ARITHMETIC CONSTRAINTS IN KALMAN FILTER IMPLEMENTATION BY USING IMS A100 DEVICES

I. D'Antone +, L. Fortuna *, G. Muscato *, G. Nunnari *$

Introduction

In this paper the realization of Kalman filter algorithms by using a parallel architecture based on Transputers and IMS A100 DSP is discussed. The implementations take full advantage from the particular IMS A100 internal structure to speed up the computations. In fact each IMS A100 contains 32 high speed, high accuracy 16x16 bit multiply accumulators and it can be cascadable to other similar devices to produce powerful transversal filters. Such type of devices have been successful applied to implement digital filtering, correlation and convolution, discrete Fourier transform algorithms etc.

Two implementations are studied in this paper. The first one concerns with a classical optimal observer scheme, essentially consisting of a MIMO IIR filter, where filter gain updating is not required, while the second one refers with the implementation of a Kalman filter in general form.

Due to the fact that Kalman filter implementations are very sensitive both regarding the propagating round-off and quantization errors[1,2] particular solutions are required to minimize their effects and avoid the probability of numerical overflow. For this purposes the use of a particular realization of the original dynamical system, referring to the balanced realization, is considered[3]. The numerical drawbacks due to the fixed point arithmetic used by the IMS A100 devices, mainly for the Kalman filter in general form are pointed out by appropriate examples.

The saving in time in using the proposed architecture are shown in comparison with analogous sequential schemes.

Hardware and Software Support

The hardware environment used in the present application is represented by an IBM PC AT with standard configuration equipped with a INMOS B009 card containing a T414 32 bit transputer, and a T212 16 bit transputer controlling a chain of four IMS A100. Each transputer contains, on the same chip, a processor, a bank of on-chip DRAM and four links to communicate with other transputers and/or I/O devices.

+ Istituto Nazionale di Fisica Nucleare - Sez. di Bologna - Italy
* Istituto di Elettrotecnica ed Elettronica
Universita' di Catania, Viale A. Doria, 6 - Italy

The B009 card communicates with the PC via a link adapter and can be programmed by using the IMS Development System which represents a complete environment for the development, execution and debugging of IMS A100 based applications.

The OCCAM 2 high level concurrent programming language, is suitable to support applications consisting of individual parts operating separately, interacting through appropriate communication channels.

The Kalman Filter Algorithm

Let be consider a discrete dynamical system in state space representation

$$x(t) = A(t-1)\ x(t-1) + B(t-1)\ u(t-1) + W(t-1)\ w(t-1) \qquad (1)$$
$$y(t) = C(t)\ x(t) + v(t)$$

The problem to compute the state vector estimation x(t) can be solved in recursive form performing the following steps[4]:

- state prediction: $\hat{x}(t/t-1) = A(t-1)\hat{x}(t-1/t-1) + B(t-1)u(t-1);$ (2)

- covariance prediction:

$$\tilde{P}(t/t-1) = A(t-1)\ \tilde{P}(t-1/t-1)\ A'(t-1) + W(t-1)\ Rw(t-1)\ W'(t-1); \qquad (3)$$

- innovation: $e(t) = y(t) - C(t)\ \hat{x}(t/t-1);$ (4)

- innovation covariance: $Re(t) = C(t)\ \tilde{P}(t/t-1)\ C'(t) + Rv(t);$ (5)

- Kalman gain: $K(t) = \tilde{P}(t/t-1)\ C'(t)\ Re^{-1}(t);$ (6)

- state correction: $\hat{x}(t/t) = \hat{x}(t/t-1) + K(t)\ e(t);$ (7)

- covariance correction: $\tilde{P}(t/t) = \left[I - K(t)\ C(t)\right]\ P(t/t-1);$ (8)

Fixed Gain Kalman Filter Implementation

Let be consider a discrete time linear system with time invariant matrices A, B, C, W, Rw and Rv. In this case it is easy to prove that the state estimation can be performed as

$$x(t) = \left[A-KC \mid B \mid K\right] \begin{bmatrix} x(t-1) \\ u(t-1) \\ y(t-1) \end{bmatrix} \qquad (9)$$

The matrix by vector multiplication required to compute x(t) has been conveniently implemented, by using a chain of IMS A100 devices controlled by a T212 transputer, which allow to obtain a considerable speed-up of the algorithm, as shown in our previous paper[5]. The execution time versus the model order in performing the

Kalman filter algorithm are reported in Fig. 1 referring to three different processors:

- the T212 controlling the IMS A100 chain (referred as T212 + IMS A100 below);
- a 32 bit T414 transputer;
- a 32 bit T800 transputer, which is one of the most powerfull microprocessor available today on the market.

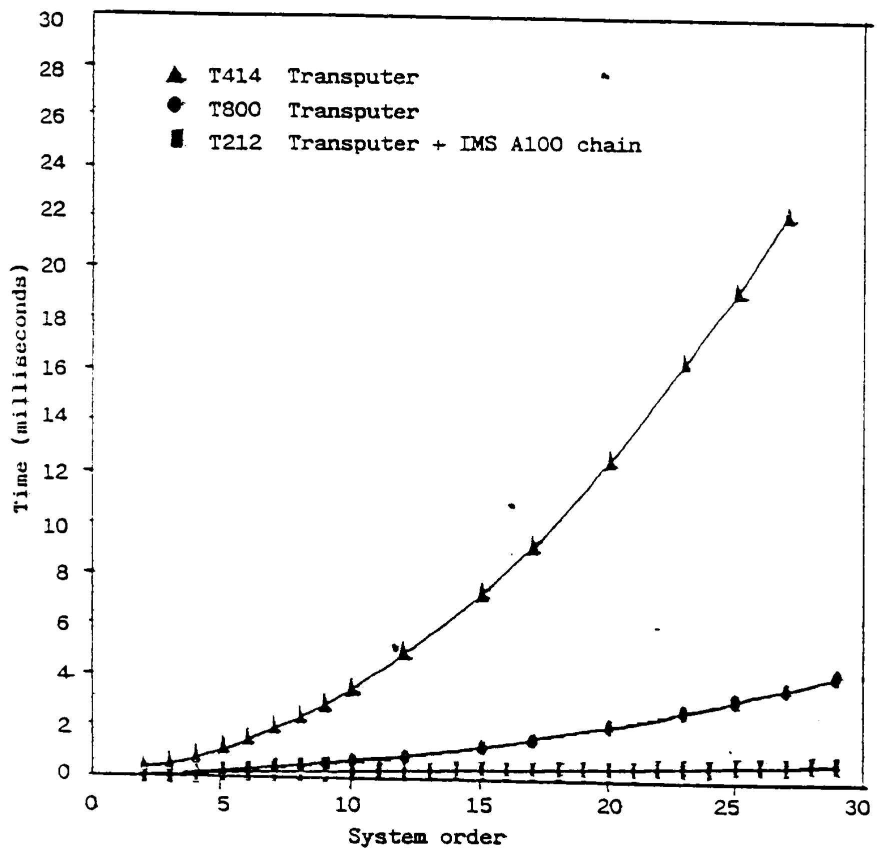

Fig. 1 - Kalman filter execution time versus system order

The advantages in using the IMS A100 based processor appear evident.

However some drawbacks can arise in implementing the Kalman filter based on the matrix multiplication algorithm mentioned above, due to the IMS A100 16 bit fixed point arithmetic.

In order to illustrated these implementation problems, an example is reported below. Let be consider the discrete time model in the state space representation:

$$A = \begin{bmatrix} -0.7668 & 0.1203 \\ -0.9637 & 0.6020 \end{bmatrix};$$

$$B' = W' = \begin{bmatrix} 0.5529 & 2.7613 \end{bmatrix};$$

$$C = \begin{bmatrix} 0.8589 & 0.1840 \end{bmatrix}$$

Let the model input $x(t)$ be a signal ranging between 0 and 10^4. Suppose also that $x(t)$ be affected by a noise with variance $R(t)=10^4$. The x1 state variable computed from the model and the estimated values are shown in Fig. 2a and 2b respectively. It is possible to observe that the estimated values are not correct due to an occurred overflow.

Let be consider now the previous model in balanced representation form:

$$A = \begin{bmatrix} 0.4681 & -0.2224 \\ -0.2224 & -0.6329 \end{bmatrix}$$

$$B' = C = W' = \begin{bmatrix} 0.8808 & 0.1022 \end{bmatrix}$$

applying the same input considered in the previous example the obtained computed and estimated values for the x1 state variable are shown in Fig. 3a and 3b. where it is possible to observe a good estimation of the true value.

The different numerical behaviour in the two considered cases can be explained considering that in the former example the numerical values of the model state variables are wide-spread in a range greater than the latter example, where the considered model was represented in balanced form. Such observation has been proved from the theoretical point of view by Mullis and Robert[3], which shown as the use of balanced realization minimizes the effect of the round-off error for a pre-fixed probability of overflow.

In conclusion in the described Kalman filter implementation some arithmetic constraints can arise but they can be generally avoided by representing the model in balanced realization form.

Time Varying Gain Kalman Filter Implementation

The implementation of a Kalman filter with variable gain has been also studied. The gain vector K is computed at each iteration by using the expression reported above.

However in this case the previous T212 + IMS A100 based architecture, cannot be successful used due to the 16 bit IMS A100 fixed point arithmetic, even if models in balanced representation form are used. In fact the $K(t)$ computation involves the matrix multiplication described in the expressions (4),(5) and (6). If the T212+IMS A100 structure is used to compute $K(t)$, large errors are obtained after a few computation steps, due to the round-off errors generated in the 16 bit truncation.

However such problem can be solved performing the computation involved in $K(t)$ updating by using a floating point algorithm running on the T414 transputer contained on the B009 card, while the previous T212 + IMS A100 structure executes the state estimation.

The balanced representation form of the models improves the numerical performances also in this case.

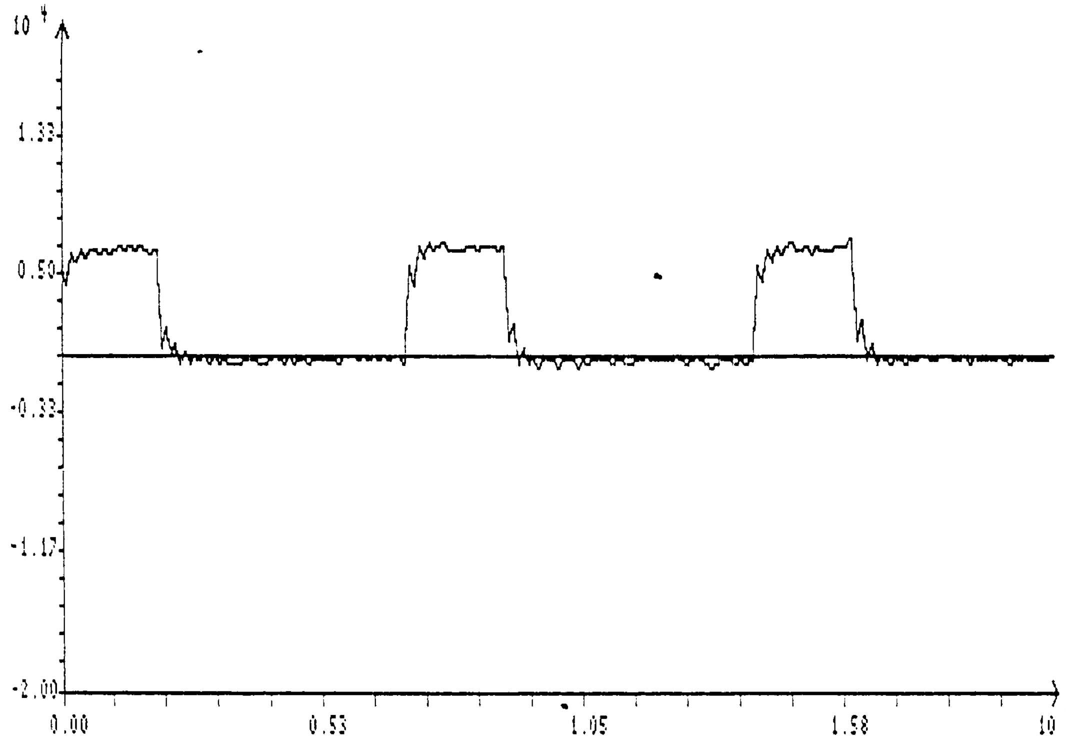

Fig. 2a x_1 state variable trend

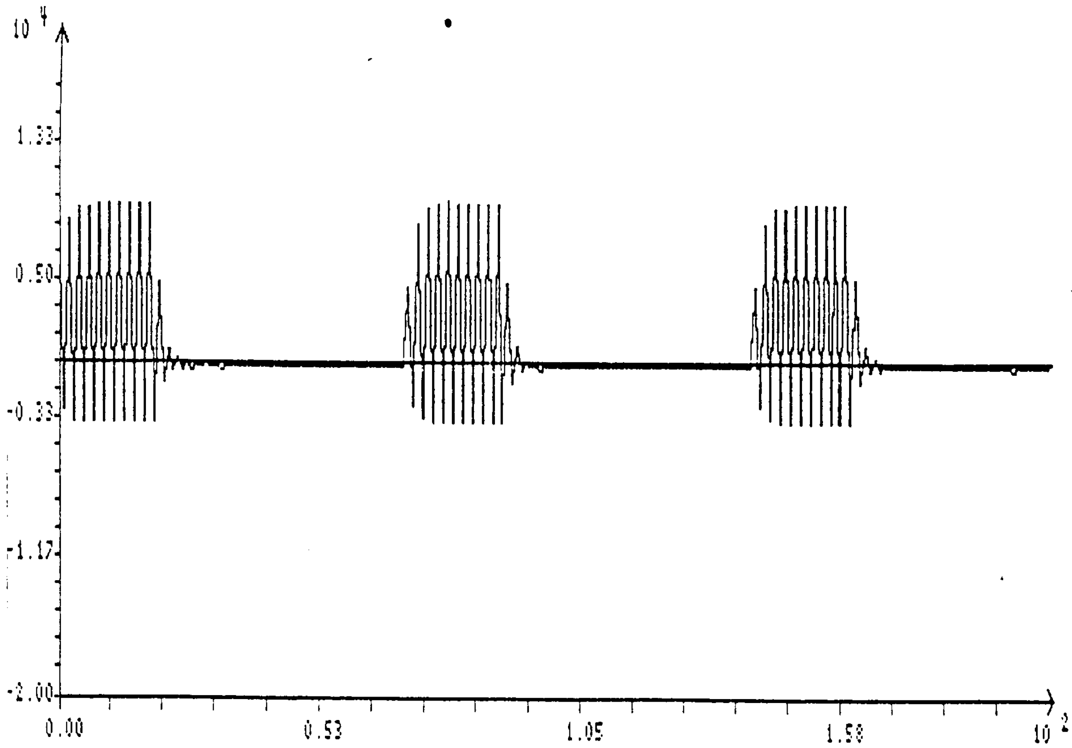

Fig. 2b x_1 state variable trend estimated by using transversal filter architecture system.

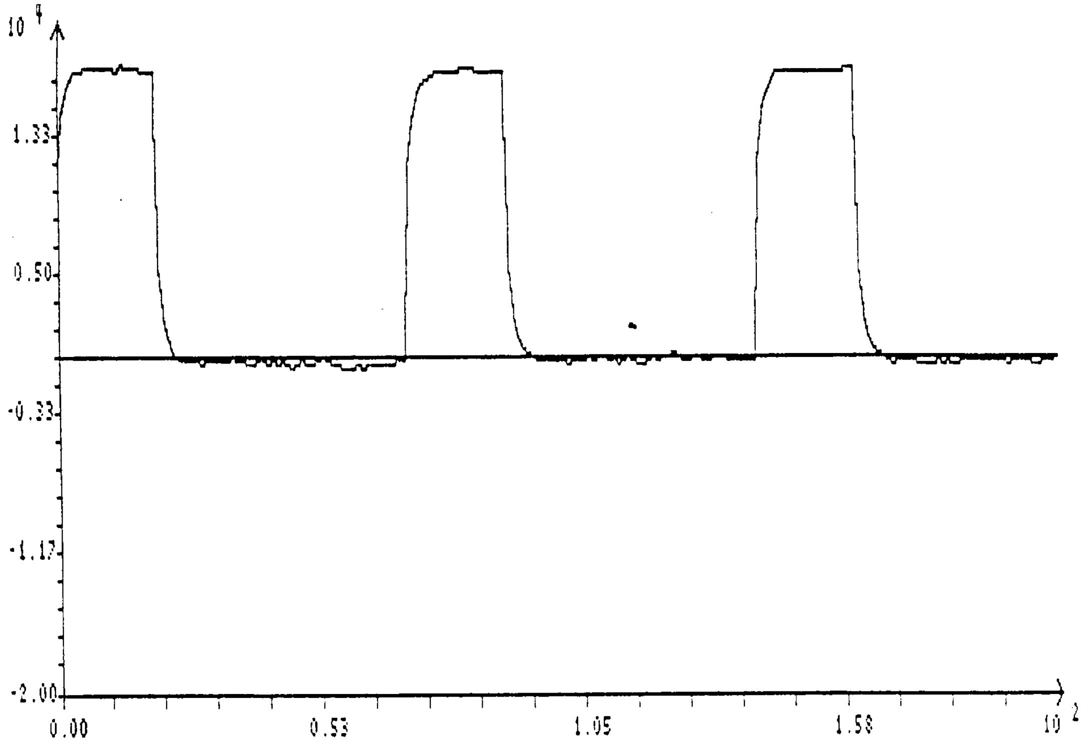

Fig. 3a x_1 state variable trend in the open-loop balanced representation scheme

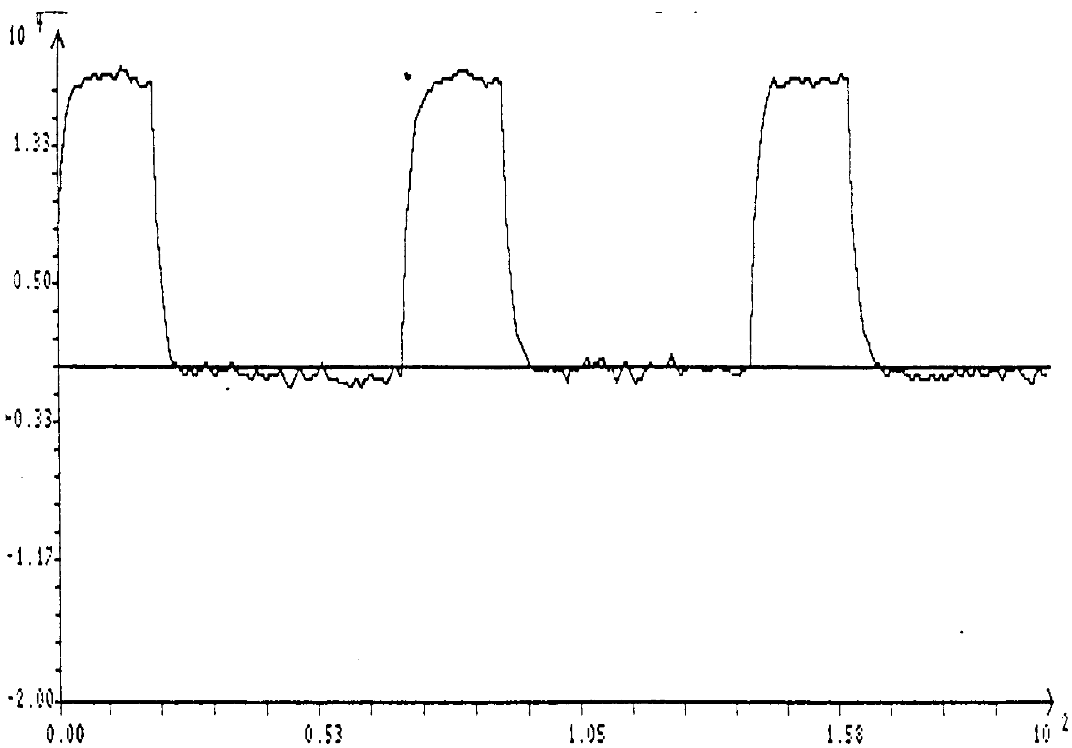

Fig. 3b x_1 state variable trend in the open-loop balanced representation scheme estimated by using transversal filter architecture system

Conclusions

In this paper some arithmetic constraints regarding the digital Kalman filter implementation by using digital signal processors with transversal filter architecture have been outlined.

Referring the fixed gain Kalman filter suitable results can be obtained both as regards the computation speed and the precision by using particular state space representation. The proposed approach can represent a first step in implementing high order linear state based regulator for fast dynamical system controlling.

Studying the time varying gain Kalman filter realization by using the T212 + IMS A100 hardware architecture, the 16 bit word lenght and the fixed point arithmetic due to IMS A100 devices lead to some drawbacks. However such difficulties overcomed executing the K(t) updating by using a concurrent algorithm running on the T414 transputer.

References

1. M. Verhaegen, P. Van Dooren, Numerical Aspects of Different Kalman Filter Implementation, IEEE Trans. on AC, vol AC-31, 1986.

2. D. Williamson, Finite Wordlenght Design of Kalman Filters for State Estimation, IEEE Trans. on AC, Vol. AC-30, N. 10, 1985.

3. C.T. Mullis, R.A. Roberts, Synthesis of minimum round noise in fixed point digital filters, IEEE trans. on Circuit and Systems, Vol CAS-23, 1986.

4. J. V. Candy, Signal Processing the model based approach, MacGraw-Hill, 1986.

5. I. D'Antone, L. Fortuna, A. Gallo, G. Nunnari, IMS A100 digital signal processor in the simulation of discrete time control systems, Proc. 1988 Summer Computer Simulation Conference, Seattle, 1988.