MODELING, IDENTIFICATION AND CONTROL, 1991, VOL. 12, NO. 2, 69–80
doi:10.4173/mic.1991.2.2

Sensor integration using state estimators

J. G. BALCHEN†, F. DESSEN and G. SKOFTELAND

Keywords: *Sensor integration, Kalman filter, 3-D motion estimation, Robot vision.*

Means for including very different types of sensors using one single unit are described. Accumulated data are represented using an updatable dynamic model, a Kalman filter. The scheme handles common phenomena such as skewed sampling, finite resolution measurements and information delays. Included is an example where 3D motion information is collected by one or more vision sensors.

1. Introduction

Many robotic systems employ a multiplicity of sensory devices which extract information about the behaviour of the system and its environment. For the purpose of deriving specific information useful for control of the system, it is necessary to process the individual type of information in a systematic manner. This task is referred to as sensor integration (fusion). Sensors and sensor systems are established fields of research within robotics as well as within general control and dynamic estimation theory. Within the latter area, the concept of state and parameter estimators (observers), (Jazwinski, 1970) is well established and found useful at least in cases where the process can be considered continuous in the quantities to be estimated. Within robotics (including autonomous vehicles) contributions come from a large number of research communities resulting in a situation where many different, however no general, tools have been described for the fusion and interpretation of sensor data. This partly results since the number of different types of sensors is large and also because of the special research made on specific devices, such as cameras.

Often sensor integration is employed to resolve the problem of sensory redundancy which occurs when the system has more sensors than strictly necessary to determine a state variable of the system, say position. A frequently encountered example of this kind is when the position of an object is measured by a number of different sensors based on different physical principles and with different characteristics with respect to accuracy, noise, availability, etc. under varying operating conditions. If for instance one sensor fails, the other sensors can still be available to fulfill the task provided they have been properly integrated.

This contribution describes sensor systems for robotics in terms of state and parameter estimation schemes. Basic estimation schemes are outlined in § 2. A systematic means which allows the time of appearance of sensor data to be irregular or even stochastic is developed in § 3. The resulting scheme is intended for use in large sensory systems. A example is given in § 4, where the motion of an object is monitored by a number of cameras.

Received 20 November 1990.

† Norwegian Institute of Technology, Division of Engineering Cybernetics, N-7034 Trondheim, Norway.

Reprinted with the permission of Springer-Verlag, Heidelberg.

2. Methods for sensor integration

In Fig. 1, two alternatives are indicated for sensor integration. Part A suggests that the primary information from the sensors is handled directly by some algorithm producing the integrated information whereby proper account must be made for the individual characteristics of the sensors, their measuring uncertainties, etc. No general concept exists for this type of direct integration, but a large number of proposals have been made.

Part B of Fig. 1 is shows a block diagram of a system sensor integration using a recursive algorithm. This reflects a very general approach applicable to a large number of practical problems. It consists of two mathematical models and an updating algorithm. One of the models describes how the primary information is generated by the sensors in rclation to the behaviour of the system (the motion of robots, objects and so on). The other model describes the behaviour of the system (robot arm states, object states, etc.). The updating algorithm takes the residuals which is the difference between the primary information received from the sensors and the estimated sensor information received from the model of the sensors and corrects the behaviour of the system model in such a way to reduce the size of the residuals. The model of the system behaviour may assume levels of complexity. If the model describes the free movement in three dimensional space of a point (very small object), its total state vector may be composed of a position vector $\boldsymbol{p}$, a velocity vector $\boldsymbol{v}$ and an acceleration vector $\boldsymbol{a}$ which are all three-dimensional and related by a differential equation involving 9 variables,

$$\dot{\boldsymbol{p}}=\boldsymbol{v} \quad ; \quad \dot{\boldsymbol{v}}=\boldsymbol{a} \tag{1}$$

If the object has a finite size, its position and orientation needs to be described by 6 independent variables (6 dof). The same applies to the velocities and accelerations. A number of different coordinate systems may be chosen to describe the equations of rigid body movement yielding equivalent types of information.

One particular, theoretical basis for the realization of the recursive algorithm shown in Fig. 1 is the well known Kalman Filter (KF). The updating mechanism of the KF is a least squares, minimum variance algorithm. The KF appears in a continuous version applicable when the system model is in the form of differential equations and a discrete version applicable when the dynamic model consists of different equations. The elementary KF is based on linear models whereas the Extended Kalman Filter (EKF) allows for nonlinear models both of the system behaviour and of the sensors.

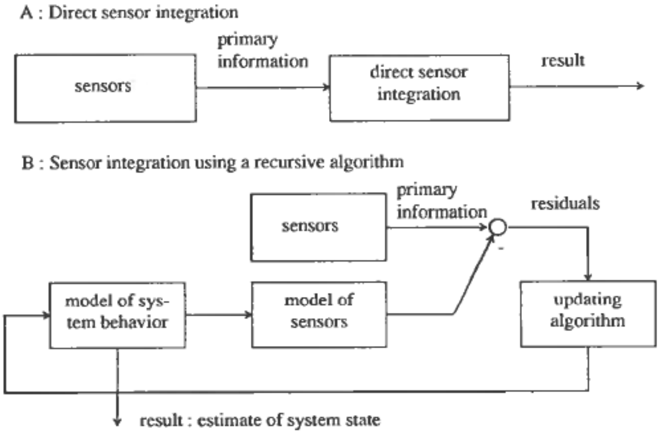
A : Direct sensor integration
sensors
primary information
direct sensor integration
result
B : Sensor integration using a recursive algorithm
sensors
primary information
residuals
-
model of system behavior
model of sensors
updating algorithm
result : estimate of system state

Figure 1. Two alternatives for sensor integration.

The EKF algorithm is reviewed as follows using standard state space notation:

$$\boldsymbol{x}(k+1)=f(\boldsymbol{x}(k))+\boldsymbol{n}(k) \tag{2}$$

with measurement model

$$\boldsymbol{y}(k)=h(\boldsymbol{x}(k))+\boldsymbol{w}(k) \tag{3}$$

Here

$\boldsymbol{x}(\cdot)$ state vector of positions, velocities and accelerations

k discrete time variable

$f(\cdot)$ vector of nonlinear functions in behaviour model

$\boldsymbol{n}(\cdot)$ vector of excitation forces, covariance matrix $\boldsymbol{N}$

$\boldsymbol{y}(\cdot)$ vector of primary sensor variables

$h(\cdot)$ vector of nonlinear sensor relations

$\boldsymbol{w}(\cdot)$ vector of sensor uncertainties (noise), covariance matrix $\boldsymbol{W}$

A posteriori state estimate:

$$\hat{\boldsymbol{x}}(k)=\bar{\boldsymbol{x}}(k)+\boldsymbol{K}(k)[\boldsymbol{y}(k)-h(\bar{\boldsymbol{x}}(k))] \tag{4}$$

Predicted (a priori) state estimate:

$$\bar{\boldsymbol{x}}(k+1)=f(\hat{\boldsymbol{x}}(k)) \tag{5}$$

KF gain matrix:

$$\boldsymbol{K}(k)=\bar{\boldsymbol{X}}(k)\boldsymbol{D}^{\mathrm{T}}(k)[\boldsymbol{D}(k)\bar{\boldsymbol{X}}(k)\boldsymbol{D}^{\mathrm{T}}(k)+\boldsymbol{W}(k)]^{-1} \tag{6}$$

Predicted covariance matrix of a priori state estimate:

$$\bar{\boldsymbol{X}}(k+1)=\Phi(k)\hat{\boldsymbol{X}}(k)\Phi^{\mathrm{T}}(k)+\boldsymbol{N}(k) \tag{7}$$

Covariance matrix of a posteriori state estimate:

$$\hat{\boldsymbol{X}}(k)=[\boldsymbol{I}-\boldsymbol{K}(k)\boldsymbol{D}(k)]\bar{\boldsymbol{X}}(k) \tag{8}$$

where

$$\boldsymbol{D}(k)=\partial h(\boldsymbol{x})/\partial \boldsymbol{x}^{\mathrm{T}}\big|_{\boldsymbol{x}=\bar{\boldsymbol{x}}(k)} \quad ; \quad \Phi(k)=\partial f(\boldsymbol{x})/\partial \boldsymbol{x}^{\mathrm{T}}\big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}(k)}$$

The KF gain matrix of (6) is the contents of the block 'updating algorithm' in Fig. 1. One of the major advantages of employing the recursive algorithm of the EKF is that it is very general and allows for the application of an arbitrary number of sensors as long as a sensor model, $h(\quad)$ has been established. If more sensors are employed than strictly necessary from an observability point of view, it is possible to arrive at more accurate estimates of system states and insensitivity to sensor failure. Above, the basic EKF is described. This is often modified to cope with special situations.

3. Event driven state estimators

State estimators are usually described in terms of continuous-time differential equations or by discrete-time difference equations where all measurements are available simultaneously at equally spaced instances in time. Thus, common phenomena such as skewed sampling are neglected. Also, a fixed rate of appearance of data implies that a complete update of the state estimate is made even if little new information is available. The schemes are convenient since their structures are well defined, and conceptually simple means are available for system design and performance analysis. The present section describes means for the systematic treatment of now-and-then appearing sensor data, useful in large-scale sensory systems.

3.1. *Basic concept*

To make is simple, the system below is based on a continuous-time kernel state estimator. This also emphasizes the difference between ballistic estimation and sensor-based updates, and makes it possible to take into account continuously available data as well as data available at specific moments. For digital computer implementations, the continuous-time kernel can be replaced by a fixed-date discrete-time equivalent in the usual manner. It is assumed that the process is represented by the differential equation.

$$\dot{\boldsymbol{x}} = g(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{n}) \tag{9}$$

where $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ the control vector and $\boldsymbol{n}$ the process disturbance. Assuming continuity, the process can be linearized about trajectories $\bar{\boldsymbol{x}}(t)$, $\bar{\boldsymbol{u}}(t)$ and $\bar{\boldsymbol{n}}(t)$ of a state estimator by

$$\dot{\xi} = \boldsymbol{A}\xi + \boldsymbol{B}\eta + \boldsymbol{C}\zeta \tag{10}$$

where

$$\xi = \boldsymbol{x} - \bar{\boldsymbol{x}}, \eta = \boldsymbol{u} - \bar{\boldsymbol{u}}, \zeta = \boldsymbol{n} - \bar{\boldsymbol{n}}$$

and

$$\dot{\bar{\boldsymbol{x}}} = g(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{n}}) \tag{11}$$

Usually $\bar{\boldsymbol{u}} = \boldsymbol{u}$ so that $\eta = 0$, simplifying (10). ξ is the state estimate error whereas ζ is considered to be the white-noise, zero-mean part of $\boldsymbol{n}$. Thus the state estimate covariance develops (Jazwinski, 1970) by

$$\dot{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{C}\boldsymbol{N}\boldsymbol{C}^{\mathrm{T}} \tag{12}$$

where $\boldsymbol{X} = E(\xi\xi^{\mathrm{T}})$, $\boldsymbol{N} = E(\zeta\zeta^{\mathrm{T}})$ are covariance matrices. Continuously available measurements $\boldsymbol{y}_0 = \boldsymbol{D}\boldsymbol{x}_0 + \boldsymbol{w}_0$ will modify the covariance to

$$\dot{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{X}\boldsymbol{D}_0^{\mathrm{T}}\boldsymbol{W}_0^{-1}\boldsymbol{D}_0\boldsymbol{X} + \boldsymbol{C}\boldsymbol{N}\boldsymbol{C}^{\mathrm{T}} \tag{13}$$

where $\boldsymbol{W}_0$ is the covariance of measurement vector $\boldsymbol{y}_0$. These measurements may or may not be present. The important point is that it is possible continuously to keep track of the covariance of the error in the state estimate. Measurements available at discrete points in time are divided into groups $\boldsymbol{y}_1$ through $\boldsymbol{y}_R$ such that when presented,

$$\boldsymbol{y}_1 = \boldsymbol{h}_1(\boldsymbol{x}) + \boldsymbol{w}_1 \quad , \boldsymbol{y}_2 = \boldsymbol{h}_2(\boldsymbol{x}) + \boldsymbol{w}_2 \quad , \ldots \tag{14}$$

and so on. Here $\boldsymbol{w}_i$ is a white-noise, zero-mean disturbance vector for measurement vector $\boldsymbol{y}_i$, with covariance matrix $\boldsymbol{W}_i$. Linearization of an arbitrary measurement group $\boldsymbol{y}_i$ about $\bar{\boldsymbol{y}}_i$ and $\bar{\boldsymbol{x}}$ such that $\bar{\boldsymbol{y}}_i = \boldsymbol{h}_i(\bar{\boldsymbol{x}})$, is represented by

$$\boldsymbol{v}_i = \boldsymbol{D}_i\,\xi + \boldsymbol{w}_i \tag{15}$$

where $\boldsymbol{v}_i = \boldsymbol{y}_i = \bar{\boldsymbol{y}}_i$. It is now assumed that measurement vector $\boldsymbol{y}_i$ is available at time $t(j)$. Vector $\boldsymbol{y}_i$ may or may not be sufficient for the observability of the dynamic system.

Available from (11) is an a priori state estimate $\bar{\boldsymbol{x}}(j)$ with error covariance $\boldsymbol{X}(j)$ given by (12) or (13) ($\bar{\boldsymbol{x}}(j)$ and $\boldsymbol{X}(j)$ are short for $\bar{\boldsymbol{x}}(t(j))$ and $\boldsymbol{X}(t(j))$). The ordinary minimum variance estimate $\hat{\boldsymbol{x}}(j)$ and $\boldsymbol{y}_i(j)$ is then

$$\hat{\boldsymbol{x}}(j) = \bar{\boldsymbol{x}}(j) + \boldsymbol{K}_i(j)[\boldsymbol{y}_i(j) - \boldsymbol{h}_i(\bar{\boldsymbol{x}}(j))] \tag{16}$$

where

$$\boldsymbol{K}_i(j) = \boldsymbol{X}(j)\boldsymbol{D}_i^{\mathrm{T}}[\boldsymbol{D}_i\boldsymbol{X}(j)\boldsymbol{D}_i^{\mathrm{T}} + \boldsymbol{W}_i]^{-1} \tag{17}$$

Similar to (8), the error covariance of this estimate is given by

$$\hat{\boldsymbol{X}}(j)=[\boldsymbol{I}-\boldsymbol{K}_i(j)\boldsymbol{D}_i]\boldsymbol{X}(j) \tag{18}$$

When these computations have been made, $\boldsymbol{X}(j)$ in (12) or (13) is reinitiated by $\hat{\boldsymbol{X}}(j)$ whereas $\bar{\boldsymbol{x}}(j)$ is set to $\boldsymbol{x}(j)$ before the continuous updating resumes. In this way, means are available for correcting the state estimator at arbitrary instances $t(j)$ based on a part $\boldsymbol{y}_i(j)$ of the complete measurement vector.

3.2. *Covariance based data selector*

Future robots and autonomous vehicles are assumed to have installed large numbers of very different sensors. The number and diversity can, when considering the computation and data transmission capacities, imply that only a lesser number of sensors can be handled at a sufficient regularity. The problem is then to select those measurements which result in a sufficient quality for the continuously updated state estimate. The relevant system states should at least be observable, but a better quality measure can result from the covariance matrix $\boldsymbol{X}$ itself.

$$J=E(\xi^{\mathrm{T}}\xi)=\mathrm{trace}\boldsymbol{X} \tag{19}$$

is a simple example. Due to measurements $\boldsymbol{y}_i(j)$ it is altered by

$$\Delta J_i=\mathrm{trace}[\hat{\boldsymbol{X}}(j)-\boldsymbol{X}(j)] \tag{20}$$

which can be computed using (17) and (18).

The covariance based data selector is intended for on-line use in large-scale sensory systems. However, off-line modifications have their use as well, for instance as the means for inferring the necessary sample rates of each group of measurements in medium and small-scale control systems. It is the authors' opinion that, in general, the sample rates for measurements should be considered as a separate problem and without reference to the rates for setting of the control values.

3.3. *A simple servo*

As an example, a servo is assumed which is controlled using analogue computation, or at a high sample rate. However, positional measurements are available at a low rate only. As indicated in Fig. 2, the state estimator models the mechanism as three cascaded integrators, and takes feedback from the continuous state estimate. This approach

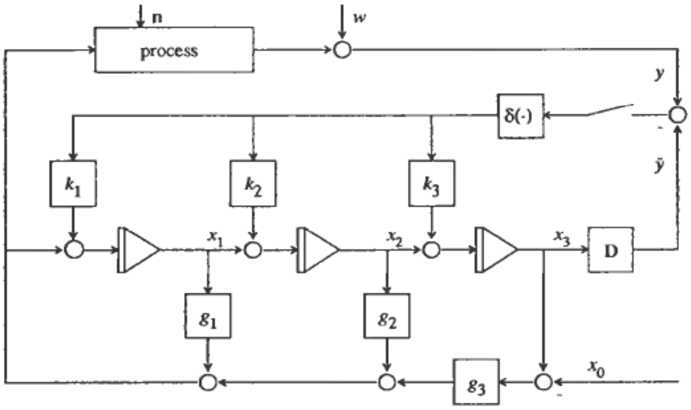

Figure 2. Servo control system.

allows very high feedback gains $[g_1, g_2, g_3]$, resulting in a high-performance servo. Using a traditional controller, where feedback is taken directly from positional measurement, the performance is inherently bounded due to the low rate of appearance of y. The presented estimation and control structure is usable for instance in camera-guided autonomous vehicles as well as in high-performance servo mechanisms. An application within 3D vision is presented below.

4. Camera based estimation of 3D motion

The Kalman Filter can be used for sequential estimation of position and motion parameters of a rigid body based on camera measurements (Broida, 1986; Dickmanns, 1985). In this section the stereoscopic camera system of Fig. 4 is used for estimating motion in 3D space. The positions of the measured features in the pictures are compared to the predicted positions, and the differences are used to refine the estimates. Only a small set of simple, easy-extractable features, in needed. These can be assumed noisy, and the uncertainty is reduced in a least squares fashion in the Kalman Filter. The feature extraction time can also be reduced, because the Kalman Filter not only gives a prediction of the position and motion parameters, but provides a representation of the uncertairty associated with the estimates as well. The method is not restricted to a stereoscopic camera system. More cameras and other sensors can be integrated in the state estimator by just expanding the Kalman measurement vector. The measurements don't even have to come at the same time, which shall be shown later in this section.

One application for this type of camera system is direct and flexible programming of a robot based on monitoring an experienced operator moving the tool (e.g. spray-painting).

4.1. *The measurement model*

The camera model is that of central projection. The camera coordinate system (x^c, y^c, z^c) has the image plane coincident with the x-y plane and the optical axis along the z-axis. The centre of the image plane is at $(0, 0, -\lambda)$, and the centre of lens is at the origin. Here it is assumed for simplicity that the camera coordinate system is aligned with the world coordinate system (x^0, y^0, z^0). The image plane coordinates (x^c, y^c) of the projected 3D point (x^0, y^0, z^0) are

$$x^c = \frac{-\lambda x^0}{z^0} \tag{21}$$

$$y^c = \frac{-\lambda y^0}{z^0} \tag{22}$$

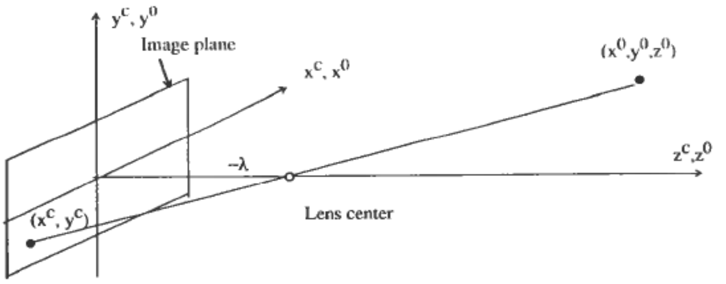

Figure 3. Basic model of the imaging process. Camera coordinate system (x^c, y^c, z^c) is aligned with the world coordinate system (x^0, y^0, z^0).

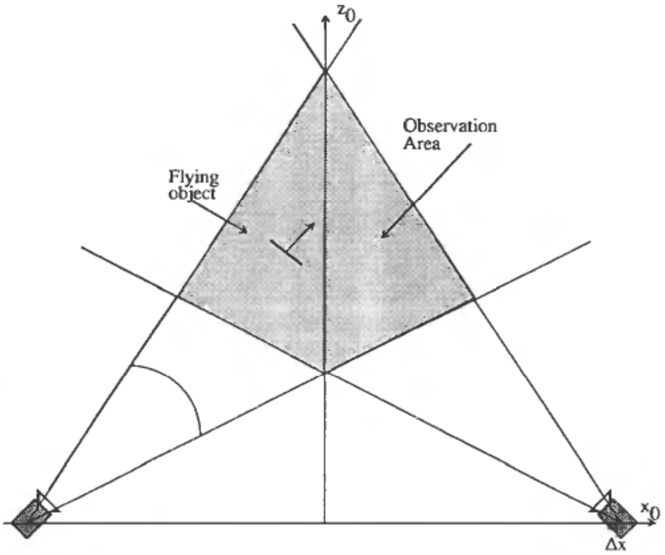

Figure 4. Camera configuration.

The measurement equations are nonlinear which means that the Extended Kalman Filter must be used. The measurements are linearized about the most recent estimate of the object position. The new D-matrix is

$$\boldsymbol{D}_k = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}^{\mathrm{T}}}\right|_{x^{2}\bar{x}_k} \tag{23}$$

The two cameras are placed as in Fig. 4 at $(\Delta x,\ 0, 0)$ and $(-\Delta x, 0, 0)$. The two camera axes cross in the middle of the observation area, (at $z = x_o$) so that depth resolution is increased.

The measured features are the camera coordinates of two matchpoints on the observed object, which give four measurements in each camera, a total of eight measurements. How these coordinates are deduced from the camera image is not a central point in this paper. It might be extracted simply by the knowledge of intensity value of the endpoints in the images, or the gradient in the discrete image function around the endpoints. This can be further simplified by placing light points (e.g. LEDs) at the matchpoints. The two cameras don't have to measure the same points on the object, so the correspondence problem is not dealt with in this strategy. As pointed out earlier, the Kalman Filter gives a prediction of the measurements with an associated uncertainty. This should be used to reduce the search area in the image and thus speed up the feature extracting algorithms. The measurements are assumed noisy. The physical sources of noise are quantization error, calibration error and vibrations in the equipment.

4.2. *Motion and kinematics*

The camera system is used to predict the motion of a free flying rigid object. The state-space model can be developed from Newton's law for the translation, and for Euler's law for the rotation around the centre of gravity. An inertial base frame 0 is used to describe the motion of the rigid body. The centre of mass has the position $\boldsymbol{p}$ with respect to the origin in the inertial frame. The velocity of the centre of gravity is denoted

$\boldsymbol{v}$, while the angular velocity vector with respect to the inertial frame is ω. A superscript 0 is used to denote decomposition of a vector in the base frame, while decomposition in the body frame is denoted by superscript b. The equations of motion are

$$\ddot{\boldsymbol{v}}^0 = 0 \tag{24}$$

In the special case that the body is symmetric around its centre of gravity, the equation of angular motion is simply

$$\ddot{\omega}^0 = 0 \tag{25}$$

The differential equation for the position vector $\boldsymbol{p}$ is

$$\dot{\boldsymbol{p}} = \boldsymbol{v}^0 \tag{26}$$

The angular velocity, however, cannot be integrated to an orientation vector in the same way. Classical Euler angles describe orientation through the rotations ψ around the z axis, then θ around the y axis, then finally a rotation ϕ around the rotated z axis. The kinematic differential equation is found from

$$\omega^0 = \begin{bmatrix} \sin\theta\cos\psi & -\sin\psi & 0 \\ \sin\theta\sin\psi & \cos\psi & 0 \\ \cos\theta & 0 & 1 \end{bmatrix} \dot{\boldsymbol{r}}_{\boldsymbol{E}} \tag{27}$$

where

$$\boldsymbol{r}_{\boldsymbol{E}} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{28}$$

The desired inverse relation is

$$\dot{\boldsymbol{r}}_{\boldsymbol{E}} = \boldsymbol{C}_{\boldsymbol{E}}\,\omega^0 \tag{29}$$

where

$$\boldsymbol{C}_{\boldsymbol{E}} = \frac{1}{\sin\theta} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\theta\sin\psi & -\sin\theta\cos\psi & 0 \\ -\cos\theta\cos\psi & -\sin\psi\cos\psi & 1 \end{bmatrix} \tag{30}$$

The description has a singularity at $\sin\theta = 0$.

An alternative set of Euler angles are the roll-pitch-yaw angles: a rotation α around the z axis, β around the y axis, and finally γ around the x axis. The inverse transformation is

$$\dot{\boldsymbol{r}}_{\boldsymbol{R}} = \boldsymbol{C}_{\boldsymbol{R}}\,\omega^0 \tag{31}$$

where

$$\boldsymbol{r}_{\boldsymbol{R}} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \tag{32}$$

In this case the singularity appears at $\cos\beta = 0$.

A singularity-free description is obtained by using roll-pitch-yaw angles when $|\beta| < = \pi/4$ and switch to classical Euler angles when $|\theta| > \pi/4$.

When classical Euler angles are used, the state vector is

$$\boldsymbol{x}=[\boldsymbol{p}\boldsymbol{r}_{\boldsymbol{E}}\boldsymbol{v}^0\omega^0\dot{\boldsymbol{v}}^0\dot{\omega}^0]^{\mathrm{T}} \tag{33}$$

The state-space model is then

$$\dot{\boldsymbol{x}}=\boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} \tag{34}$$

where

$$\boldsymbol{A}(\boldsymbol{x})=\begin{bmatrix} 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{C}_{\boldsymbol{E}} & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{35}$$

When θ becomes $\pi/4$, a switch is made to

$$\boldsymbol{x}=[\boldsymbol{p}\boldsymbol{r}_{\boldsymbol{R}}\boldsymbol{v}^0\omega^0\dot{\boldsymbol{v}}^0\dot{\omega}^0]^{\mathrm{T}} \tag{36}$$

$$\boldsymbol{A}(\boldsymbol{x})=\begin{bmatrix} 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{C}_{\boldsymbol{R}} & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{37}$$

4.3. *Simulation results*

The simple object of Fig. 5 is to be used in the simulation run. It is symmetric about its major axis. The length is 0·3 m. The measurements come from the stereoscopic camera system of Fig. 4, where Δx is 2·5 m. The cameras have a resolution of 512 by 512 pixels and a field of view of 30° with a focus length of $\lambda=0{\cdot}016$ m. The matchpoints are the two end points of the object. These measurements are distorted with gaussian random noise. The noise level has standard deviation of 5 pixels which corresponds to 5–7 percent of the picture of the rod in the simulation runs.

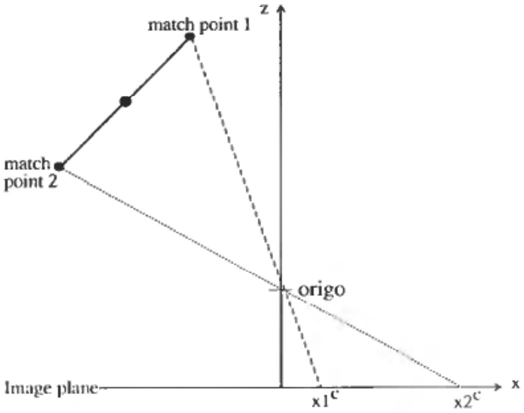

Figure 5. Object and measurement model.

The initial values of the simulator were: $\boldsymbol{p}=[0,0,2{\cdot}5]^T$, $\boldsymbol{v}=[-1,-1,-1]$, $\dot{\boldsymbol{v}}=[0{\cdot}75,0{\cdot}75,0{\cdot}75]$. In the simulation run the object is rotating in the x-y plane, with initial angular velocity and acceleration of $-1{\cdot}0$ rad/s and $0{\cdot}75$ rad/s^2. The initial estimates are different from the true values and are therefore associated with a low certainty, reflected by a large value in the covariance matrix. Assuming 30 pictures per second, the sample interval is $T=1/30$ s. Two different results are presented, differing in the picture sampling. In the first simulation, measurements come from both cameras simultaneously at a rate of 30 pictures per second. In the second simulation the first measurements come from camera 1, the second from camera 2, then again from camera 1 and so on. The data rate is in this manner reduced to the half of that in the first simulation.

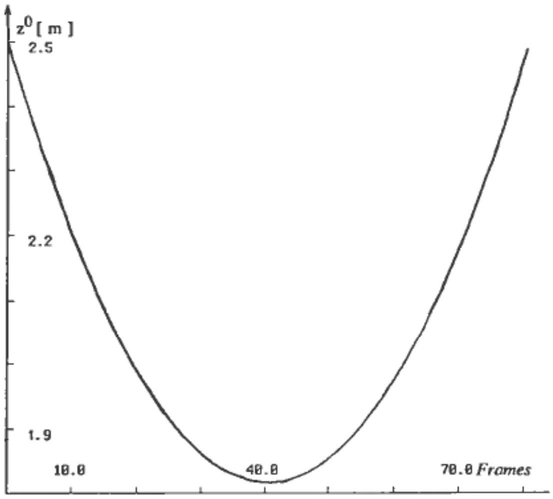

Figure 6. Translation in z-direction (depth) of the objects centre of mass in both simulation runs. Prediction error is shown in Fig. 7. 30 image frames are provided per second.

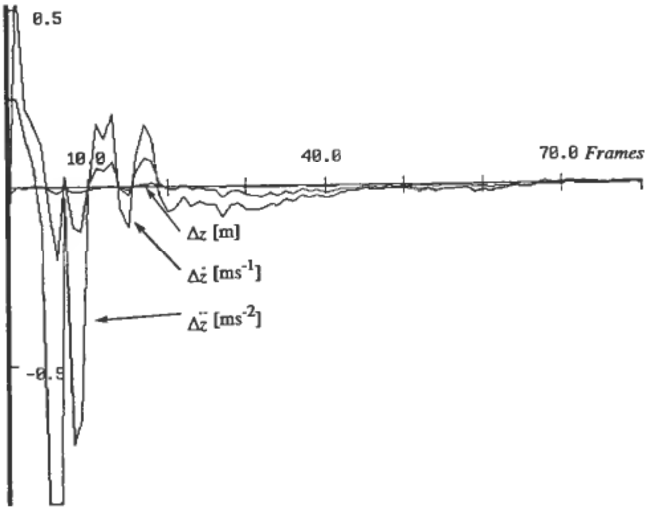

Figure 7. Error in predicted position, velocity and acceleration of the centre of mass of the object (the z-component) in the first simulation run. All estimates converge. Position converges the fastest. 30 picture frames/sec.

Some results from the first simulation run are presented in Figs. 6–8. The measurements come from the two cameras simultaneously. All the state variables converge quickly to their correct values. Ths position parameters, which are the most important, converge very fast, and their covariances converge to a low value. The error in z^c improves to less than 1 cm which is about 1 or 2 pixels in a camera. Figure 8 shows that the error in the camera measurements are reduced from a standard deviation of 5 pixels to a standard deviation of 1–2 pixels.

The result from the second simulation run, where measurements from only one camera at the time are available, is presented in Fig. 8. This is a special case of the theory

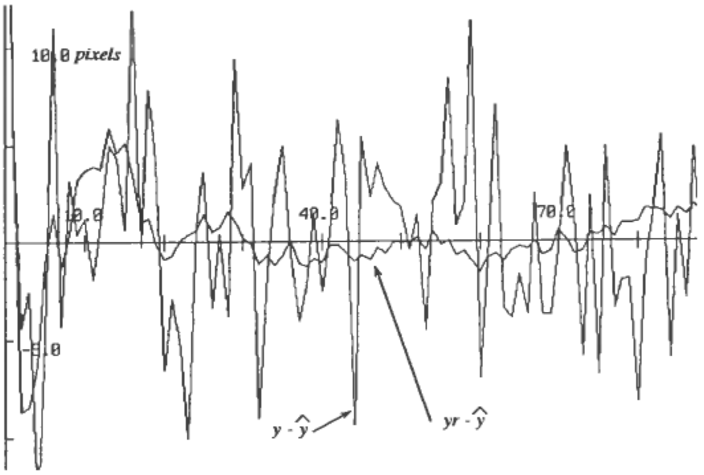

Figure 8. From simulation 1. Innovation process $\varepsilon = \hat{y} - y$ and $y_r - \hat{y}$, where $y = y_r + w$ (*w*: the noise term). Predicted measurement $\hat{y}$ is very close to the noise-free measurement y_r. It differs with only 1–2 pixels. Difference between the noisy measurements y and the predicted measurements $\hat{y}$ is almost uncorrelated.

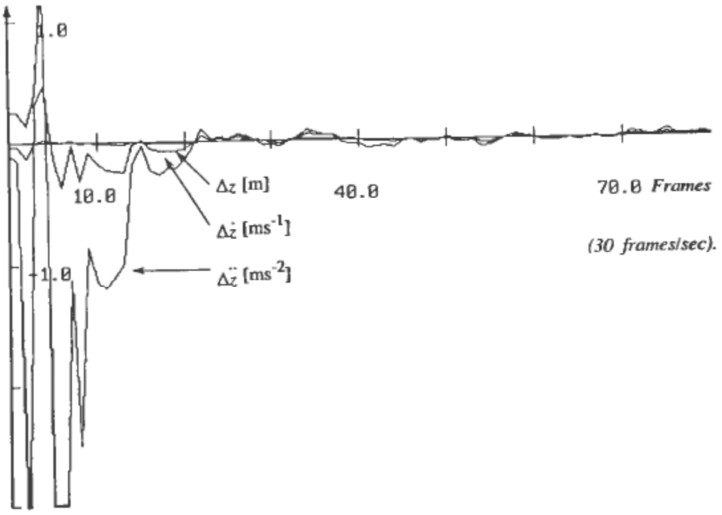

Figure 9. Error in predicted position, velocity and acceleration of the centre of mass of the object (*z*-component) when measurements from only one of the two cameras are available at each instance (simulation 2). The figure corresponds to Fig. 7. The state estimator needs a few more frames to converge, compared to simulation run 1, Fig. 7.

presented in § 3, where the measurements could be available at arbitrary instances $t(j)$. The state estimator is updated according to the available part of the measurement vector, $\boldsymbol{y}_i(j)$ of the complete vector $\boldsymbol{y}$.

The result shows that the state estimator still works well. Slower convergence is expected since the number of measurements are only half of the number used in the first simulation.

5. Conclusion

It has been shown that applying the Kalman Filter algorithm is a powerful and general technique for integrating sensors. This has been demonstrated for the task of inverting primary camera data in order to establish estimates of coordinate values, as well as of the velocity and acceleration of a moving object in 3D space. The KF can, using simple means, integrate data from any number of possibly very different sensors. Also, sample rates and the time of appearance of the information need not be synchronized. Feedback to a control system is made available without direct reference to the sensors.

Parts of the work was sponsored by the Royal Norwegian Council for Scientific and Industrial Research, on contract TU 22301.

References

Broida, T. J. and Chellappa, R. (1986). Estimation of Object Motion Parameters from Noisy Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **8,** 90–98.

Dickmanns, E. D. and Zapp, A. (1985). Guiding Land Vehicles along Roadways by Computer Vision. *Proc. Congrès Automatique 1985, The Tools for Tomorrow*, Toulouse, 233–244.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory* (Academic Press, New York).