ISSN 0005-1144
ATKAFF 51(3), 264–274(2010)

Josip Musić, Mojmil Cecić, Vlasta Zanchi

Real-Time Body Orientation Estimation Based on Two-Layer Stochastic Filter Architecture

UDK 004.932.75'1:681.586.7
IFAC 4.2.2; 2.8

Preliminary communication

The article presents real time rigid body orientation estimation using inertial and magnetic sensors. Based on the review of orientation estimation literature we suggest, as possible alternative to standard approaches, novel two-layer stochastic estimation filter architecture based on Kalman and particle filters combined into two layers. Two-layer architecture was chosen because it enables greater applicability via upgrade of already implemented Kalman or particle filters. Four two-layer filter architectures were designed, each one enabling different layer interaction. Estimation of human head orientation was chosen as a case example. Simulated data and batch head orientation measurement data were used to test the proposed architectures in terms of accuracy and computational efficiency and to select the best one in terms of aforementioned performance parameters. Selected architecture was then implemented in real time for human-computer interaction and was tested on several practical applications. Obtained results are presented and discussed and future research directions suggested.

Key words: Estimation, Orientation, Kalman filter, Particle filter, Two-layer architecture

Procjena orijentacije tijela u realnom vremenu uporabom dvorazinske stohastičke arhitekture estimacijskog filtra. Članak obrađuje postupke estimacije prostorne orijentacije krutog tijela u realnom vremenu temeljene na mjerenjima inercijskih i magnetskih senzora. Pregledom standardnih pristupa estimaciji orijentacije, kao moguća alternativa predložena je nova arhitektura stohastičkog estimacijskog filtra temeljena na kombiniranju Kalmanovog i čestičnog algoritma u dvije razine. Arhitektura dvorazinske prirode je uporabljena jer omogućava veću primjenjivost nadogradnjom već implementiranih Kalmanovih ili čestičnih filtra. U radu su projektirane četiri dvorazinske arhitekture koje su različitim kombinacijama ostvarivale interakciju među razinama. Pri tome je kao ogledni primjer uporabljena estimacija orijentacije glave čovjeka. Izvedena su ispitivanja točnosti i računske efikasnosti kako na simuliranim tako i na stvarnim mjernim podacima u off-line režimu rada, te je odabrana ona arhitektura s najboljim rezultatima. Odabrana arhitektura je zatim implementirana u realnom vremenu pri međudjelovanju čovjeka s računalom, te je ispitana na nekoliko praktičnih primjena. Na temelju ostvarenih rezultata su izvedeni zaključci te predložena moguća poboljšanja, te daljnji pravci istraživanja.

Ključne riječi: estimacija, orijentacija, Kalmanov filtar, čestični filtar, dvorazinska arhitektura

1 INTRODUCTION

In the field of human motion analysis accurate knowledge of position and orientation of individual body segments is of interest [1–3]. Usually optoelectronic methods are used for data acquisition. Although very accurate they are cumbersome, limited to laboratory setting, expensive, have limited field of view and have occlusion issues. Thus, body worn sensors, like accelerometers and gyroscopes, based on MEMS technology have recently received a lot of attention [1–5]. The main part of inertial sensor packs is estimation algorithm since individual sensors have shortcomings like bias in gyroscope output or accelerometer sensitivity to vibrations [2, 6–8]. Kalman filters and its variants like extended Kalman filter (EKF) are somewhat of a standard in human motion analysis due to their good accuracy and computational efficiency [1–3,7,9,10]. When working with nonlinear systems EKF operation is based on linearization around current working point by expansion in Taylor like series. While this kind of approach is effective for those systems that do not have severe nonlinearities or operate in limited range where nonlinearities are not severe, it can produce poor results for cases when nonlinearities are severe. Also, Kalman filters base their operation on assumption that process and measurement noises are distributed according to Gaussian distribution which is not always the case. Additionally lower EKF sampling rate can result in filter divergence [10]. In contrast to EKF, particle filters (PF) do not require linearization step and can successfully cope with different types of noise distributions. They are based on Bayesian inference and Monte Carlo sampling methods [11–14] which use number of samples (or particles) to numerically approximate *a posteriori* distribution. The higher the number of particles the better the accuracy but with higher computational cost. While computational cost still presents an important issue, with improvements in the algorithm like Rao-Blackwellization [12, 14] and with ever increasing availability of computer power, it is becoming less important. Good review of particle filters can be found in [11, 15]. It should be noted that there exists another Kalman filter variant called Unscented Kalman filter (UKF) [10] which is not routinely used in human motion analysis and whose cost effectiveness in human orientation tracking has been questioned [16]. Thus it was not considered in this work. Comparison between EKF and PF (with $N = 200$ particles) performance in terms of accuracy and computational complexity on highly nonlinear system of equations found in [10] is depicted in Fig. (1). The figure clearly illustrates superior accuracy of PF compared to EKF but with about 1000 times longer execution time.

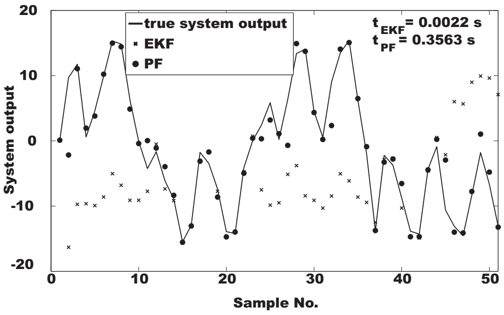

Fig. 1. Comparison between EKF and PF performance

From the previous discussion it can be concluded that combining EKF and PF could result in more reliable human motion analysis with positive/desirable characteristics of individual algorithms. Additionally EKF results can be adequate in number of cases when nonlinearities are not severe, while PF-level accuracy could be required in just some cases. Thus, we argue, that design of architecture which would employ both the EKF and PF algorithms would present a compromise between individual algorithms in terms of accuracy and computational complexity while enabling use of just one of the algorithms when needed. In the paper basic versions of two estimation algorithms are used and we note that application of more advanced algorithm variants (e.g. adaptive particle and Kalman filters) could result in improved performance. But since the gist of the paper is in comparison between stand alone algorithms and two-layer architecture, more advanced (and thus more complex) algorithms would, in authors opinion, be unpractical. In literature approach similar to one presented here can be found in [17], but without tight coupling between EKF and PF algorithms. Combination of EKF and PF algorithms can also be used for better estimation of *a posteriori* distribution in Bayesian inference [10, 14]. Expectation-maximization (EM) algorithms [18] can also use combination of EKF and PF algorithms in their operation.

The paper focuses on different ways by which EKF and PF interaction can be achieved in two-layer architecture, with case example of head orientation tracking for the purpose of improved human-computer interaction. The article is structured as follows. In the next section the problem of EKF and PF interaction in two-layer architecture is considered and four possible solutions proposed. In Section 3 proposed architectures are tested on simulated and real data sets. Several applications of human-computer interaction based on two-layer estimation architecture are considered in Section 4, while conclusions are drawn and future research directions are suggested in Section 5.

2 TWO-LAYER STOCHASTIC FILTER ARCHITECTURE

In order to design two-layer stochastic filter architecture, two main building blocks (i.e. EKF and PF) need to be defined. Since the basis for both blocks is (head) motion model it also needs to be defined. Modeling of human head motion due to several complex joints is not a trivial task [19]. Additionally, choice of head orientation notation has an effect on model performance and complexity [20]. In our research we chose Euler angles and their time propagation model since they are simple and computationally effective. The problem of singularities associated with Euler angles is eliminated by the fact that normal human range of motion of cervical spine due to anatomical constraints does not permit such situations [21]. Time propagation of Euler angles is well documented in literature [20]. It uses current angles in global coordinate frame and angular velocities in body coordinate frame to propagate orientation angles via integration of angular velocities in global coordinate frame. Since gyroscopes which measure angular velocities suffer from accuracy issues related to existence of temperature dependent bias it makes sense to include biases in the model, thus minimizing/eliminating their influence. Resulting model equations are

$$\begin{aligned}\dot{\phi} &= [(\omega_y - \delta\omega_y)\sin(\phi(t)) + (\omega_z - \delta\omega_z) \\ &\quad \cos(\phi(t))]\tan(\theta(t)) + (\omega_x - \delta\omega_x)\end{aligned} \tag{1}$$

$$\dot{\theta} = (\omega_y - \delta\omega_y)\cos(\phi(t)) + (\omega_z - \delta\omega_z)\sin(\phi(t)) \tag{2}$$

$$\begin{aligned}\dot{\psi} &= [(\omega_y - \delta\omega_y)\sin(\phi(t)) + \\ &\quad +(\omega_z - \delta\omega_z)\cos(\phi(t))]\sec(\theta(t))\end{aligned} \tag{3}$$

where ϕ (roll), θ (pitch) and ψ (yaw) are head orientation angles defined in Fig. (2), $\delta\omega_x$, $\delta\omega_y$ and $\delta\omega_z$ are gyroscope biases for respective rotation axis, while ω_x, ω_y and ω_z are rotational velocities in body coordinate frame.

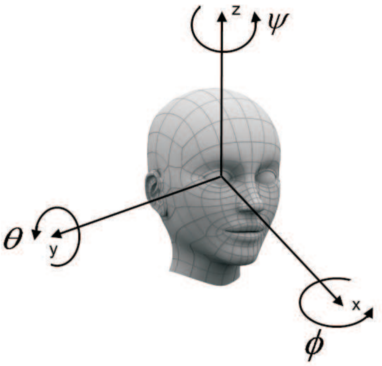

Fig. 2. Definition of head orientation angles

2.1 Layer 1 - Extended Kalman filter

From equations (1), (2) and (3) it can be seen that motion model is nonlinear, thus EKF needs to be used. EKF algorithm is performed in two steps: time and measurement update. Time update is based on the motion model and is in general executed once per sampling time. In the paper it was defined in continues domain by (1), (2), (3) and

$$\dot{\mathbf{P}} = \mathbf{A}\mathbf{P}^- + \mathbf{P}^-\mathbf{A} + \mathbf{Q} \tag{4}$$

where $\mathbf{A} = \frac{\partial f}{\partial x}$ is input Jacobian matrix, $\mathbf{P}^-$ is *a priori* estimate covariance matrix and $\mathbf{Q}$ is process noise covariance matrix. Measurement update is executed at those times instances when measurement(s) become available which in general doesn't have to match time update sampling rate. In our research, sampling rates of time and measurement updates were the same. Measurement update equations usually found in the literature [10] are of the form

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{5}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{y}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-, t_k)] \tag{6}$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{7}$$

where $\mathbf{P}_k^+$ is *a posteriori* estimate covariance matrix, $\hat{\mathbf{x}}_k^+$ is *a posteriori* estimate, $\hat{\mathbf{x}}_k^-$ is *a priori* estimate, $\mathbf{K}_k$ is Kalman gain, $\mathbf{y}_k$ is measurement vector, $\mathbf{h}_k$ is measurement function relating measurements to system states, $\mathbf{H}_k$ is Jacobian matrix of $\mathbf{h}_k$ in respect to $\mathbf{x}_k$ and $\mathbf{R}_k$ is measurement noise covariance matrix. These equations demonstrated numerical instability i.e. Kalman gain could not be calculated due to appearance of singularity in the inverse member of equation (5) after several iterations. Thus alternative form of the equation was used [10]

$$\mathbf{K}_k = \mathbf{P}_k^+\mathbf{H}_k^T\mathbf{R}_k^{-1} \tag{8}$$

Due to explicit dependence of equation (7) on $\mathbf{P}_k^+$ which is now part of equation (8) alternative form of equation (7) also had to be used [10]

$$\mathbf{P}_k^+ = [(\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k]^{-1} \tag{9}$$

Equations (8) and (9) were used throughout subsequent testing without any singularity issues or numerical instability. It is interesting to note that hybrid version of Kalman filter was used since time update is defined in continues domain while measurement update is defined in discrete domain.

Measurement vector $\mathbf{y}_k$ consisted of indirect measurements of head orientation angles. Roll and pitch angles were calculated from accelerometer measurements using equations

$$\theta_{ac} = -\arcsin\left(\frac{a_{ccX}}{g_T}\right) \tag{10}$$

$$\phi_{ac} = \arcsin\left(\frac{a_{ccY}}{g_T\cos(\theta_{ac})}\right) \tag{11}$$

where a_{ccX} and a_{ccY} are accelerometer measurements along x and y axis, while g_T is total measured acceleration defined as $g_T = \sqrt{a_{ccX}^2 + a_{ccY}^2 + a_{ccZ}^2}$. Note that g_T should (ideally) equal gravitational acceleration constant g but in order to avoid trigonometric function domain issues we chose g_T to be as defined before. The yaw angle of rotation was calculated from (tilt compensated) magnetometer measurements as

$$\psi_{mag} = \frac{-m_y c(\phi_{ac}) + m_z s(\phi_{ac})}{m_x c(\theta_{ac}) + m_y s(\theta_{ac}) s(\phi_{ac}) + m_z c(\phi_{ac}) s(\theta_{ac})} \tag{12}$$

where m_x, m_y and m_z are magnetometer measurements in local coordinate frame, c is cosine and s sine function. It should be noted that orientation angles calculated from accelerometer measurements are used to compensate for tilt in magnetometer measurements. Gyroscope measurements are not part of measurement vector $\mathbf{y}_k$ since they are included in the model equations (1), (2) and (3). The complete algorithm can be seen in Algorithm (1).

Algorithm 1 Extended Kalman filter used in the paper

```
Input: $\hat{\mathbf{x}}_0^+$, $\mathbf{P}_0^+$, $\mathbf{Q}$, $\mathbf{R}_0$
Output: $\hat{\mathbf{x}}_k^+$, $\mathbf{P}_k^+$
for $k = 1$ to $\infty$ do
    $\dot{\hat{\mathbf{x}}}_k^- = f(\hat{\mathbf{x}}_{k-1}^+, u, t)$
    $\dot{\mathbf{P}}^- = \mathbf{A}\mathbf{P}_{k-1}^+ + \mathbf{P}_{k-1}^+\mathbf{A} + \mathbf{Q}$
    $\mathbf{K}_k = \mathbf{P}_k^+\mathbf{H}_k^T\mathbf{R}_k^{-1}$
    $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{y}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-, t_k)]$
    $\mathbf{P}_k^+ = [(\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k]^{-1}$
end for
```

2.2 Layer 2 - Particle filter

Particle filter operation is based on Bayesian inference with prediction-correction steps much like EKF [13, 14]. The prediction step in PF is based on Chapman-Kolomgorov equation

$$p(x_k \mid y_{k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid y_{k-1}) dx_{k-1} \tag{13}$$

while correction step is defined with equation

$$p(x_k \mid y_k) = \frac{p(y_k \mid x_k) p(x_k \mid y_{k-1})}{\int p(y_k \mid x_k) p(x_k \mid y_{k-1}) dx_k} \tag{14}$$

Since equations (13) and (14) can be analytically solved just for few special cases, Monte Carlo sampling methods are used. Ideal Monte Carlo sampling can not be implemented in practical applications since *a posteriori* distribution $p(y_k \mid x_k)$ needs to be known [14]. Instead, sampling is based on importance density function where relative importance is assumed. There are several different methods for selection of importance density function: optimal, transitional, measurement, etc. Arguably the most used importance density function (and the one used in the paper) is transitional density function in which importance weights associated with each particle are calculated based on prediction step i.e. model output $p(x_k \mid x_{k-1})$ [14]. The downside of such approach is that the currently available measurements are not considered for sample generation resulting in reduced accuracy.

In the paper particle filter was implemented in sequential importance sampling with resampling (SIR) form. Resampling was applied at every iteration resulting in worst case scenario in terms of computational cost of the algorithm. Better computational efficiency could be achieved if resampling would be executed only at those time instances when effective sample number exceeds predefined threshold [12–14]. Resampling method used was systematic resampling since compared to other two widely used resampling techniques (multinomial and residual) it is computationally more effective and has lower weight variance [14,22]. Sample impoverishment is undesirable effect which is a byproduct of resampling step and which can result in undesirable algorithm performance. In the paper a simple approach called roughening was used [14]: after the resampling and before the beginning of next iteration certain amount of noise was added to particles in order to increase their diversity.

Number of particles in PF play an important role both in accuracy and in computational efficiency. Thus, testing was done to identify dependency between accuracy and particle number as well as computational complexity and particle number. Obtained results are depicted in Fig. (3). These results suggest that $N = 1000$ would be a good choice since after that accuracy (in all three orientation angles) does not improve significantly while computational cost rises quickly.

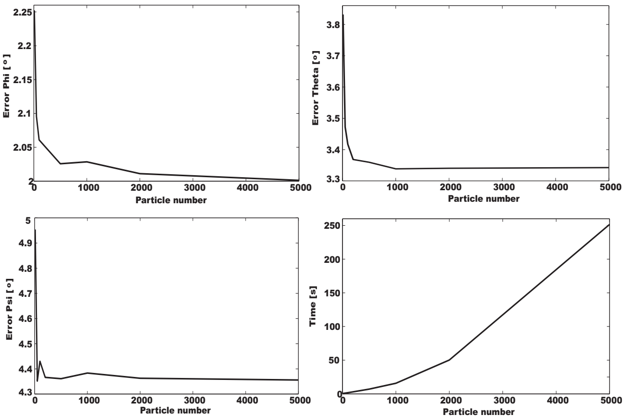

Fig. 3. Influence of particle number on accuracy and computational cost

The complete PF algorithm used in the paper can be seen in Algorithm (2).

Algorithm 2 Particle filter used in the paper

```
Input: $p(\mathbf{x}_0)$, $w_0^i = \frac{1}{N}$
Output: $p(\mathbf{x}_k \mid \mathbf{y}_k)$
for $k = 1$ to $\infty$ do
    $\mathbf{x}_k^i \sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$
    $\mathbf{x}_{0:k} = \{\mathbf{x}_{0:k-1}, \mathbf{x}_k\}$
    $w_k^i \sim p(\mathbf{y}_k \mid \mathbf{x}_k^{(i)})$
    Systematic resampling to get new sample set $x_k^{(i)}$
    Roughening
end for
```

2.3 Two-layer interaction

In the proposed approach each layer contains only one estimation algorithm: first layer houses EKF while the second houses PF. The way in which these two layers interact is (as subsequent testing showed) critical for achieving good performance both for accuracy and computational efficiency. Four interaction schemes were designed and tested. For the experiments XSens MTx inertial sensor pack[1] with XBusMaster was used for data acquisition and as a referent measurement device for head orientation measurements[2]. Interaction schemes were as follows:

- SCHEME "A": The layer 1 containing EKF algorithm is working at the desired sampling rate (e.g. 50

[1] http://www.xsens.com/en/general/mtx

[2] static accuracy $< 1^o$; dynamic accuracy 2^o RMSE depending on movement type.

Hz), while layer 2 containing PF algorithm works at the reduced system sampling rate (in our experiments it was set to $\frac{1}{10}$ of EKF sampling rate). Layer interaction is achieved only at those moments at which PF layer is active in a way that results from PF are presented to EKF as additional measurements. This is possible because mean value and standard deviation of PF estimate can be calculated based on all particle values. This means that PF can affect EKF performance, but EKF performance has no effect on PF. Estimated system output is output from layer 1 (i.e. EKF output). This interaction scheme is graphically depicted in Fig. (4).

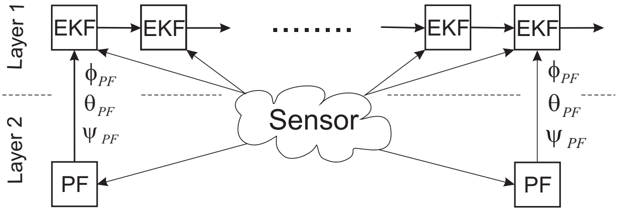

Fig. 4. Between layer interaction according to scheme A

- SCHEME "B": In this interaction scheme layer 1 containing the EKF algorithm is working with $(f_{sampling} - 1)$ sampling rate while layer 2 is active at those time instances when layer 1 is not active. Here both layers influence each other so a good hand-off procedure is needed. When layer 2 is finished with its estimation and layer 1 starts, transition occurs much in the same way as in scheme A. Transition from layer 1 to layer 2 is somewhat different. Here, in order to reduce computational burden of PFs in layer 2, gyroscope biases are not considered in that layer but are rather "tunneled" to the next layer 1 iteration. The (reasonable) assumption made here is that due to relatively high architecture sampling rate (25 Hz - 50 Hz) biases can not change significantly in that short time. The transition itself occurs in a way that outputs from layer 1 (estimate $\hat{\mathbf{x}}_k^+$ and covariance matrix $\mathbf{P}_k^+$) are used to generate required number of particles according to Gaussian distribution. Estimated system output is output from layer 1 or layer 2 depending which one is active at particular time instance. This interaction scheme is graphically depicted in Fig. (5).
- SCHEME "C": This interaction scheme is similar to scheme B and was motivated by the desire to explore if more complex version of scheme B, although more computationally demanding, would be more accurate. Transition procedure between layer 1 and layer 2 was modified in a way that each particle from layer 2 had its own EKF filter in layer 1 (i.e. N particles in layer 2 = N EKF filters in layer 1). This meant that calculation of mean and standard deviation for all particles (when transitioning from layer 2 to layer 1) as well as generation of particle values based on EKF estimate (when transitioning from layer 1 to layer 2) was no longer necessary. Bias "tunneling" feature from scheme B was also applied. Estimated system output is the same as in scheme B. This interaction scheme is graphically depicted in Fig. (6).

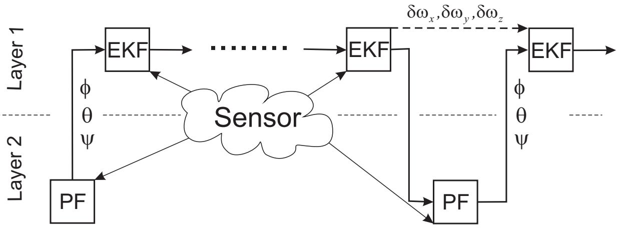

Fig. 5. Between layer interaction according to scheme B

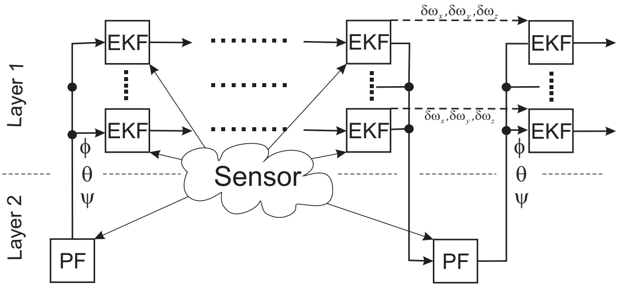

Fig. 6. Between layer interaction according to scheme C

- SCHEME "D": This interaction scheme is inverse version of scheme A. Layer 1 is working at the system sampling rate as was case in scheme A. Layer 1 now has effect on layer 2 while layer 2 does not have any direct influence on layer 1. Estimated system output is output from layer 2 meaning that this interaction scheme has reduced system sampling rate. This interaction scheme is graphically depicted in Fig. (7).

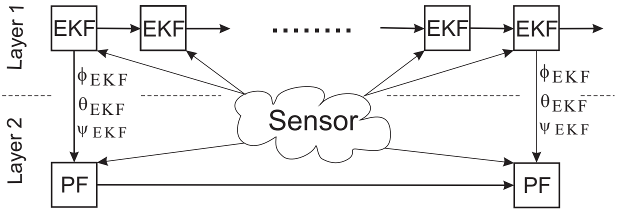

Fig. 7. Between layer interaction according to scheme D

In all interaction schemes both layers have access to sensor measurements and are based on the same head orientation model.

3 RESULTS AND DISCUSSION

As was stated earlier, performance in terms of computational efficiency and accuracy of the four proposed two-layer interaction schemes was tested both on simulated and on measurement data. Two performance parameters were used: execution time (in seconds) and root mean square error (RMSE) defined by

$$RMSE = \frac{1}{n}\sqrt{\sum_{i=1}^{N}(x_i - \hat{x}_i)} \qquad (15)$$

where x_i is referent and $\hat{x}_i$ estimated value while n is total number of samples.

3.1 Simulated data

First, three orientation velocity signals and associated biases were arbitrarily chosen and then based on mathematical equations (1), (2) and (3) head orientation angles were calculated. Accelerometer measurements were calculated using (10) and (11). Values of local magnetic field in all three directions (H_x, H_y and H_z) were used in

$$m_x = H_x c(\theta)c(\psi) + H_y c(\theta)s(\psi) - H_z s(\theta) \qquad (16)$$

$$\begin{aligned} m_y \;=\;& H_x(s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi)) + \\ & +H_y(s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi)) + \\ & +H_z s(\phi)c(\theta) \end{aligned} \qquad (17)$$

$$\begin{aligned} m_z \;=\;& H_x(c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi)) + \\ & +H_y(c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi)) + \\ & +H_z c(\phi)c(\theta) \end{aligned} \qquad (18)$$

to calculate simulated magnetometer readings (m_x, m_y and m_z). Simulated sensor readings were then injected with artificial zero-mean Gaussian distributed noise and were then used as inputs to the two-layer estimation architecture. Sampling frequency was set to 50 Hz. It should be noted that estimation was also achieved with EKF, PF as well as PF with reduced sampling rate. Comparison of obtained results for all estimation algorithms is presented in Table (1).

Examination of data presented in Table (1) reveals the following: a) EKF is the fastest while PF is the most accurate estimation algorithm, b) reduction in PF sampling rate alone by half reduces execution time by half but also reduces accuracy compared to full PF (in case of yaw angle as much as half), c) scheme B has the best combination of performance parameters of all the proposed interaction schemes. Graphical comparison of referent values for roll angle with PF, EKF and scheme B estimation results is presented in Fig. (8) while Fig. (9) depicts $\sqrt{\mathbf{CRLB}}^3$ and standard deviations of 100 Monte Carlo simulations for the same angle.

Table 1. Simulation results

Algorithm	RMSE [^{o}]			Time [s]
	Roll	Pitch	Yaw	
EKF	8.67	8.29	12.04	0.0597
PF	2.07	2.55	4.85	2.9407
PF (reduced)	2.67	6.48	8.6	1.4723
Scheme A	3.32	3.51	8.35	0.6967
Scheme B	3.22	2.91	7.29	1.0083
Scheme C	3.2	5.63	9.06	20.1439
Scheme D	8.68	11.8	14.02	0.6067

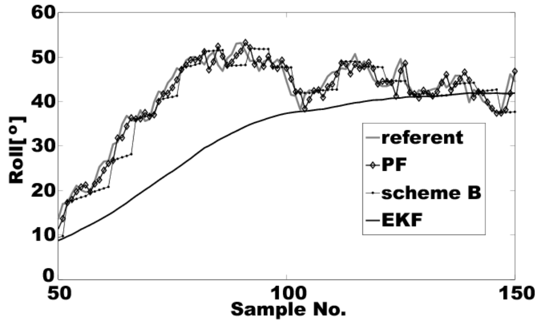

Fig. 8. Comparison of PF, EKF and scheme B roll estimation results with referent values for simulated data

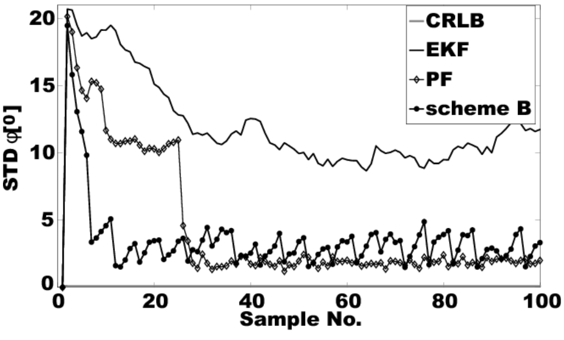

Fig. 9. Comparison of $\sqrt{\mathbf{CRLB}}$ values for head roll angle with EKF, PF and scheme B results

The simulation results showed that the proposed two-layer architecture is feasible and that its results present a com-

[3] $CRLB$ ≡Cramer-Rao lower bound

promise in terms of accuracy and computational complexity between EKF and PF and thus justify further testing.

3.2 Measurement data - batch mode

Measurements for performance testing were achieved with one test subject who was instructed to arbitrarily move his head in all directions and with varying speeds. This procedure was repeated several times. XSens MTx sensor was used in the measurements and was attached to subject's head via elastic strap. The sensor outputs included raw data (accelerometer, gyroscope and magnetometer data) as well as orientation data. Raw data was used as input to the proposed two-layer architecture tested on Pentium 4 with 2 GB memory PC. Recorded data set (of 61 second length) was presented to estimation algorithm(s) in batch mode.

Under these test conditions exact process and measurement noises were unknown i.e. matrices $\mathbf{Q}$ and $\mathbf{R}$ could not be defined analytically. Thus, through process of trial and error (before actual batch data processing and on separate data set) these values were experimentally determined in a way that RMSE was minimized. Estimation sampling frequency was not constant and was dependent on available sensor measurements which in turn depended on available computer resources. Its value was between 9 and 50 Hz with mean value of 32 Hz. Comparison for both performance parameters for all tested algorithms is presented in Table (2).

Table 2. Real measurement batch mode results

Algorithm	RMSE [^{o}]			Time [s]
	Roll	Pitch	Yaw	
EKF	1.49	1.42	2.63	0.3702
PF	1.08	1.38	2.17	21.5238
PF (reduced)	2.85	4.83	4.68	4.4136
Scheme A	1.14	1.24	3.5	4.9215
Scheme B	1.15	1.37	2.21	5.0051
Scheme C	0.92	1.68	2.72	155.659
Scheme D	3.22	8.08	10.41	4.2802

From the presented data the following conclusions can be drawn: a) EKF is the fastest method while PF is again the most accurate, b) the difference between EKF and PF accuracy is not significant as it was for simulated data which we attribute to variable sampling frequency, c) effects of PF down-sampling are the same as for simulated data, d) scheme B again has the best combination of performance parameters of all suggested interaction architectures. Graphical comparison of referent values for roll angle with PF, EKF and scheme B estimation results is depicted in Fig. (10). Obtained results are in accordance with results from simulated data.

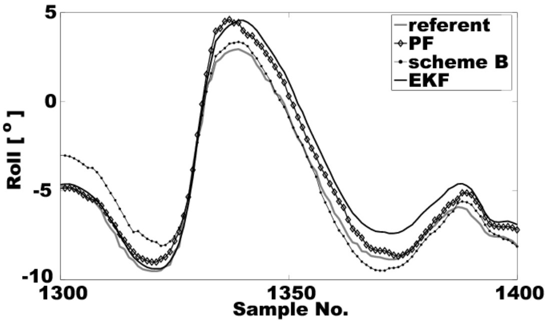

Fig. 10. Comparison of PF, EKF and scheme B roll estimation results with referent values for beatch mode

3.3 Measurement data - real time

Scheme B, as interaction architecture with most promising performance results, was implemented in real time in Visual C# programing environment. Measurement procedure was the same as in Subsection (3.2). No additional estimation algorithms were tested and RMSE value was used as sole performance parameter. Sampling frequency was not constant and depended on available computer resources. Its value was between 10 and 50 Hz, with mean value of 37 Hz. Comparison of referent and estimated head orientation angles is depicted in Fig. (11). For clarity reasons, the figure depicts only 200 measurement samples and not the whole measurement sequence.

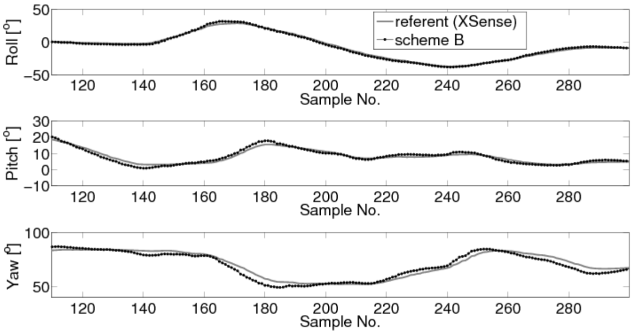

Fig. 11. Comparison of scheme B estimation results with referent values in real time

RMSE values for all three orientation angles were as follows: 1.42^o for roll, 1.46^o for pitch and 3.68^o for yaw angle. Obtained results (both the values and error trends) are in accordance with previous conclusions. During the entire testing which lasted more than 11 hours no real time or implementation (e.g. numerical instability) issues were detected. Thus we concluded that the proposed two-layer estimation architecture based on interaction scheme B is appropriate for real time application and has accuracy suitable for number of practical applications.

4 APPLICATIONS

In order to demonstrate applicability of the proposed two-layer architecture in real world situations three application scenarios were considered: text entry, gesture recognition and robotic arm control. Calibration had to be performed before each of the applications relating head orientation angles to desired control signal(s).

4.1 Text entry

Text entry task was achieved through on-screen keyboard provided by freeware program Click-N-Type[4]. The user controlled Windows pointer with head motion and positioned the pointer over the desired on-screen key and left it there for certain amount of time since key selection was time triggered (in our experiments this parameter was set to 0.7 s). A sentence with 319 characters (including spaces) was red at comfortable pace to 15 test subjects who had to enter it as fast as possible. User performance was measured with words per minute (WPM) with normalized word length of 5 characters including spaces. This meant that in our particular case users had to enter 63.8 normalized words. Average "typing" speed was measured to be 5.01 WPM with standard deviation of 1.33. In the literature average WPM values for similar non-keyboard and non-mouse systems range from 1.9 [23] to 25 [24] depending on system characteristics and test subjects. Similar system, based on head motion tracking for text entry which didn't use text entry prediction and was tested on two impaired subjects [25] produced similar results (5.4 WPM).

4.2 Head gesture recognition

The feasibility of inertial sensor based system with two-layer estimation architecture for head gesture recognition was examined next. Six different head gestures whose on-screen pointer traces are depicted in Fig. (12) were used. Each gesture was repeated five times by each of 15 test subjects. Three repetitions were used for learning purposes while two were used for recognition.

Discrete dynamic movement primitives (DPMs) [26, 27] were used for gesture modeling and recognition. This approach then enabled calculation of base function $f(x)$ defined by

$$f(x) = \frac{\sum_{i=1}^{N} \psi_i}{\sum_{i=1}^{N} \psi_i w_i x_i} \tag{19}$$

where N is number of base functions (in the experiments we used $N = 25$), ψ_i are equally spaced Gaussian functions, w_i are Gaussian base function weights and x_i is phase variable which is used as substitute for time. The equation (19) is solved by linear regression for variable w_i which can be shown to have characteristic set of values for particular discrete movement. Since in the paper $2D$ motion of on-screen pointer is considered, two sets of Gaussian function weights (one for x and one for y axis) need to be calculated. Once all of the function weights were calculated based on first three data sets, two remaining measurement were used for recognition by means of simple correlation coefficient defined as

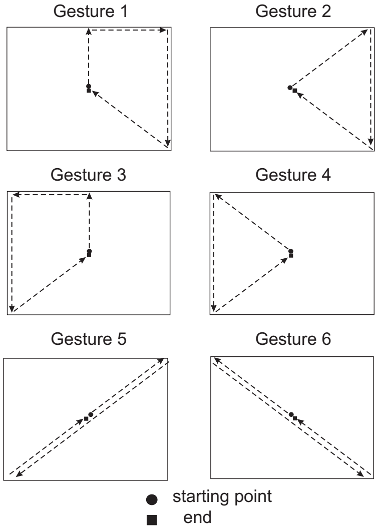

Fig. 12. On-screen pointer traces for used head gestures

$$c_i = \frac{w_L^{iT} w_T}{\mid w_{iL} \mid \mid w_{iT} \mid} \tag{20}$$

where w_{iL} are learned weights while w_{iT} are test weights. Obtained results for each gesture are shown in Table (3).

Table 3. Gesture recognition rate for all test subjects

Gesture No.	Average recognition rate [%]
1	77
2	83
3	80
4	80
5	57
6	87

Average gesture recognition rate was 77% with $\pm 10\%$ standard deviation, compared to 84% recognition rate found in literature [26]. But when making this kind of comparison it should be noted that in [26] advanced form of linear regression (Locally Weighted Projection Regression - LWPR) was used along with k-nearest neighbor algorithm. Thus, we concluded that obtained recognition results are comparable to ones found in literature further

[4] Lake Software, www.lakefolks.org/cat

establishing applicability of two-layer estimation architecture for head orientation tracking.

4.3 Robot manipulator control

Final application scenario on which we tested new estimation architecture was control of small robot manipulator with 5 DOF (all revolute joints) + grasper open-close. Several control schemes were tested with 2D inverse kinematic approach producing best results in terms of execution time of specified task. Task at hand was relocation of plastic glass from starting to final position and return to the robot initial position as is shown in Fig. (13).

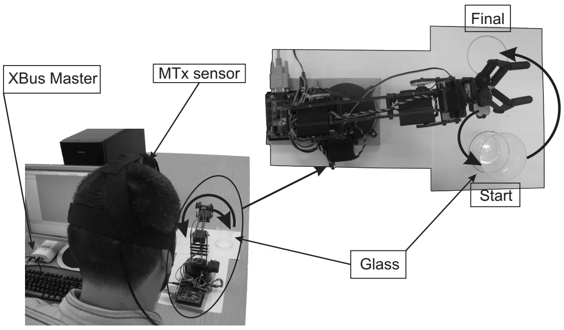

Fig. 13. Experimental setup for head motion based robot control

2D inverse kinematic model enabled users to independently set x, y and z coordinates of robot end-effector to the desired position while the model calculated commands to individual joint actuators. Three test subjects participated in the experiment, each one completing the task at hand three times without significant difficulties. Test subjects had some problems achieving desired task when no inverse model was used i.e. users had control over every individual robot actuator. The difference can clearly be seen from task completion times presented in Table (4). This can be attributed to the fact that users did not know the dynamics of robot and coupling between joint angles, resulting in numerous corrections.

Table 4. Comparison of task execution times

Subject No.	Time without the model [s]	Time with model [s]
1	390	186
2	361	180
3	374	201
Average	375 (±14.5)	189 (±10.8)

Although no similar data could be found in literature, we believe that successful (multiple time) completion of task demonstrated feasibility of head sensor based approach to robot control as well as applicability of two-layer estimation architecture.

5 CONCLUSION

The two layer estimation architecture based on extended Kalman (EKF) and particle filter (PF) for the purpose of 3D head orientation tracking using inertial and magnetic sensor measurements is presented. The architecture is validated on three data sets: simulated data, real measurement data in batch mode and in real time. Measurements were achieved with XSens MTx inertial sensor pack. The work is a first step towards the development of generic architecture that would enable simple upgrade of already implemented estimation systems achieving results that would in terms of accuracy and computational complexity be a compromise between PF and EKF. Four techniques for between layer interaction were designed and tested.

The proposed interaction techniques for two-layer architecture were first validated by simulation. Using known mathematical equations for Euler angle time propagation as well as sensor models, noisy sensor measurements were generated. This noisy data represented input to two-layer estimation architecture. Obtained results demonstrated feasibility of the approach with best interaction scheme having root mean square error (RMSE) of 3.22^o for roll, 2.91^o for pitch and 7.29^o for yaw angle. The tracking performance on the measurement data (in batch mode) is encouraging and has RMSE values smaller then simulated data which we attribute to variable system sampling rate present in real measurements due to limited computer resources. However trends in signal errors (i.e. yaw having the largest error) are the same. RMSE values for best interaction scheme in batch mode are: 1.15^o for roll, 1.37^o for pitch and 2.21^o for yaw angle. Since the same interaction scheme again produced the best combination of results, it was selected for real time implementation. Real time implementation was achieved in Visual C# environment with similar RMSE values (1.42^o for roll, 1.46^o for pitch and 3.68^o for yaw angle) and in $11+$ hour test period showed numerical stability.

The proposed estimation architecture for head tracking applications in human-computer interaction was tested on three practical scenarios: text entry, gesture recognition and control of robot manipulator. Applicability was demonstrated in all three situations with results comparable to those found in literature.

The presented results show that two-layer estimation architecture for head tracking and human-computer interaction is a reliable alternative to standard EKF and PF algorithms. Possible improvements include implementation of adaptive EKF algorithms [7], adaptive number of PF particles [28], development of new layer interaction schemes and implementation of central overseer which would (based on available resources and desired accuracy) switch between standard EKF and PF algorithms and two-layer architecture.

REFERENCES

[1] D. Roetenberg, L. Henk, C. T. Baten, and P. H. Veltink, “Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 3, pp. 395–405, 2005.

[2] J. Musić, R. Kamnik, and M. Munih, “Model Based Inertial Sensing of Human Body Motion Kinematics in Sit-to-Stand Movement,” *Simulation Modelling Practice and Theory*, vol. 16, no. 8, pp. 933–944, 2008.

[3] R. Zhu and Z. Zhou, “A Real-Time Articulated Human Motion Tracking Using Tri-Axis Inertial/Magnetic Sensor Package,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, no. 2, pp. 295–302, 2004.

[4] X. Yun and E. R. Bachmann, “Design, Implementation, and Experimental results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking,” *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1216–1227, 2006.

[5] D. Roetenberg, P. Slycke, and P. H. Veltink, “Ambulatory Position and Orientation Tracking Fusing Magnetic and Inertial Sensing,” *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 5, pp. 883–890, 2007.

[6] E. R. Bachmann, X. Yun, and A. Brumfield, “Limitations of Attitude Estimation Algorithms for Inertial/Magnetic Sensor Modules,” *IEEE Robotics and Automation Magazine*, vol. 14, no. 3, pp. 76–87, 2007.

[7] D. Jurman, M. Jankovec, R. Kamnik, and M. Topič, “Calibration and Data Fusion Solution for the Miniature Attitude and Heading Reference System,” *Sensors and Actuators A*, vol. 138, no. 2, pp. 411–420, 2007.

[8] E. Foxlin, Harrington, and Y. Altshuler, “Miniature 6-DOF inertial system for tracking HMDs,” in *Proceedings of AIAA 3td Unmanned Unlimited Technical Conference, Workshop and Exhibit*, pp. 214–228, 1998.

[9] X. Yun, E. R. Bachmann, and R. B. McGhee, “A Simplified Quaternion-Based Algorithm for Orientation Estimation From Earth Gravity and Magnetic Field Measurements,” *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 3, pp. 638–650, 2008.

[10] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. New Jersy, USA: Wiley-Interscience, 1st ed., 2006.

[11] J. Crassidis, F. L. Markley, and Y. Cheng, “A Survey of Nonlinear Attitude Estimation Methods,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[12] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science, New York, USA: Springer Verlag, 1st ed., 2001.

[13] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[14] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, USA: Artech House, 1st ed., 2004.

[15] J. Rawlings and B. Bakshi, “Particle Filtering and Moving Horizon Estimation,” *Computers and Chemical Engineering*, vol. 30, no. 10-12, pp. 1529–1541, 2006.

[16] J. J. LaViola, “A Comparison of Unscented and Extended Kalman Filter for Estimating Quaternion Motion,” in *Proceedings of 2003 American Control Conference*, pp. 2435–2440, 2003.

[17] B. Krach and P. Robertson, “Cascade Estimation Architecture for Integration of Foot-Mounted Inertial Sensors,” in *Proceedings of 2008 IEEE/ION Position Location and Navigation Symposium*, 2008.

[18] S. Haykin, *Kalman Filtering and Neural Networks*. New York, USA: John Wiley, 1st ed., 2001.

[19] M. Ouerfelli and W. S. H. Vijay Kumar, “Kinematic Modeling of Head-Neck Movements,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A*, vol. 29, no. 6, pp. 604–615, 1999.

[20] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology: Progress in Astronautics and Aeronautics*. Radar, Sonar and Navigation Series 17, Herts, UK: The Institution of Engineering and Technology, 2nd ed., 2005.

[21] “Human Characteristic Database,” tech. rep., National Institute of Technology and Evaluation, Japan, 2003.

[22] M. Bolic, P. Djuric, and S. Hong, “Resampling Algorithms for Particle Filters: A Computational Complexity Perspective,” *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 15, pp. 2267–2277, 2004.

[23] B. Blankertz, M. Krauledat, G. Dornhege, J. Williamson, R. Murray-Smith, and K.-R. Muller, “A Note on Brain Actuated Spelling with the Berlin-Brain-computer Interface,” in *Universal Science in Human-Computer Interaction. Ambient Interaction*, vol. LNCS 2555, pp. 759–768, 2007.

[24] D. J. Ward and D. J. MacKay, “Fast Hands-free Writing by Gaze Direction,” *Nature*, vol. 418, no. 6900, p. 838, 2002.

[25] R. DeVries, J. Deitz, and D. Anson, “A Comparison of Two Computer Access Systems for Functional Text Entry,” *The American Journal of Occupational Therapy*, vol. 52, no. 8, pp. 656–665, 1998.

[26] A. J. Ijspeert, J. Nakanishi, and S. Schaal, “Learning Attractor Landscapes for Learning Motor Primitives,” in *Advances in Neural Information Processing Systems 15*, vol. 2, pp. 1547–1554, 2002.

[27] S. Schaal, J. Peters, J. Nakaninshi, and A. J. Ijspeert, “Learning Movement Primitives,” in *International Symposium on Robotic Research*, pp. 561–572, 2003.

[28] A. Carmi and Y. Oshman, “Adaptive Particle Filtering for Spacecraft Attitude Estimation from Vector Observations,” *Journal of Guidance, Control and Dynamics*, vol. 32, no. 1, pp. 232–241, 2009.

Josip Musić received B.S. degree in 2004. from Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, M. Sc. degree in 2007. from Faculty of Electrical Engineering, University of Ljubljana, Slovenia and Ph. D. degree in 2010. from Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split (all in electrical engineering). Since 2004 he is employed as a researcher and teaching assistant at Laboratory for biomechanics, automatic control and systems, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split. He has been visiting researcher at University of Ljubljana and University of Glasgow. He is a member of IEEE. The main area of his scientific research includes rehabilitation robotics and assistive technology, human-machine interaction, inertial sensing and related estimation algorithms.

Mojmil Cecić was born in 1960. in Grohote, Croatia. He received his B.S. degree in 1984., M.Sc. degree in 1993. and Ph.D. degree in 1999. all in electrical engineering from Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split. Since 1985. he is employed at Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, where he currently holds position of full professor. He participated in successful completion of 8 scientific projects. He is a member of CROMBES, KOREMA, CROSSIM and HDR societies. He's research interests include rehabilitation engineering, human motion modelling and gait biomechanics.

Vlasta Zanchi is currently professor emeritus in the Faculty of Electrical Engineering, Mechanical Engineering, and Naval Architecture at the University of Split in Croatia, where she is the head of the Laboratory for Biomechanics, Automatic Control Systems (LaBACS). She received her Dipl.Ing. from the Faculty of Electrical Engineering at the University of Zagreb in Croatia in 1963, followed by her M.Sc. and Ph.D. in Electrical Engineering from the University of Ljubljana in Slovenia in 1974 and 1977, respectively. She became professor emeritus in December 2008. She is currently involved in undergraduate and postgraduate teaching in the areas of Systems Theory, Non-linear Systems, Optimization, Biomechanics, Identification, and Estimation and Simulation Theory.

AUTHORS' ADDRESSES

Josip Musić, Ph. D.
Prof. Mojmil Cecić, Ph. D.
Prof. Vlasta Zanchi, Ph. D.
Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
University of Split
R. Boškovića 32, HR-21000 Split, Croatia
email: jmusic@fesb.hr, mcecic@fesb.hr, vzanchi@fesb.hr

Received: 2010-07-12
Accepted: 2010-10-26