



HAL
open science

Dynamic system stochastic identification mixed with video processing: Validation on a real case

Goeller Adrien, Jean-Luc Dion, Ronan Le Breton, Thierry Soriano, Bernard Roux

► **To cite this version:**

Goeller Adrien, Jean-Luc Dion, Ronan Le Breton, Thierry Soriano, Bernard Roux. Dynamic system stochastic identification mixed with video processing: Validation on a real case. 17th International Conference on Research and Education in Mechatronics (REM), 2016 11th France-Japan & 9th Europe-Asia Congress on, Jun 2016, COMPIEGNE, France. pp.146-151, 10.1109/MECATRONICS.2016.7547131 . hal-01589631

HAL Id: hal-01589631

<https://hal.science/hal-01589631>

Submitted on 24 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Dynamic system stochastic identification mixed with video processing: validation on a real case

Adrien Goeller – Jean Luc Dion – Ronan Le Breton –
Thierry Soriano
SUPMECA QUARTZ Laboratoire – EA 7393
Saint-Ouen, France
adrien.goeller@supmeca.fr

Bernard Roux
Vannier-Kinoptik
Antony, France

Abstract— Nowadays, visual information is everyday more present, cameras are more and more fast, small and accurate. Improvements in computer vision enable to go further: to consider cameras as smart sensors.

In this framework, this article propose to mix complex mechanical models with video processing in order to estimate invisible parameter and to enable the camera to forecast in the video, with a dynamic meaning. The concept is developed on particular cases of objects moving in a scene where the scenario is known. This application is a proof of concept on a case study of a 2D scenario of a ruler sliding on a table. The aim is to prove the feasibility on a real case with real time architecture.

Dynamic systems, stochastic identification, computer vision, Kalman filters

I. INTRODUCTION

In several parts in the world, manufacturers are expected to make a revolution. With the explosion of the Internet of Things on the consumer market, and the improvement on domains such as augmented reality, virtual reality, or robotics, manufacturing is being redesigned in several development smart industry programs. Leaders companies are already taking the leap and investing in new methods of manufacturing. Improvements of costs, better quality monitoring, new diagnostic tools, data management, the field for innovation in those industries is very large. Cameras are particularly one of the smart sensor of those manufacture method by improving quality, traceability or manufacturing time. In this way, the development of a system able to mix complex mechanical model directly with frame from camera could open the path for the machine to understand the filmed scene and process augmented variables as prediction, interpolation, or estimation of hidden variables.

Mixing data with models is not a new trend. Marchand & al. [1] use UKF to do identification and model calibration, Naets and al. [2] use accelerometers to display an augmented reality measurement on a vibrating beam, Simon [3] applied it on complex model. Kalman filters have been used in image tracking [4] but with linear models, but this is Particle filter which is most used in video processing to calibrate models (see [5]).

Following the development initiated in a previous publication [6], this article carry on the promising concept to enable vision machine dealing with all the points mentioned before by improving the comprehension of the object by the computer. Based on a physical model of the object seen, this algorithm enables the machine to understand the behavior of the scene and to extract some hidden physical parameters directly from observation. This article introduce those concept are applied on a case study of a ruler sliding on a table.

After a general overview of the algorithm, the model of the case study is described. Kalman filters and extraction process are developed after. Then the experimental setup is explained before the results.

II. CASE STUDY

A. Algorithm overview

The algorithm introduced here is composed of 3 main parts described on Fig. 1. The important part is that every block is designed to be real time with architecture frame to frame. In spite of the current results which are not yet real time, the aim is to switch to real time during a future code optimizing phase.

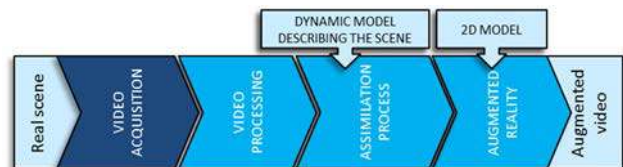


Fig. 1: Macro architecture of the algorithm

The video is respecting a classical frame of object detection. We took a static background hypothesis. It allows using background subtraction technic [7]. Here the detector used is a GMM (Gaussian Mixed Mixture) [8] and gives a binary mask of the ruler. Extracted information is the center position of the object and the main orientation angle α using the statistical moments [9]. From those position are calculated the object center speed and angular speed. Length and width are measured directly on the image and correspond to the variables. The state vector used later in the assimilation part is created from those extracted values.

The assimilation process is composed by two parts. First, a decision process is designed to initialize and manage the second part. This first part is based on the number on consecutive frames where the object is detected. The second part is an Unscented Kalman Filter (UKF) [10]. This architecture of Kalman filter management allows to run the Kalman filter without observation and to be more robust.

Finally, the video output is augmented with the data from the dynamic model using a 2D model of the ruler. Here, the 2D model chose here is a simple rectangle. In order to separate the detection errors and the filter errors, two rectangles are displayed, one for the detected object, one for the assessed object ruled by the filter result (see Fig. 2)

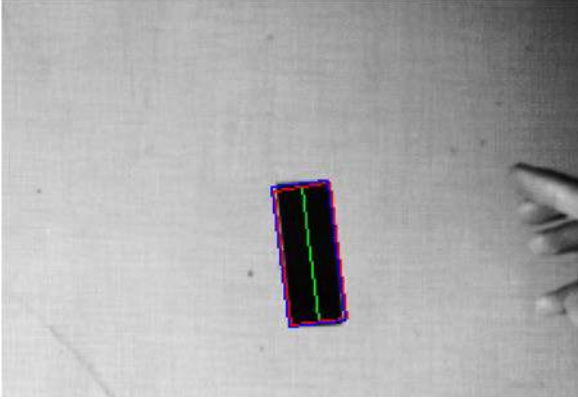


Fig. 2: Augmented video with real ruler in black, extracted ruler in blue, assessed ruler in red and length assessment in green.

B. Model description

The model chosen to simulate the behavior of the ruler is a double sliding contact point, A and B in the Fig. 3: ruler consideration for study, which models the ruler resistance by solid friction. T_A and T_B are the friction forces applied on A and B.

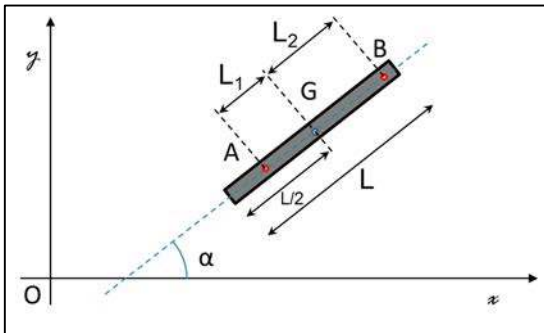


Fig. 3: Ruler parametrization for the study

The position of A and B could be variable in order to model some rough variation on a real table. The position of the two sliding contacts enables to model a ruler not perfectly flat.

$$\begin{cases} m\ddot{x} = T_{Ax} + T_{Bx} \\ m\ddot{y} = T_{Ay} + T_{By} \\ \bar{I}\ddot{\alpha} = L_1(T_{Ax}\sin(\alpha) - T_{Ay}\cos(\alpha)) \\ \quad - L_2(T_{Bx}\sin(\alpha) - T_{By}\cos(\alpha)) \end{cases}$$

Fig. 4: Equation system of the model

Two main hypothesis have been taken here in the model:

- The table surface is considered to be perfect which imply that the friction coefficient is considered constant.
- The 2 contact points are symmetric regarding the ruler gravity center.

III. EXTRACTION PROCESS

A. Gaussian Mixture Models (GMM)

The GMM has been first proposed by Stauffer and Grimson [8]. It is first a method to model an unknown distribution with a sum of weighted Gaussian.

Let's explain with an application related to object extraction. We define for a particular pixel (u_0, v_0) an history of intensity grayscale value: $I = \{I_1, \dots, I_t\}$.

The aim here is to assess which intensities are most probably in the background. Supposing we want to model a group of pixel by K Gaussian, let's define a canonical basis of dimension K based on those Gaussian: $\{e_1, \dots, e_K\}$

We define a hidden random variable $Z \in \{e_1, \dots, e_K\}$ and the probability $P(Z = e_k) = \beta_k$.

So when $Z = e_k$, X is projected onto the k^{th} Gaussian: $I \sim N(m_k, \Sigma_k)$ with m_k the mean and Σ_k the covariance of the Gaussian.

Which gave for $p(I) = \sum_Z p(I|Z) \cdot p(Z)$

$$\begin{aligned} &= \sum_{k=1}^K p(Z = e_k) \cdot p(I|Z = e_k) \\ &= \sum_{k=1}^K \beta_k \cdot N(I|m_k, \Sigma_k) \end{aligned}$$

This means that the probability of X can be formulated as a sum of K weighed Gaussian. The problem consists on optimizing the values $(\beta_k, \mu_k, \Sigma_k)$ for each $k \in \{1, \dots, K\}$

The procedure of initialization is the following:

1. Define the N first frame without the object to find a set of the values $(\beta_k, \mu_k, \Sigma_k)$, $k \in \{1, \dots, K\}$. In our case $N = 5$.
2. After those N first frame, for each frame, classify each pixel into the category: {Foreground, Background}.

3. Use the connected component in order to consolidate the detection.
4. Update the GMM with the “Background” labeled pixels.

The output of the GMM is a labelled frame, which means a frame which groups the different binary masks of the objects detected.

The GMM detection is very efficient for detecting moving object which appears to be as transitory signal regarding the background model. There is a limit when the object becomes static: the object is progressively integrated in the background. To become more robust, the solution is to provide more frame in the initialization phase.

Some authors add a tracking step [11] or a shadow detection step [12] to consolidate the GMM detector.

B. Statistical moments method

Statistical moments are here used only to find basic mechanical parameters of each object: centroid, area, and orientation. Only few parts of the Hu moment are described here, regarding the needs we had.

Considering a 2D continuous function $f(x,y)$, the raw moment is defined as :

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy$$

Thus the discrete equivalent to apply in a numeric image is:

$$M_{pq} = \sum_u \sum_v u^p v^q I(u,v)$$

I is the 2 variables function describing the intensity of each gray-level pixel.

In this framework, some moments refers to interesting images properties for a given object in the image: M_{00} as the area, or the centroid $(\bar{x}, \bar{y}) = (\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}})$.

To obtain the orientation, we place in the object reference by defining the central moments M'_{pq} as for the discrete version:

$$M'_{pq} = \sum_u \sum_v (u - \bar{x})^p (v - \bar{y})^q I(u,v)$$

A covariance matrix of $I(u,v)$ can be construct from those moments. Defining:

$$\overline{M}_{pq} = \frac{M'_{pq}}{M'_{00}}$$

$$\text{The covariance matrix: } cov(I(u,v)) = \begin{bmatrix} \overline{M}_{20} & \overline{M}_{11} \\ \overline{M}_{11} & \overline{M}_{02} \end{bmatrix}$$

From this matrix, the major axe of the object is given by the eigenvector of the largest eigenvalue, which means for the object orientation:

$$\theta = \frac{1}{2} \arctan \left(\frac{2 \overline{M}_{11}}{\overline{M}_{20} - \overline{M}_{02}} \right)$$

IV. STOCHASTIC IDENTIFICATION

A. Kalman Filters

Kalman filtering refers to a family of algorithms that tracks the temporal evolution of a dynamic model based on noised measurements. This evolution is described here in the discrete time domain:

$$\begin{cases} X_{k+1} = f(X_k, w_k) \\ Z_{k+1} = h(X_{k+1}, w'_{k+1}) \end{cases}$$

The first equation of the system above is the propagation equation whereas the second is the observation equation. X_k is the state vector at the step k , $f(\cdot)$ is the model function, w_k the process noise assumed to be Gaussian $N(0,Q)$, and Z_k is the observation vector, $h(\cdot)$ the observation function and w'_k the observation noise assumed to be Gaussian $N(0,R)$.

The Kalman filter is an efficient algorithm able to predict the future state vector using its means and covariances. It is based on a model of the system described by the function $f(\cdot)$ compared to the observation Z_k with the observation function $h(\cdot)$. The Kalman filter is based on two major hypotheses: the model function and the observation function are linear, and the variables considered in the system are Gaussian. Due to the linear hypotheses, $f(\cdot)$ is the F matrix and $h(\cdot)$ the H matrix in the following.

An approach of Kalman algorithm is to separate it in two phases. The predict phase gives an a priori estimate of the state and covariance based on previous time step t_k :

$$\begin{aligned} \hat{X}_{k+1|k} &= F_{k|k} \hat{X}_{k|k} \\ \hat{P}_{k+1|k}^{XX} &= F_{k|k} \hat{P}_{k|k}^{XX} F_{k|k}^T + Q \end{aligned}$$

The update phase corrects the deviation of these estimations based on new observations at time stem t_{k+1} . It is composed by an innovation phase:

$$\begin{aligned} \hat{Y}_{k+1|k} &= Z_{k+1} - H \hat{X}_{k+1|k} \\ \hat{P}_{k+1|k}^{YY} &= H \hat{P}_{k|k}^{YY} H^T + R \end{aligned}$$

Then the calculus of the Kalman gain:

$$K_{k+1} = \hat{P}_{k+1|k}^{XX} H^T (\hat{P}_{k+1|k}^{YY})^{-1}$$

Finally the update of the state and covariance:

$$\begin{aligned} \hat{X}_{k+1|k+1} &= \hat{X}_{k+1|k} + K_{k+1} \hat{Y}_{k+1|k} \\ \hat{P}_{k+1|k+1}^{XX} &= (I - K_{k+1} H) \hat{P}_{k+1|k}^{XX} \end{aligned}$$

B. Unscented Kalman Filters

Kalman has been derived in a lot of different forms in order to pass through the problem of non-linear cases. Among those methods, the Unscented Kalman filter method (called UKF) has been chosen to deal with the two issues of our case: the

mechanical model is highly nonlinear, and it is described by unsolved differential equations.

Unscented Kalman filter has a totally different approach compared to others filters as Extended Kalman filters [2]. The Extended version proposed to linearize using a first order development of the nonlinear function. The Unscented version proposes to use the probability distribution to approximate the nonlinear function.

The unscented transform enables the propagation of the state vector through the model function $f(\cdot)$ or the observation function $h(\cdot)$. In this way, a deterministic sampling technique is used to catch the state probability distribution and the two first moments. The sampling set is composed of sigma points and those sigma points are propagated through the nonlinear function. The minimal number of points is $n+1$ with n the size of the state vector. In our case, it has been chosen $2n+1$ sigma points. After that, the inverse unscented transformation is used to obtain the state vector and the covariance from the sigma points. This method can be repeated for the innovation phase whether the observation function is nonlinear too.

After the unscented step, the standard Kalman filter can be applied. Another unscented step is added around the innovation phase whether the $h(\cdot)$ function is nonlinear.

V. EXPERIMENTAL SETUP

A. State system choice

The state equation transform is a crucial step in the model process. Regarding to the model equation and description (Fig. 3 and Fig. 4), we can separate the state vector in 3 different parts:

- The observed variable from the camera: object centroid (x_G, y_G) , object angle (α) , and ruler length (L) .
- The model is using the translation and rotation speed as intermediate variables. These speeds have been added into the state vector in order to have a smoother value: object center speed and angular speed $(V_{x_G}, V_{y_G}, V_\alpha)$.
- The equation parameters: the friction coefficient μ and the position of the virtual points L_1 and L_2 .

The motion equations are quickly complicating to a nonlinear form due to the intervention of sinus/cosinus for the intervention of α . Moreover the differential system is a coupled one, making an explicit solution impossible. All these points lead to associate the Unscented Kalman filter with a Runge-Kutta 4 solving method.

The adopted approach is to make an observation matrix simple and linear and transfer all the nonlinearities to the model function.

The state vector and the observation vector at discrete time k are:

$$X_k = \begin{pmatrix} x_G \\ y_G \\ L \\ \alpha \\ V_x \\ V_y \\ V_\alpha \\ L_1 \\ L_2 \\ \mu \end{pmatrix} \text{ and } Z_k = \begin{pmatrix} x_G \\ y_G \\ L \\ \alpha \end{pmatrix}$$

The propagation matrix is linear and has the form:

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

B. Experimental setup

The test bench used to film the ruler movement is described on Fig. 5 below.

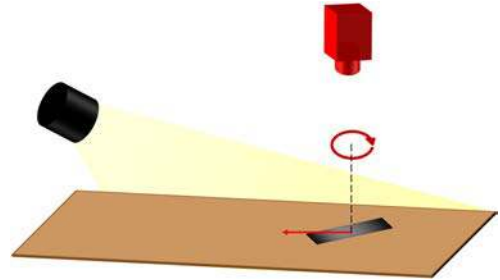


Fig. 5: Test bench

The setup of the experimental was defined in order to respect some hypotheses to simplify the video processing and to respect the framework used on the simulated ruler:

- A static background
- The real scene projected on the camera sensor can be approximated to a 2D scene.
- The object is considered to have the same aspect during the scene which means no reflection

The sensor plane has been set parallel to the table and the thickness of the ruler is about $\frac{1}{2}$ mm: this setup is avoiding shadows and perspective changes.

In order to avoid reflection, the light was set up with an angle of 30° in order to light up the table and keep the ruler as a dark object.

In order to solicit the whole model, 3 different scenarios have been chosen:

- Quasi exclusive translation movement
- Quasi exclusive rotation movement
- Mixed translation/rotation movement

VI. RESULTS

A. Extraction

The method used to quantify the results is to compare the area detected by the GMM detector and the real area measured on the frame.

The measure consists on extracting a first binary image from the detected polygon and a second binary image from the frame threshold with an adapted value to extract only the ruler. An “exclusive or” function allows to identify the pixels P which belongs to the not overlapped area by the two extracted masks (in white on Fig. 6). Then, the matching ratio is:

$$\text{Matching ratio} = \frac{\sum(P \in \text{Area}_{\text{not overlapped}})}{\sum(P \in \text{Area}_{\text{Real ruler}})}$$



Fig. 6: “Exclusive or” between the real ruler in orange and the detected ruler in blue

As we can see on Fig. 7 and Fig. 8 the extraction has good performance. There is one non-detection because of a parasite reflection, and the detection is not complete when the ruler appears in the frame and when it begins to disappear because of its static behavior.

Mean	Minimum	Maximum
89.0%	83.3%	96.2%

Fig. 7: Performance of detection during the movement without the non-detection

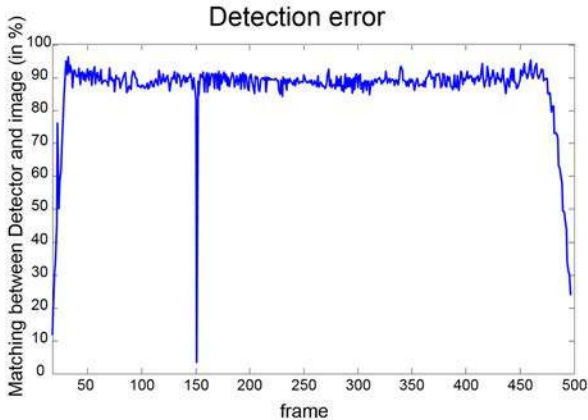


Fig. 8: Detection error

B. Assimilation

The assimilation part can be characterized in the restricted framework of our model of two kind of comparison.

1) Tracked variables

The variables observed. In our case it is the position, angle and ruler length. In the table below (Fig. 9) are summarizing the performances using the same method than the extraction (see Fig. 10).

Mean	Minimum	Maximum
68.5%	14.4%	100%

Fig. 9: Performance of assimilation compared to detection during the movement without the non-detection

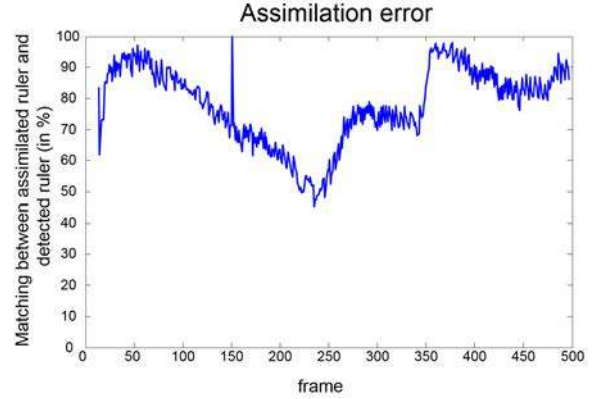


Fig. 10: Assimilation Error

Those results are far from satisfactory. But, it is interesting to compare it to the Fig. 11 of the different tracked variables. The results are more satisfying because the variables are tracked. The bad performances on the matching tests are due to a bad assessment of the velocities, which induce an error on the position. The correcting behavior of the Kalman filter can be seen from the frame 250 on Fig. 10.

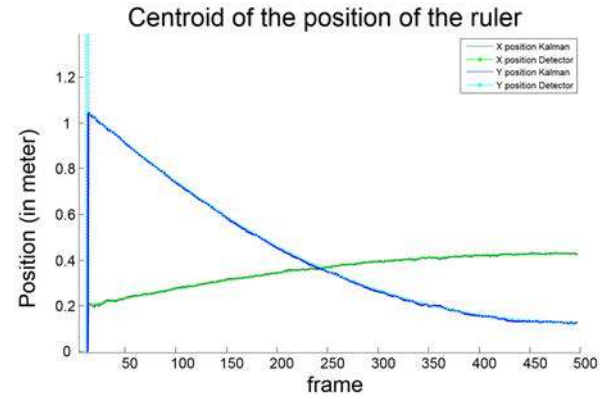


Fig. 11: Comparison between the assimilated position and the real position

2) Assessed variables

The variable linked to the observed variables by the model. It is the centroid velocity and the angular velocity. In order to do a comparison, the different velocities are also computed directly from the video processing data.

We can see on Fig. 12 that the assessment of the angular velocity is efficient and robust compare to the video processing computed angular velocity.

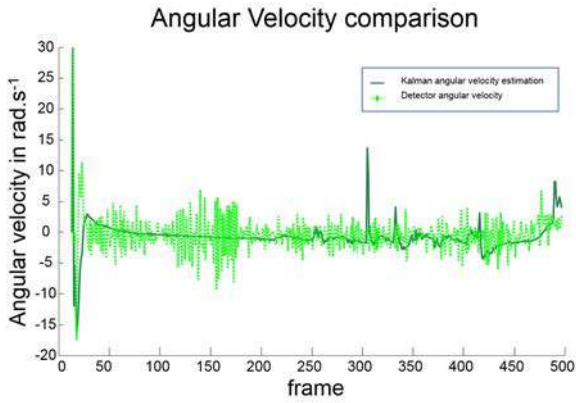


Fig. 12: Comparison between the assessed angular velocity and the angular velocity computed from detected angles

C. Estimation

The model has 2 parameters: the sliding friction coefficient and the 2 symmetric contact points. The sliding friction coefficient has been experimentally determined at about 0.88. We can see on the Fig. 13 that the value estimated is converging to the expected value which is the experimentally determined.

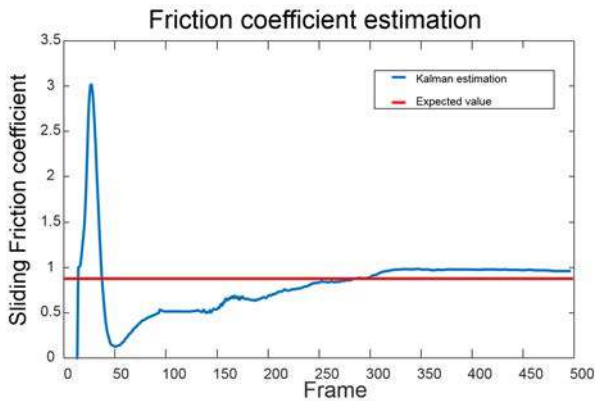


Fig. 13 : Comparison between the estimation value of the sliding friction coefficient and the expected value

VII. CONCLUSION

This article applies a concept of mixing dynamic models with video processing. The concept is developed here in 2D and for known objects with a known scenario in the filmed scene. The aim is to enable the camera to understand the mechanical behavior of the object in order to do prediction, interpolation, compression and augmented reality. The results developed here show the feasibility of the concept applied on a real case study of a ruler sliding on a table: variables tracking, variable assessment and parameter estimation.

VIII. REFERENCES

- [1] B. Marchand, L. Chamoin, and C. Rey, "Recalage en temps réel de modèle mécanique par couplage entre filtre de Kalman, erreur en relation de comportement modifiée et réduction de modèle," *S07 Mécanique Struct.*, 2015.
- [2] Naets, F. ; Dept. of Mech. Eng., KU Leuven, Leuven, Belgium ; Cosco, F. ; Desmet, W., "An extended Kalman filter approach for augmented strain/stress visualization in mechanical systems."
- [3] D. Simon and D. L. Simon, "Aircraft Turbofan Engine Health Estimation Using Constrained Kalman Filtering," *J. Eng. Gas Turbines Power*, vol. 127, no. 2, p. 323, 2005.
- [4] S. Xu and A. Chang, "Robust object tracking using Kalman filters with dynamic covariance," *Cornell Univ.*
- [5] C. Choi and H. I. Christensen, "3D textureless object detection and tracking: An edge-based approach," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 3877–3884.
- [6] A. Goeller, J.-L. Dion, T. Soriano, and B. Roux, "Real Time Dynamic System Stochastic Identification in Video Capture for Data Compression, Image Interpolation, Prediction, and Augmented Reality," presented at the 27th Conference on Mechanical Vibration and Noise, Boston, Massachusetts, USA, 2015, vol. 8.
- [7] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13–es, Dec. 2006.
- [8] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999, vol. 2.
- [9] M.-K. Hu, "Visual pattern recognition by moment invariants," *Inf. Theory IRE Trans. On*, vol. 8, no. 2, pp. 179–187, 1962.
- [10] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.
- [11] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *Pattern Anal. Mach. Intell. IEEE Trans. On*, vol. 22, no. 8, pp. 747–757, 2000.
- [12] P. KaewTraKulPong and R. Bowden, "An improved Background Mixture Model for Real-time Tracking with shadow Detection," presented at the European Workshop on Advanced Video Based Surveillance Systems, AVBS01, 2001.