changes for each set of visual data, but the cross-correlation is computed over at least one period of the sinusoidal signals. Consequently, one must take more into consideration the P values received after the last supervisor refreshment than the previous ones.

To do so, let's re-write (5), explicitly considering the last P measured values:

$$\begin{aligned} \mathcal{C}_{yx}(m) &= \frac{1}{M}\sum_{n=1}^{M} y(n)x(n-m) \\ &= \frac{1}{M}\sum_{n=1}^{M-P} y(n)x(n-m) \\ &\qquad + \frac{1}{M}\sum_{n=M-P+1}^{M} y(n)x(n-m) \\ &= \frac{M-P}{M}\mathcal{C}_{yx}^{old}(m) + \frac{P}{M}\mathcal{C}_{yx}^{new}(m) \\ &= \mathcal{C}_{yx}^{old}(m) + \frac{P}{M}\left(\mathcal{C}_{yx}^{new}(m) - \mathcal{C}_{yx}^{old}(m)\right) \end{aligned} \tag{10}$$

where $\mathcal{C}_{yx}^{old}(m)$ represents the cross-correlation computed from the values before the last reception of visual data, and $\mathcal{C}_{yx}^{new}(m)$ represents the cross-correlation from the latest values. From (10) the new estimate value of $\mathcal{C}_{yx}(m)$ can be regarded as the old estimate $\mathcal{C}_{yx}^{old}(m)$ plus a correction term.

All along (10), all measured values have the same weight. In our case, as for any non-stationary system, recently measured values should be weighed more heavily. One method to achieve this is to introduce a forgetting factor ρ:

$$\begin{aligned} \mathcal{C}_{yx}(m) &= \mathcal{C}_{yx}^{old}(m) \\ &+ \frac{\rho P}{(1-\rho)(M-P)+\rho P}\left(\mathcal{C}_{yx}^{new}(m) - \mathcal{C}_{yx}^{old}(m)\right) \end{aligned} \tag{11}$$

ρ is such that $0 \leq \rho \leq 1$. Note that, for $\rho = \frac{1}{2}$, the whole data has the same weight and (11) is equivalent to (10).

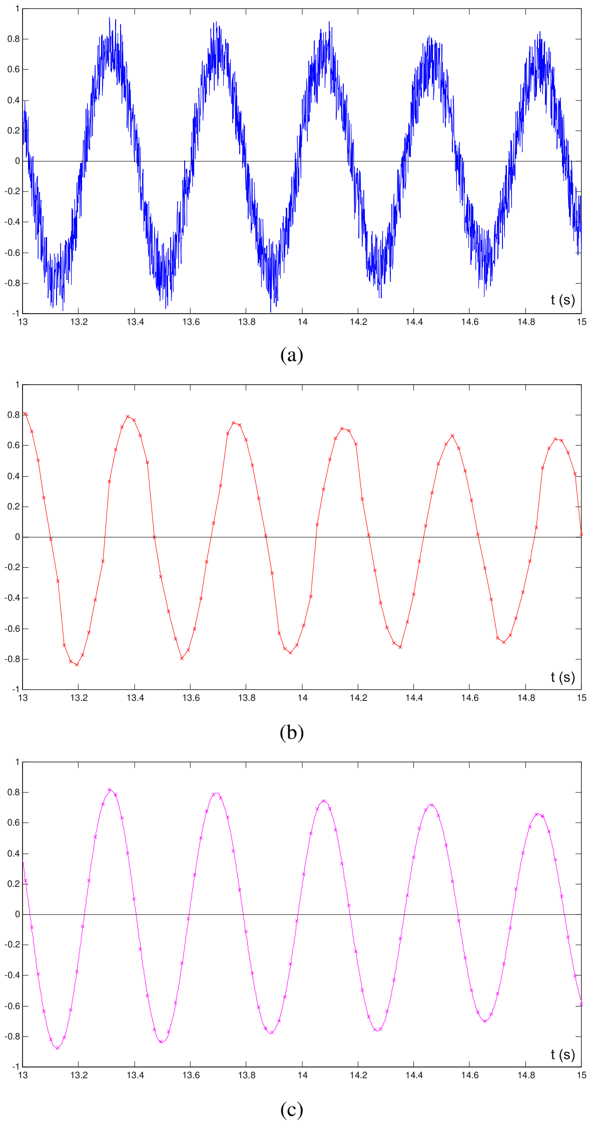
At the end of the day, the capture delay Δ is defined by $n_0 \times \tau_c$ with n_0 such as $\mathcal{C}_{yx}(n_0) = \max\left[\mathcal{C}_{yx}(m)\right]_{0\leq m\leq N}$.

IV. Experimental validation

The experimental validation is based on the data collected during the previous works carried out on the experimental mock-up which was accurately described in both [4] and [13]. For the needs of this experimentation, it has been equipped with a tip-mounted industrial camera IDS uEye UI-122xLE (resolution: 640×480) and a triple axis accelerometer LIS3LV02DQ. The controller runs on the real-time OS VxWorks at a sampling time of 1 ms. The overall vision-based application is based on the ViSP software [14] and runs at around 60-70Hz.

As in [1], we only aspire to damp the fundamental which is situated around $2.6Hz$ in the case of this representative experimental setup. Figure 4 illustrates the vibration reconstruction in the case when the capture delay is estimated as described in section III. Figure 4(a) and Figure 4(b) respectively depict the accelerometer signal and the visual data. Several comments can be made on these graphs. First, the accelerometer signal is clearly too noisy to perform a quality vibration rejection. All the more that this signal has been obtained from a non-irradiated sensor. Beside, the signal resulting from the visual tracking is far cleaner. However one can notice that an important delay (in the order of 1/10 s) exists between these two graphs. Moreover the fact this delay is variable mainly explains why the sinusoidal oscillation seen by the camera is so distorted. Figure 4(c) illustrates the ability of the proposed algorithm to properly predict a clean vibration measurement which is synchronized with the physical phenomenon. It is sampled at the controller sampling rate; crosses only indicate the visual data refreshment times.

The delay time variability is highlighted on Fig.5 and Fig.6 which show how the cross-correlation function between the accelerometer signal and visual data enables an accurate estimation of the capture delay. One can note that, on the selected period of time, the capture delay is comprised between 70 ms and 76 ms over the considered time interval. During the whole test, its mean value was $\bar{\Delta} = 74$ ms while its standard deviation was $\sigma_{\Delta} = 6.4$ ms. Such a result might partially be explained by the uncertainty on the capture instant as explained in section II-C. However one must keep in mind that the supervisor runs a non-real-time OS and this variability of the capture delay could be more probably due to unpredictable changes in the computed load.



(a)

(b)

(c)

Fig. 4. Available measurements when the capture delay is estimated thanks to a synchronization sensor: (a) accelerometer signal (normalized), (b) visual data (normalized), (c) predicted oscillation (normalized)

At last Fig. 7 compares the two methods described in sections II-A and III respectively. On the one hand, there is no big difference regarding the general shape of the predicted oscillations obtained with these two methods. In both cases, the sine regression algorithm does its job and the classic Pearson correlation coefficient (PMCC), respectively of 0.9981 and 0.99424, shows that the reconstructed signals are both very close to decreasing sinusoids. On the other hand, Fig.7(a) also highlights the fact that the deflection estimation is roughly synchronous with the accelerometer signal, i.e. the physical phenomenon, when the capture delay is computed thanks to the correlation method, which is clearly not the case when it is measured thanks to timestamps. Figure 7(b) depicts the errors between the predicted oscillations and the accelerometer signal which has been cleaned up offline with a zero-phase filter. On the selected period of time, which is quite reprentative of the whole measurements, the mean errors are in the order of 15.1% and 2.2% of the sine amplitude, with the timestamp method and the correlation method respectively. Moreover, using timestamps, the maximum error on the deflection estimation can add up to one third of the deflexion maximal amplitude, while it is around 5.7% with the correlation method. In other words, by yielding a better estmation of the camera capture delay, the proposed method enables an average reduction of the sine regression error in the order of 70% to 80%, which is clearly beneficial to the vibration rejection scheme.

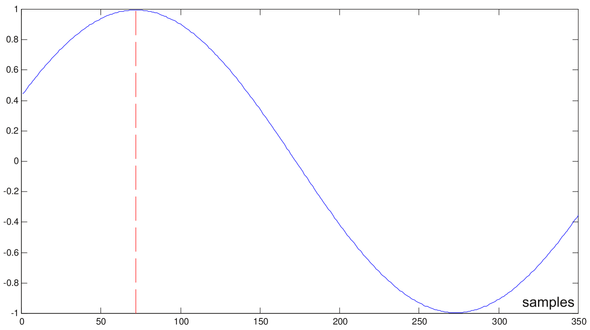


Fig. 5. Cross-correlation between the two signals at $t = 13s$

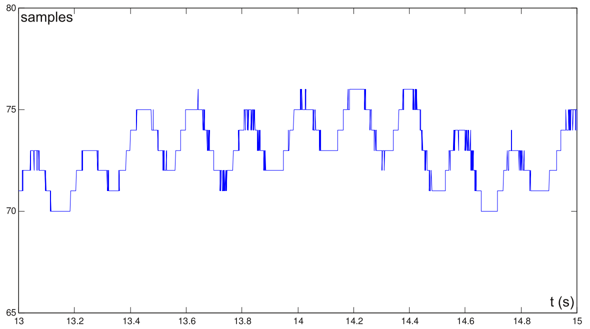


Fig. 6. Estimated capture delay

V. Conclusion

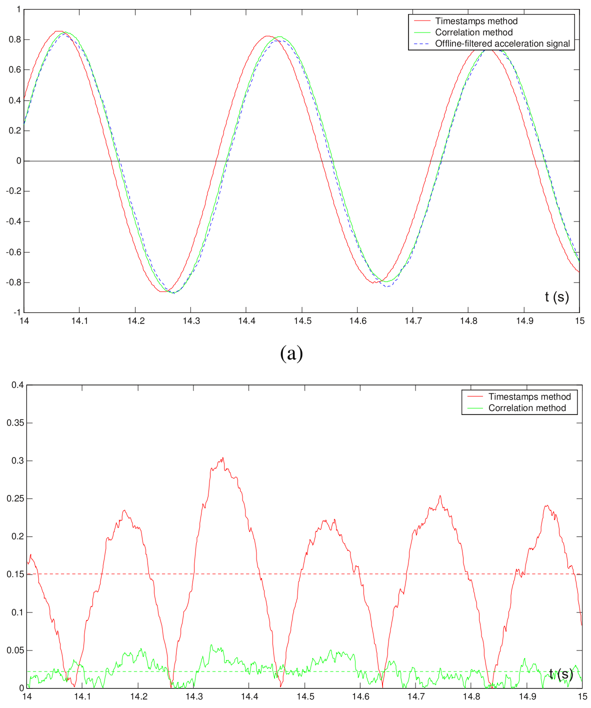
This paper suggests to use both clean but delayed visual data and noisy but synchronous inertial data to properly reconstruct the end-effector displacement of a flexible manipulator. Consequently an on-line algorithm to estimate time varying image processing delays has been described. It is based on a cross-correlation technique that explicitly computes the time-delay between the visual data and the output of a synchronization sensor. Because of its recursive formulation, the computational load of this algorithm is quite limited. Synchronization of the vibration measurement with the physical phenomenon using the proposed method is demonstrated through an experimental example. This algorithm provides a robust delay estimate in the presence of noise and is an improvement over the approach reported in [1] and [4].

One limitation of these works lies in the experiment not being performed in the field, where the accelerometer would be seriously affected by the environment. This could be bypassed by using an already irradiated accelerometer. However such an experimental setup would require an authorization from the competent nuclear regulatory authority which would be far beyond the framework of this study.

Beside the control of flexible manipulators, this method could be extended to the monitoring of any periodic phenomenon by a vision device.

References

[1] G. Dubus, O. David, and Y. Measson, “A vision-based method for estimating vibrations of a flexible arm using on-line sinusoidal regression,” in *Proc. of IEEE ICRA’10*, Anchorage USA, May 2010.



(a)

(b)

Fig. 7. Comparison between the two methods: (a) predicted oscillations (normalized), (b) error between the predicted oscillations and the physically synchronous signal from the accelerometer

[2] L. Houssay, “Robotics and radiation hardening in the nuclear industry,” Master’s thesis, University of Florida, 2000.

[3] A. Knudson, S. Buchner, P. McDonald, W. Stapor, A. Campbell, K. Grabowski, D. Knies, S. Lewis, and Y. Zhao, “The effects of radiation on MEMS accelerometers,” *IEEE Transactions on Nuclear Science*, vol. 43, pp. 3122–3126, dec 1996.

[4] G. Dubus, O. David, and Y. Measson, “Vibration control of a flexible arm for the iter maintenance using unknown visual features from inside the vessel,” in *Proc. of IEEE IROS’09*, Saint Louis USA, Oct. 2009.

[5] N. C. Singer and W. P. Seering, “Preshaping command inputs to reduce system vibration,” in *Artificial intelligence at MIT*, 1990, pp. 128–147.

[6] A. Khalid, W. Singhose, J. Huey, and et al., “Study of operator behavior and performance using an input-shaped bridge crane,” in *IEEE CCA*, 2004, pp. 759–764.

[7] L. Bascetta and P. Rocco, “Two-time scale visual servoing of eye-in-hand flexible manipulators,” *IEEE Trans Robot*, vol. 22, no. 4, pp. 818–830, 2006.

[8] ——, “End-point vibration sensing of planar flexible manipulators through visual servoing,” *Mechatronics*, vol. 16, no. 3-4, pp. 221–232, 2006.

[9] X. Jiang, Y. Yosuke, A. Konno, and M. Uchiyama, “Vibration suppression control of a flexible arm using image features of unknown objects,” in *Proc. of IROS*, 2008, pp. 3783–3788.

[10] M. C. Corke, P. I. Good, “Dynamic effects in high-performance visual servoing,” in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, Nice, May 1992, pp. 1838–1843.

[11] S. Hutchinson, G. D. Hager, and P. I. Corke, “A tutorial on visual servo control,” *IEEE Trans Robot Autom*, vol. 12, pp. 651–670, 1996.

[12] J. Shi and C. Tomasi, “Good features to track,” in *Proc. of CVPR*, 1994, pp. 593–600.

[13] T. Gagarina-Sasia, O. David, G. Dubus, and et al., “Remote handling dynamical modelling: assessment on a new approach to enhance positioning accuracy with heavy load manipulation,” *Fus. Eng. Des.*, vol. 83, no. 10-12, pp. 1856 – 1860, 2008.

[14] E. Marchand, F. Spindler, and F. Chaumette, “Visp for visual servoing: a generic software platform with a wide class of robot control skills,” in *IEEE Robotics and Automation Magazine*, Dec. 2005, pp. 40–52.