

Vibration control of a flexible arm for the ITER maintenance using unknown visual features from inside the vessel

Gregory Dubus, Olivier David and Yvan Measson

Abstract—In this paper we present a vibration control scheme for a long-reach inspection arm in an environment as constrained as a fusion reactor. The ultra-high vacuum, the high operating temperature, the significant residual magnetic field and the nuclear activation in ITER will prevent from considering common on-the-shelf dynamics sensors. Therefore we propose to use the already developed rad-hardened viewing tool to feed an oscillation observer with visual information. In our approach the visual data are extracted from a totally unknown environment. Particular attention is paid to obtain a robust algorithm. Experimental results validate the proposed strategy.

I. CONTEXT OF THE STUDY

A. Need for an inspection arm for the ITER maintenance

Population growth and steadily rising standards of living will keep demand for energy growing substantially in the years to come. All potential energy options must be kept open to ensure that responses are as appropriate as possible, both environmentally and economically. Thermonuclear fusion is one of these options.

The International Thermonuclear Experimental Reactor (ITER) is the next generation of experimental fusion reactors. It aims at demonstrating the scientific and technological feasibility of fusion energy. Inside the torus fusion reactions between Deuterium and Tritium isotopes produce high-energy neutron fluxes that irradiate the structure. Because of this neutron activation, which forbids direct human access inside the reactor, the in-vessel plasma facing components of ITER must be inspected and maintained remotely.

The In-Vessel Viewing System (IVVS) project, which is still mainly at the concept level, assumes that a long reach deployer equipped with a probe penetrates the ITER chamber to perform periodic inspections in a short time between two plasma shots. In that purpose the operation of the robot should be realised under ITER operating conditions i.e. ultra-high vacuum (10^{-6} Pa), high temperature ($120 - 200^\circ\text{C}$), significant residual magnetic field and nuclear activation.

B. A carrier demonstrator: the AIA

A feasibility analysis drove the design of the so called Articulated Inspection Arm (AIA) demonstrator [1], which is a 8m-long multi-link carrier with 5 identical modules made in titanium (see Fig. 1). Combination of elevation and rotation motions gives to the robot 8 DOF. With a payload up to 10 kg and a total weight of 150 kg this polyarticulated

arm can be introduced through a small port of 250 mm diameter. To cope with the high temperature requirements, all AIA articulations are actuated by specific electrotechnical components qualified up to 120°C in use and 200°C when switched off. However operating in a magnetic field and in a nuclear ambiance has not been fully considered yet.

Full remote control under blind conditions requires on line collision avoidance and monitoring. A challenging problem has been addressed to deal with the structural flexibilities of the robot which cause limited positioning accuracy performance. At this stage a quasi-static model-based control scheme has been developed [2] and will be implemented in 2010 after the flexible model calibration.

C. Prototypes of diagnostic tools

In parallel, various probe prototypes have been developed to equip such a long-reach carrier.

First of all a viewing system (shown in Fig. 1) has been designed to make close visual inspection of plasma facing components in Tore Supra with the AIA demonstrator (see inset in Fig. 1). All the camera inner electronic components (CCD sensor, zoom, LEDs,...) are actively cooled by Nitrogen gas to keep the operating temperature below 60°C .

The so called 3D Inspection System [3] is another plug tool especially designed for ITER which aims at performing sub-millimetric 3D images during maintenance procedure. In this process a coherent single mode laser is modulated and transmitted to the target via a rotating prism. When the scanning head is stable the measurement accuracy is up to 0.1mm at 1m. Unfortunately this value drastically decreases

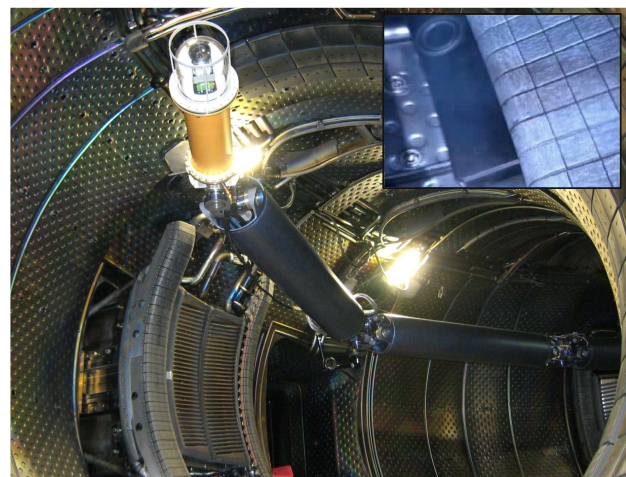


Fig. 1. Deployment inside TORE SUPRA vacuum vessel of the AIA equipped with the viewing system. Inset: visual feedback

This work, supported by the EC in the context of the Euratom Fusion Training Schemes, was carried out in the framework of the PREFIT contract.

G. Dubus, O. David and Y. Measson are with CEA, LIST, Interactive Robotics Unit, BP6 Fontenay-aux-Roses Cedex, F-92265 France. Corresponding author: gregory.dubus@cea.fr

when the scanning head is at the tip of a carrier since the rotating prism induces vibrations in the whole structure.

Other diagnostic tools are currently developed for tokamak in-vessel inspection, such as a leak detection system based on helium sniffing or a compact laser system for carbon codeposited layers characterizations or treatments [4].

D. Motivation

Man-in-the-loop technique would provide very useful help during inspection giving the operator the ability and flexibility to locate and examine unplanned targets. But vibrations due to the structure high flexibility are probably the main identified problem in such a master-slave mode. As a consequence it needs the integration of high level compensation schemes to complete the tasks within the requirements.

The stimulation of the structural modes can result from:

- a critical trajectory imposed by the operator
- a collision or interaction with the environment
- internal unmodelled dynamics (from carried processes, e.g. the rotating prism of the 3D Inspection System)

Input shaping techniques [5, 6] are very efficient to avoid critical trajectories by adjusting the input command to the actuators in such a way that the vibrational modes are not excited. Considering the two other origins, the vibrational behaviour of the arm cannot be foreseen and it needs to be damped as soon as the vibrations occur.

Unfortunately the ITER remote handling equipment will be subjected during a shutdown to a cumulated radiation dose in the order of several MGy. This constraint limits the use of dedicated on-the-shelf electronics such as accelerometers. Moreover the use of strain gauges suffer from the inherent high noise due to electromagnetic interferences.

The main idea behind this paper is to control the oscillatory behaviour of the flexible carrier without considering any extra sensor apart from its embedded vision process. The tip displacement induced by vibrations is estimated exploiting a simple physical model of the manipulator. Thanks to the camera mounted in an eye-in-hand configuration [7] this model is then readjusted using direct measurement of the oscillations of the tip with respect to the static environment.

This paper continues along the lines of several other works on the vibration control of flexible links using visual data [8–11]. [12] proved the feasibility of using a two-time scale Kalman filter to observe the arm vibrational behaviour. Yet in this work the tracking is based on the detection of a set of particular markers (red dots on a plain background). Even if no information is assumed to be known regarding their geometry, the problem is greatly simplified since the need for a robust tracking is seriously toned down. Our primary contribution is the implementation of a markerless tracker to determine the tip-camera velocity in an untrimmed environment. The addition of an M-estimator ensures the robustness of the algorithm. Moreover we look into the notion of on-line computation of the Image Jacobian and propose a modification to make such an estimator self-adjustable.

At the end of the day the all-in-one method we propose solves the problem of vibration suppression using visual

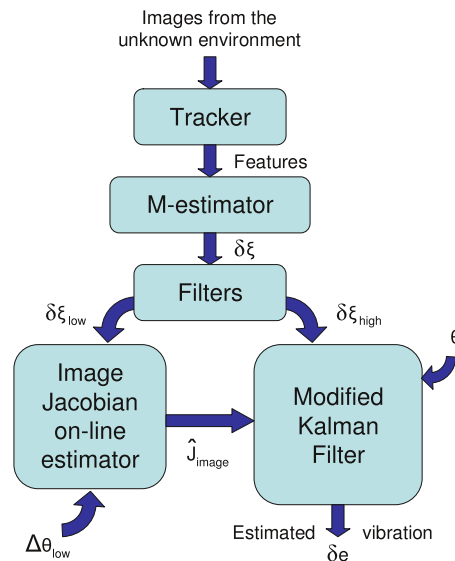


Fig. 2. Principle of the vibration estimator

features from a totally unknown and untrimmed static environment, which had surprisingly never been treated before. It is definitely an application paper putting the emphasis rather on the robustness and adaptiveness of the technical implementation than on new theoretical inputs.

This paper is organised as follows. After an overview of the context and constraints of the study in section I, section II presents the different components of the control scheme: the dynamic model of the flexible arm, the two-time scale Kalman filter, the robust tracker and the real-time Jacobian matrix estimator. The LQR controller designed in section III is experimentally validated in section IV, which is followed by concluding remarks in section V.

II. VISION BASED VIBRATION ESTIMATION

In this section we consider the problem of designing an on-line vibration estimator using a camera and without any knowledge of the environment (see Fig. 2). In the first step a tracker extracts and tracks features from the camera images. From this set of features an M-estimator rejects the outliers possibly resulting from the extraction noise and gives a robust estimation of the environment overall displacement seen by the camera. Afterwards this signal is filtered and its low-dynamic part is used to reconstruct the Image Jacobian matrix, which relates the motion of the environment in the 2D-image and the motion of camera in the 3D-world. In the last step of the proposed algorithm the on-line estimated Image Jacobian and the high-dynamic part of the image features displacement both feed a discrete time Kalman filter. Due to the long processing time of visual data the Kalman filter is modified to deal with delayed measurements.

A. System Equations

Let's consider a flexible planar arm consisting of n joints. It is modelled by lumped masses and massless springs. High order vibrational modes are neglected. By a standard procedure based on Lagrange equations two dynamic equations can be written as follows:

$$M_1(\theta, e)\ddot{\theta} + M_2(\theta, e)\ddot{e} + c_1(\theta, e, \dot{\theta}, \dot{e}) + g_1(\theta, e) = \tau \quad (1)$$

$$M_2^T(\theta, e)\ddot{\theta} + M_3(\theta, e)\ddot{e} + K_3e + c_2(\theta, e, \dot{\theta}, \dot{e}) + g_2(\theta, e) = 0 \quad (2)$$

where $\theta = [\theta_1 \dots \theta_n]^T$ is the vector of the n joint angles and $e = [e_1 \dots e_m]^T$ is the vector of the m deflections variables. M_1 , M_2 and M_3 are inertia matrices, K_3 is a stiffness matrix, c_1 and c_2 are the centrifugal and Coriolis torques and forces vectors, g_1 and g_2 are gravitational terms vectors, and τ represents the joints input torques.

Since this paper deals with relatively slow motions we assume that the action of c_1 and c_2 can be neglected. Moreover the influence of the deflections on the gravity terms is ignored. Thus the nonlinear dynamic model given by eqs. (1) and (2) can be approximated by a linear model around a given steady state position. Such an equilibrium is characterized by the following conditions:

$$g_1(\theta) = \tau_0 \quad (3)$$

$$K_3e_0 + g_2(\theta) = 0 \quad (4)$$

where e_0 is the static deflection for a given joint position θ and τ_0 is the constant torque that balances gravity.

Let's consider $\delta e = e - e_0$ and $\delta\tau = \tau - \tau_0$ as the deviations of e and τ from their respective static values. Subtracting eq. (3) from eq. (1) and eq. (4) from eq. (2) we obtain the following linearized model:

$$\underbrace{\begin{bmatrix} M_1 & M_2 \\ M_2^T & M_3 \end{bmatrix}}_M \underbrace{\begin{bmatrix} \dot{\theta} \\ \dot{\delta e} \end{bmatrix}}_K + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & K_3 \end{bmatrix}}_U \underbrace{\begin{bmatrix} \theta \\ \delta e \end{bmatrix}}_U = \underbrace{\begin{bmatrix} I \\ 0 \end{bmatrix}}_U \delta\tau \quad (5)$$

One can notice that the lower part of eq. (5) can be seen as the dominant equation for the vibrational behaviour and it can be expressed as the following state-space model:

$$\begin{bmatrix} \dot{\delta e} \\ \delta e \end{bmatrix} = \begin{bmatrix} 0 & I \\ -M_3^{-1}K_3 & 0 \end{bmatrix} \begin{bmatrix} \delta e \\ \delta e \end{bmatrix} + \begin{bmatrix} 0 \\ -M_3^{-1}M_2^T \end{bmatrix} \ddot{\theta} \quad (6)$$

Equation (6) fully describes the vibrational dynamics of our arm. We could also have chosen to include $\delta\tau$ in this model by replacing $\ddot{\theta}$ in the lower part of eq. (5) by its expression in the upper part. As we will see below our choice is mainly motivated by the ease of measuring $\ddot{\theta}$ as the model input rather than $\delta\tau$. Being given this model an observer of the deflection deviation δe can be built provided that we add to eq. (6) a measurement equation. In our case the measurement is made of visual data taken from features points of the static environment projected in the image plane.

The velocity $\dot{\xi}$ of these 2D features can be related to the tip camera velocity \dot{P} using the Image Jacobian concept [9]:

$$\dot{\xi} = \frac{\partial \xi}{\partial P} \dot{P} = J_{image} \dot{P} \quad (7)$$

We will see in section II-F how to efficiently evaluate J_{image} . Moreover the velocity of the tip camera \dot{P} depends on both the rigid motion $\dot{\theta}$ and the elastic motion \dot{e} of the arm:

$$\dot{P} = \frac{\partial P}{\partial \theta} \dot{\theta} + \frac{\partial P}{\partial e} \dot{e} = J_\theta \dot{\theta} + J_e \dot{e} \quad (8)$$

From eqs. (7) and (8) we deduce:

$$\dot{\xi} = J_{image} J_\theta \dot{\theta} + J_{image} J_e \dot{e} \quad (9)$$

where J_θ and J_e refer to the Jacobian matrices of the end-point with respect to the articular positions and the deflexion variables respectively.

Now let's consider that the velocity $\dot{\xi}$ of the features can be split in a low dynamics component $\dot{\xi}_{low}$ and a high dynamics component $\dot{\xi}_{high}$. It is plausible to assume that $\dot{\xi}_{low}$ mainly results from the articular movement whereas $\dot{\xi}_{high}$ mainly results from the vibration. Therefore, by considering the linear approximation of ξ_{high} for a deflection e around e_0 :

$$\xi_{high} \approx \xi_{high,0} + J_{image} J_e (e - e_0) \quad (10)$$

which can be written according to the state vector of eq. (6):

$$\delta \xi_{high} \approx \begin{bmatrix} J_{image} J_e & 0 \end{bmatrix} \begin{bmatrix} \delta e \\ \delta \dot{e} \end{bmatrix} \quad (11)$$

As a consequence the process to be estimated can be expressed as the following continuous state-space model:

$$\dot{x}_1(t) = A_1 x_1(t) + B_1 \ddot{\theta} + w_1(t) \quad (12)$$

$$z_1(t) = C_1 x_1(t) + v_1(t) \quad (13)$$

where $x_1 = [\delta e \ \delta \dot{e}]^T$ and $z_1 = \delta \xi_{high}$. w_1 and v_1 are the usual white Gaussian noises, with respective covariances Q_1 and R_1 . Matrices A_1 , B_1 and C_1 are defined by:

$$A_1 = \begin{bmatrix} 0 & I \\ -M_3^{-1}K_3 & 0 \end{bmatrix} \quad B_1 = \begin{bmatrix} 0 \\ -M_3^{-1}M_2^T \end{bmatrix} \\ C_1 = [J_{image} J_e \ 0]$$

From there a linear state observer can be designed using a discrete steady-state Kalman filter whose gains are optimized for the above assumed noises.

B. Incorporation of the acceleration estimation

The robust vibration estimation algorithm in this paper is based on the dynamic model (12) and therefore assumes that the joints accelerations $\ddot{\theta}$ are known exactly. Unfortunately, because of the very constrained environment, the use of accelerometers is made impossible. As it is commonly the case in robotics, only the joint positions θ are available from optical encoders and the joint velocities $\dot{\theta}$ and accelerations $\ddot{\theta}$ must be estimated from these discrete-time quantized signals. The classic Euler approximation combined with low-pass filtering can be considered but it does not yield good results at high sampling rates. Moreover such filtering often smoothes excessively the measurement of transient dynamics which stimulate the flexible arm but could not be taken into account in the vibration estimator. As a consequence stochastic methods are preferred to deterministic ones in order to reduce the variances while reconstructing the accelerations. A more sophisticated method to estimate the velocities is then by using an optimized Kalman filter for the following continuous model [13]:

$$\dot{x}_2(t) = A_2 x_2(t) + \Gamma_2 w_2(t) \quad (14)$$

$$z_2(t) = C_2 x_2(t) + v_2(t) \quad (15)$$

where $x_2 = [\theta \ \dot{\theta} \ \ddot{\theta}]^T$. The white, zero mean Gaussian noise $w_2(t)$ is a surrogate for the jerk $\ddot{\ddot{\theta}}$, which has to be

considered as a wide-band signal for better reconstruction. Its covariance Q_2 may be regarded as a filter parameter to be adjusted. $v_2(t)$ represents the quantization error, assumed white, zero-mean, and of constant variance R_2 . Matrices A_2 , C_2 and Γ_2 are defined by:

$$A_2 = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} \quad C_2 = [I \quad 0 \quad 0] \quad \Gamma_2 = \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}$$

As a consequence we propose to merge systems (12–13) and (14–15), by incorporating the joint position, velocity and acceleration as a part of the state and reconstructing the latter from the encoder measurement:

$$\dot{x}(t) = Ax(t) + w(t) \quad (16)$$

$$z(t) = Cx(t) + v(t) \quad (17)$$

where $x = [\theta \quad \dot{\theta} \quad \ddot{\theta} \quad \delta e \quad \dot{\delta e}]^T$, $z = [\theta_{meas} \quad \delta \xi_{high}]^T$, $w(t)$ and $v(t)$ are white Gaussian noises, zero means, whose covariances are respectively Q and R such as:

$$Q = \begin{bmatrix} \Gamma_2 Q_2 \Gamma_2^T & 0 \\ 0 & Q_1 \end{bmatrix} \quad R = \begin{bmatrix} R_2 & 0 \\ 0 & R_1 \end{bmatrix}$$

Matrices A and C are defined by:

$$A = \left[\begin{array}{c|c|c} A_2 & & 0_{(3n \times 2m)} \\ \hline 0_{(2m \times 2n)} & B_1 & A_1 \end{array} \right]$$

$$C = \left[\begin{array}{c|c} C_2 & 0_{(n \times 2m)} \\ \hline 0_{(1 \times 3n)} & C_1 \end{array} \right]$$

C. Incorporation of Time Delayed Measurements in a Discrete-time Kalman Filter

Let's consider that the linear discrete system derived from the continuous system (16–17) is observed by non-delayed measurements where both process and measurements are influenced by additive Gaussian noises:

$$x_{k+1} = \mathbf{A}x_k + w_k \quad (18)$$

$$z_k = \mathbf{C}_k x_k + v_k \quad (19)$$

where the noise v_k and w_k are independent (respective constant variances R and Q). \mathbf{A} is assumed to be invariant.

The optimal state estimator minimizing the variance of the estimation error will then be a Kalman filter:

$$\hat{x}_k^- = \mathbf{A}\hat{x}_{k-1}^+ \quad (20)$$

$$P_k^- = \mathbf{A}P_{k-1}^+ \mathbf{A}^T + Q \quad (21)$$

$$K_k = P_k^- \mathbf{C}_k^T [\mathbf{C}_k P_k^- \mathbf{C}_k^T + R]^{-1} \quad (22)$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k [z_k - \mathbf{C}_k \hat{x}_k^-] \quad (23)$$

$$P_k^+ = [I - K_k \mathbf{C}_k] P_k^- \quad (24)$$

Compared to other kinds of sensors used in robotics, vision devices have the disadvantage of a long processing time. It leads not only to delayed measurements, but also potentially to low update rates. In our case the visual data measurement

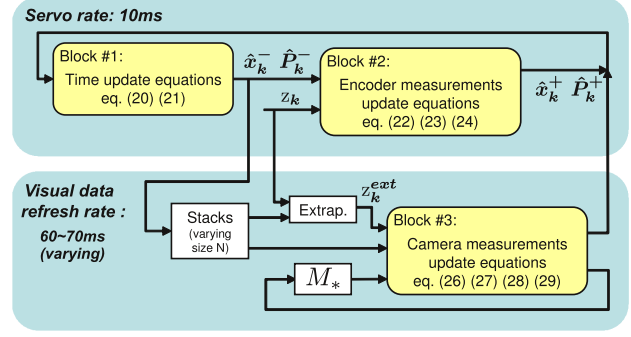


Fig. 3. Modified Kalman filter

$\delta \xi_{high}$ is delayed of about one sample and is approximately updated every 60 ~ 70ms –sometimes 100ms!–, depending on the charge of the supervisor computer which runs a non-real-time OS. On the other hand the estimation of δe is expected to be done at the servo rate, i.e. every 10 ms.

To overcome these two issues we propose to push the method exposed in [14] a little further. We consider a two-time-scale Kalman filter composed of 3 blocks (see Fig. 3):

- Block #1: the time update equations are executed at the servo rate and estimate the state variables as fast as needed to perform a stable and quality control.
- Block #2: the measurement update equations corresponding to the optical encoder measurement are also executed at the servo rate and correct the state estimation regarding the measured articular position.
- Block #3: the measurement update equations corresponding to the camera measurement are executed at the visual data refresh rate and refine the state estimation regarding the measured image features displacement.

An important point to be noted is that the time update equations, which are intrinsically stable, never stop predicting the state vector. As a result this method is robust against visual troubles such as partial occlusions or failure of the camera.

In order to avoid phase-lag between the real state and the estimated state, it is necessary to take into account the delay due to the processing time into the Kalman filter. One can use a delay compensator that extrapolates the measured output to the present time using past and present estimates. In [12] the delay is assumed constant which is quite limiting. Indeed, when the visual data processing application runs on a non-real-time OS, the delay can vary a lot and it is hardly possible to predict if previous measurements are fused in this delay period. As a consequence we have adapted the above-cited method to obtain a more flexible delay compensator.

Let's consider that the output corresponding to the camera feedback in the system (18–19) is delayed by an indeterminate number of samples N . The new output equation is:

$$z_k^* = C_s^* x_s + v_k^* \quad (25)$$

where the variance of z_k^* is R^* and $s = k - N$. This delayed measurement cannot be fused using the regular Kalman filter equations but requires some modifications in the structure of the filter. Indeed if the measurement z_k^* is delayed by N samples and fused at time k , the data update should

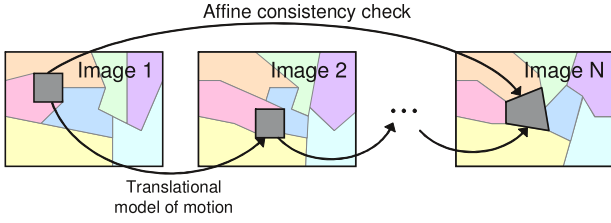


Fig. 4. Tracking method

reflect the fact that the N data updates from time s to k , and therefore the state and covariance estimates, have all been affected by the delay. A new optimal Kalman gain is computed so that the variance of the estimation error is as low as possible. The measurement update equations of this modified discrete-time Kalman filter can then be written:

$$z_k^{ext} = z_k^* + \mathbf{C}_k^* \hat{x}_k - \mathbf{C}_s^* \hat{x}_s \quad (26)$$

$$\mathbf{K}_k = \mathbf{M}_* \mathbf{P}_s \mathbf{C}_s^{*T} [\mathbf{C}_s^* \mathbf{P}_s \mathbf{C}_s^{*T} + \mathbf{R}^*]^{-1} \quad (27)$$

$$\hat{x}_k^+ = x_k^- + \mathbf{K}_k [z_k^{ext} - \mathbf{C}_k \hat{x}_k^-] \quad (28)$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{C}_s^* \mathbf{P}_s \mathbf{M}_*^T \quad (29)$$

where:

$$\mathbf{M}_* = \begin{cases} \mathbf{I} & \text{if } N = 0 \\ \prod_{i=0}^{N-1} (\mathbf{I} - \mathbf{K}'_{k-i} \mathbf{C}_{k-i}) \mathbf{A} & \text{if } N > 0 \end{cases} \quad (30)$$

The prime on \mathbf{K}' means that these Kalman gain matrices have been calculated using a covariance matrix updated at time s with the covariance of the delayed measurement.

This method guarantees all time-optimality and limits the computational burden, contrarily to other approaches [15]. From a practical point of view it is to be noted that the measurement equations of blocks 2 and 3 can be merged and executed together when the visual data is updated.

D. Tracking of features from the unknown environment

One main contribution of this paper consist in implementing the Lucas-Kanade-Tomasi (KLT) feature tracking algorithm to evaluate the speed of the environment in the camera basis, and thus to deduce the speed of the camera in the environment basis which is definitely static.

We assume that no a-priori knowledge on the environment is available as well as no markers have been placed on it.

As the manipulator moves, the tip camera moves and the patterns of image intensities change in a complex way. These changes can be described as image motion:

$$I(x, y, t + dt) = I(x - \zeta, y - \eta) + \mathbf{N}(x, y) \quad (31)$$

In other words an image taken at time $t + dt$ can be obtained by moving every point in the previous image, taken at time t , by a suitable amount. \mathbf{N} represents the visual noise. The amount of motion $\delta X(\zeta, \eta)$ is called the displacement of the point $X = (x, y)$. The image motion is better represented by an affine motion field $\delta X = \mathbf{D}X + d$ where \mathbf{D} is a deformation matrix and d is the translation of the window's centre. Given two images, tracking means determining the parameters that appear in the deformation matrix \mathbf{D} and the

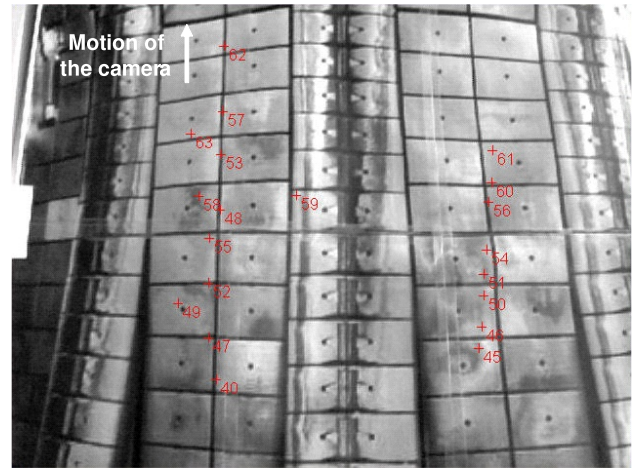


Fig. 5. Example of tracked features in an unknown and "untrimmed" environment (a tokamak vessel wall)

displacement vector d .

Practically visual noise prevent from tracking single pixels, thus we consider fixed-size feature windows. The quality of the estimate depends on the size of the window, the "texturedness" of the image within it, and the amount of camera motion between frames.

The particularity of the KLT algorithm [16] lies in the fact it is designed to select features that are more than traditional "interest" measures, that are often based on a preconceived and arbitrary idea of what a good feature is and consequently do not guarantee to be the most reliable for the tracking algorithm. From the KLT point of view the right features cannot be defined independently of the tracking method and are exactly those that make the tracker work best. As a result, the selection criterion is optimal by construction. This idea is the real strength of the KLT and makes it extremely robust.

To keep track of the selected features through the sequence a pure translation model of motion is used. At the same time the tracker performs an affine consistency check. In other words it defines a measure of dissimilarity that quantifies the change of appearance of a feature between the first and the current image. If the window has changed too much, it is discarded (see Fig. 4).

When features are lost our implementation of the algorithm replaces the lost features by finding new features in the new image and consequently keeps a constant pool of features. To ensure that the new detected features do not correspond to already detected features we build a mask image containing the current pool of tracked features and we filter out those of the new features that match it. To avoid optical distortion, features in the image borders are also junked. At last to secure a representative estimation of the environment displacement (exposed in section II-E) a minimum distance between the extracted features is set.

As shown on Fig. 5 the spatial distribution of the tracked features tends to be relatively regular because of the above mentioned minimum distance between them. Moreover one can notice that the selected features are concentrated in the centre of the image.

E. Robust estimation of the features displacement

As stated in section II-A, the quality of the vibration reconstruction is based on the accuracy of the environment displacement measurement. Because of the features extraction noise, outliers can corrupt the state observer as well as the J_{image} estimator. To minimize the influence of these outliers we employ robust statistics, which makes it possible to recover the structure that best fits the majority of the data while identifying and rejecting deviating substructures.

M-estimators can be considered as a more general form of Maximum Likelihood Estimators (MLE) because they permit the use of different minimization functions not necessarily corresponding to normally distributed data. This class of estimators can be written:

$$\hat{T}_n = \underset{T_n}{\operatorname{argmin}} \left[\sum_{i=1}^n \lambda(x_i, T_n) \right] \quad (32)$$

where λ is an influence function (Huber's, Cauchy's...).

To obtain a proper estimation of the environment displacement in the image we chose to implement Tukey's influence function:

$$\lambda(u) = \begin{cases} \frac{1}{6}[1 - (1 - u^2)^3] & \text{if } |u| \leq 1 \\ \frac{1}{6} & \text{if } |u| > 1 \end{cases} \quad (33)$$

where $u = \frac{x - T_n}{c \times S_n}$. S_n represents the Median Absolute Deviation (MAD) estimator and c is a potentiometer that adjusts the asymptotic efficiency of the obtained M-estimator.

This influence function completely rejects outliers by giving them a zero weight. This is of interest in our application to prevent detected outliers from having any effect on $\delta\xi_{high}$.

F. Real-time Jacobian Matrix Estimator

The vibration estimator built in section II-A assumes that the velocity of the tip camera can be related to the velocity of the image features through the so-called Image Jacobian. This matrix, which is used to linearly describe the differential relation between the motion of the image features to the camera motion, was first introduced by [17], who referred to it as the feature sensitivity matrix. It is also referred to as the interaction matrix [18] and the B matrix [19].

Determining this matrix analytically is not simple. One has to take into account the intrinsic parameters of the camera (focal distance, image centre coordinates, aspect ratio, distortion coefficients) as well as the depth estimation (translations and rotations between the camera and the features).

To deal with changing or unknown environments, some on-line Image Jacobian matrix estimator have been proposed by [20] and [21], in which the Jacobian is estimated recursively by just observing the process without any a priori model or introducing any extra "calibration" movements.

The estimation of the Jacobian matrix using the Broyden method [22] is an underdetermined problem and a family of solutions can be chosen as Broyden updating formulas. Among this infinite number of solutions, [20] proposes a Jacobian estimator that can be formulated as:

$$\hat{J}_t - \hat{J}_{t-dt} = \frac{(\Delta\xi_{low,t} - \hat{J}_{t-dt}\Delta\theta_t) \Delta\theta_t^T W_t}{\rho + \Delta\theta_t^T W_t \Delta\theta_t} \quad (34)$$

where $W(t)$ and $\rho(0 \leq \rho \leq 1)$ respectively denote a full rank weighting matrix and a forgetting factor. Its goal is not to estimate the true parameters of J_{image} but to provide an estimation \hat{J}_{image} that satisfies at any time the relation:

$$\dot{\xi}_{low} = \hat{J}_{image} J_e \dot{\theta} \quad (35)$$

Therefore the estimated parameters do not necessarily converge to the true physical values.

Nevertheless the estimation algorithm would display greater stability if it considered data over a period of time instead of just the previous iteration. To that purpose the so called "population-based" method has been introduced, for calibrating a linear model based on several previous iterates. This can be accomplished easily using a recursive least squares (RLS) algorithm with exponential data weighting that minimizes a cost function based on the change in the affine model. It is done by adopting the covariance matrix $P(t - dt)$ as the weighting matrix $W(t)$:

$$\hat{J}_t = \hat{J}_{t-dt} + \frac{(\Delta\xi_{low,t} - \hat{J}_{t-dt}\Delta\theta_t) \Delta\theta_t^T P_{t-dt}}{\rho + \Delta\theta_t^T P_{t-dt} \Delta\theta_t} \quad (36)$$

Here $P(t)$ denotes a covariance matrix:

$$P_t = \frac{1}{\rho} \left(P_{t-dt} - \frac{P_{t-dt} \Delta\theta_t \Delta\theta_t^T P_{t-dt}}{\rho + \Delta\theta_t^T P_{t-dt} \Delta\theta_t} \right) \quad (37)$$

As previously the behaviour of this method depends on the forgetting parameter ρ , which can be tuned from 0 to 1, and ponders previous movements. ρ settles a compromise between information provided by old data from previous movements and new data, possibly corrupted by noise. This kind of RLS algorithm has a memory approximately equal to $\frac{1}{1-\rho}$. This estimator is valid only when J_{image} is time-invariant or when the system moves slowly. But tuning forgetting factor ρ , we can apply this method when the system velocity is small.

In [23] a novel approach for uncalibrated visual servoing introduced a RLS algorithm with a modified stabilizing term, in eq.(37) only, which linearly depends on P_{t-dt} .

Curiously no paper has ever proposed to adopt an adaptive forgetting factor in both eqs. (36) and (37). As a consequence we suggest considering a memory term $\rho(\dot{\theta})$ which self-adjusts depending on the range of the camera rigid motion.

- If the camera does not move ($J_\theta \dot{\theta} \simeq 0$), ρ is put to 1. The new information is averaged with all past data, the system is hardly insensitive and very stable.
- If the camera moves fast ($J_\theta \dot{\theta} \gg 0$), ρ is put close to 0. The system becomes only sensitive to observed data.
- Between these two extreme cases, ρ linearly varies between 1 and 0. The old data are weighed less and less, and the estimator tracks the time-varying Jacobian.

This adaptive RLS algorithm displays great stability and improves sensibility to sudden movements of the camera compared to classic fixed-parameters Broyden-based methods. This method performs a robust estimation of the Image Jacobian, with a quite low sensitivity to noise and a relatively high repeatability.

III. CONTROLLER DESIGN

The control problem is to determine the torque vector $\delta\tau$ such that the joint rotation vector θ tracks the desired trajectory θ_d with a good accuracy while δe converges to zero as fast as possible. Thus we can regard each joint as a single-input multiple-output (SIMO) system. A common approach in the control of such systems involves the design of linear quadratic regulators (LQR). To that purpose we use the linear state-space representation of the entire flexible arm, including rigid and flexible modes. From eq. (5) we deduce:

$$\dot{X}(t) = \underbrace{\begin{bmatrix} 0 & I \\ -M^{-1}K & 0 \end{bmatrix}}_{A_c} X(t) + \underbrace{\begin{bmatrix} 0 \\ -M^{-1}U \end{bmatrix}}_{B_c} \delta\tau(t) \quad (38)$$

$$Y_c(t) = \underbrace{\begin{bmatrix} I & 0 & 0 & 0 \end{bmatrix}}_{C_c} X(t) \quad (39)$$

with $X = [\theta \ \delta e \ \dot{\theta} \ \dot{\delta e}]^T$. The LQR technique involves choosing a control law $\delta\tau = L_1 Y_c - L_2 X$ which stabilizes the controlled output vector Y_c to its desired values $Y_d = \theta_d$ while minimizing the quadratic cost function:

$$J = \int_0^\infty (X(t)^T Q X(t) + \delta\tau(t)^T R \delta\tau(t)) dt \quad (40)$$

where the penalty matrices Q and R are both symmetric positive semidefinite matrices. The optimization horizon N for the summation is the stopping time step for the cost function which is set most of the time to approach infinity.

Due to the coupled dynamics of the arm, the control of the low-dynamic rigid motion subsystem will be made much easier if the fast-dynamic elastic vibration subsystem is controlled as fast as possible. As a consequence the component of Q related to δe (and $\dot{\delta e}$) must be chosen large in comparison with those related to θ (and $\dot{\theta}$).

Finding L_1 and L_2 that minimise J involves solving the following Riccati equation [24]:

$$P A_c + A_c^T P - P B_c R^{-1} B_c^T P + Q = 0 \quad (41)$$

where P is the $2(n+m) \times 2(n+m)$ matrix to be determined.

At last the feedforward and feedback matrices L_1 and L_2 can be computed following the relations:

$$L_2 = R^{-1} B_c^T P \quad (42)$$

$$L_1 = (C_c (A_c - B_c L_2)^{-1} B_c)^{-1} \quad (43)$$

IV. EXPERIMENTAL RESULTS

Before implementing our vibration suppression scheme on the real AIA, a validation campaign has been carried out on the experimental mock-up shown in Fig.6 [25]. It consists of:

- an actuated joint (capacity $\simeq 1000$ N.m) driven by a motor through an Harmonic Drive based speed reducer
- a 3m-long circular beam with a calibrated tip mass
- a 5000 cpr optical encoder to measure the joint position
- a laser tracker Leica LTD800 to measure the tip position (accuracy = 5.10^{-5} m, frequency $\simeq 500$ Hz)

The controller runs on the real-time OS VxWorks at a sampling time of 10 ms. The overall vision-based application

is based on the ViSP software [26] and runs at around 10–12Hz. The joint friction and gravity torques applied on the beam have been compensated considering measurements from a rigid bar of the same weight. The KLT algorithm has been implemented using the OpenCV implementation which is computationally the most efficient at the present time. The tracker contains a pool of 20 features and it considers 10×10 windows for each. The minimum distance between features has been set to 30 pixels and the quality factor to 0.01.

Displaying an example of the obtained J_{image} wouldn't make much sense as it does not necessarily converge towards values having a physical meaning (see section II-F). Instead it is quite eloquent to rebuild the low-filtered camera velocity $\tilde{\xi}_{low}$ from the angular velocity and the estimated J_{image} , and to compare it to the measured camera velocity. Figure 7 validates the benefit of our method. The best results are obtained for $min(\rho) = 10^{-3}$ and $\dot{\theta}_{limit}$, which characterizes the self-adjustment of ρ , is set to $10^{-3} rad/s$. The error between the rebuilt and measured velocity is around 14.1%, while it is up to 46.5% when ρ is fixed to 0.4.

The robustness of the environment displacement estimation is greatly improved by the use of the Tukey M-estimator. For example the displacement estimation of a static scene disturbed by fleeting partial occlusions yields temporal standard deviation σ in the order of 0.012979. In comparison the same estimation made by a classic mean and a trimmed mean respectively yields $\sigma = 1.619542$ and $\sigma = 0.017504$.

Then we can compare the behaviour of the link with and without vibration suppression scheme. As illustrated by Fig.8 the response of the controlled system is very sensitive to the choice of the penalty matrix Q . This point is also outlined in the short video accompanying this paper. In the first overall view, which could correspond to the green curve of Fig. 8, priority has been given to the position control. The desired joint position is reached quite fast but a slight overshoot occurs on δe . In the following close-up, priority has been given to the vibration compensation. As it can also be noticed on the red curve of Fig.8, the joint is a bit slower to reach its desired position but the vibration is much better damped.



Fig. 6. Flexible mock-up at CEA LIST site in Fontenay-aux-Roses

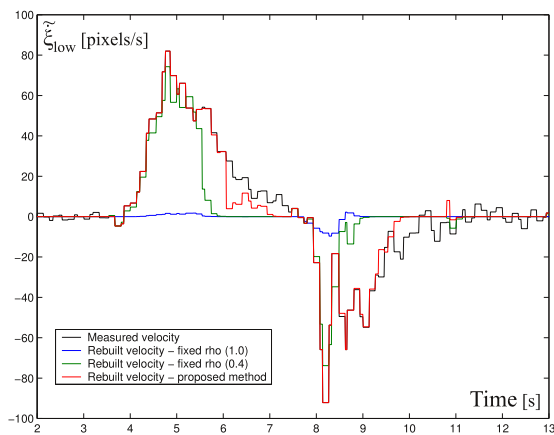


Fig. 7. Camera velocity rebuilt from different Image Jacobian estimators

V. CONCLUSION AND FUTURE WORKS

In this paper a robust vibration control has been presented for a flexible inspection arm moving in an unknown environment. Its particularity is to entail a vibration estimator that reconstructs the vibrations from visual data without any a priori knowledge of the surroundings. To that purpose a robust tracker based on the KLT algorithm feeds a two-time-scale Kalman filter in which a part of the measurements are delayed. In other applications this modified Kalman filter could be extended to multi-sensors systems with different delays and refresh rates. Moreover the issue of Image Jacobian on-line estimation has been assessed and an adaptive method is proposed to ensure both stability and sensibility of the estimation whatever the camera velocity may be.

The whole control scheme is validated on a single-joint mock-up until the availability of the AIA arm makes possible to implement it on the real 8-DOF system. As each joint of the complete arm enables an elevation and a rotation, the control of its end-effector can be considered as a collocated problem. The estimated vibration will thus be projected on an orthogonal basis and suppressed using only the motion of the fifth module. Of course this strategy cannot prevent middle parts of the structure from oscillating. Therefore we rely upon internal damping to stabilize the whole structure as long as the diagnostic tool plugged at the tip is not vibrating.

In future works we will try to extend these results to the observation of complex and not necessarily normal planes.

REFERENCES

- [1] L. Gargiulo, P. Bayetti, V. Bruno, and et al., "Operation of an ITER relevant inspection robot on Tore Supra tokamak," *Fus. Eng. Des.*, vol. 84, no. 2-6, pp. 220 – 223, 2009.
- [2] D. Keller, P. Bayetti, J. Bonnemason, and et al., "Real time command control architecture for an ITER relevant inspection robot in operation on Tore Supra," *Fus. Eng. Des.*, vol. 84, pp. 1015 – 1019, 2009.
- [3] C. Neri, L. Bartolini, A. Coletti, and et al., "The in-vessel 3D inspection system for ITER," in *Proc. of SOFE*, 2007, pp. 1-4.
- [4] C. Hernandez, "Development of a Laser Ablation System Kit (LASK) for Tokamak in vessel tritium and dust inventory control," *Fus. Eng. Des.*, vol. 84, no. 2-6, pp. 939 – 942, 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.

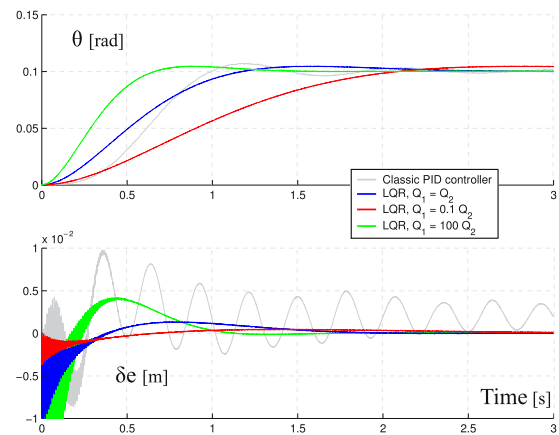


Fig. 8. Angular step response for different adjustments of the controller

- [7] 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.
- [8] P. C. Tang, H. C. Wang, and S. S. Lu, "A vision-based position control system for a one-link flexible arm," in *Proc. of the IEEE International Workshop on Intelligent Motion Control*, vol. 2, 1990, pp. 523-528.
- [9] K. Hashimoto, T. Kimoto, T. Ebine, and et al., "Manipulator control with image-based visual servo," in *Proc. of ICRA*, 1991, pp. 2267-2271.
- [10] L. Bascetta and P. Rocco, "Two-time scale visual servoing of eye-in-hand flexible manipulators," *IEEE Trans Robot*, vol. 22, no. 4, 2006.
- [11] A. Ladikos, S. Benhimane, M. Appel, and N. Navab, "Model-free markerless tracking for remote support in unknown environments," in *Proc. of VISAPP*, 2008.
- [12] 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.
- [13] P. R. Belanger, "Estimation of angular velocity and acceleration from shaft encoder measurements," in *Proc. of ICRA*, 1992, pp. 585-592.
- [14] T. D. Larsen, N. A. Andersen, O. Ravn, and N. K. Poulsen, "Incorporation of time delayed measurements in a discrete-time kalman filter," in *Proc. of CDC*, vol. 4, 1998, pp. 3972-3977.
- [15] H. L. Alexander, "State estimation for distributed systems with sensing delay," in *SPIE*, vol. 1470, 1991.
- [16] J. Shi and C. Tomasi, "Good features to track," in *Proc. of CVPR*, 1994.
- [17] L. Weiss, A. Sanderson, and C. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE J Robot Autom*, vol. RA-3, no. 1, pp. 404-417, 1987.
- [18] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans Robot Autom*, vol. 8, no. 3, pp. 313-326, 1992.
- [19] N. Papanikolopoulos, P. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision," *IEEE Trans Robot Autom*, vol. 9, pp. 14-35, 1993.
- [20] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true jacobian," in *Proc. of IROS*, 1994, pp. 186-193.
- [21] M. Jagersand, O. Fuentes, and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," in *Proc. of ICRA*, 1997, pp. 2874-2880.
- [22] C. G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Mathematics of Computation*, vol. 19, pp. 577-593, 1965.
- [23] K. Miura, K. Hashimoto, J. Gangloff, and M. de Mathelin, "Visual servoing without jacobian using modified simplex optimization," in *Proc. of ICRA*, 2005, pp. 3504-3509.
- [24] A. J. Laub, "A schur method for solving algebraic riccati equations," in *roc. of IEEE CDC*, 1978, pp. 60-65.
- [25] 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.
- [26] 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.