

## DISCRETE RICCATI EQUATION SOLUTIONS: DISTRIBUTED ALGORITHMS

D.G. LAINIOTIS<sup>a</sup>, K.N. PLATANIOTIS<sup>b,\*</sup>, MICHAEL PAPANIKOLAOU<sup>c,†</sup>, and  
PARASKEUAS PAPANIKOLAOU<sup>d,‡</sup>

<sup>a</sup>Intelligent Systems Technology, 3207 S. Hwy. A1A, Melbourne Beach, FL-32951, USA;

<sup>b</sup>Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Rd, Toronto, Ontario M5S 1A4, Canada; <sup>c</sup>Department of Computer Engineering and Informatics, University of Patras, Patras, GR-26500, Greece; <sup>d</sup>Department of Electrical and Computer Engineering, Florida Institute of Technology, Melbourne, FL-32901, USA

(Received 2 August 1995)

In this paper new distributed algorithms for the solution of the discrete Riccati equation are introduced. The algorithms are used to provide robust and computationally efficient solutions to the discrete Riccati equation. The proposed distributed algorithms are theoretically interesting and computationally attractive.

**KEYWORDS** Riccati equation, discrete estimation filters, distributed algorithms, Kalman filter, linear Lainiotis filter.

### I. INTRODUCTION

The Riccati equation (*RE*) plays an important role in many fields of mathematics, science and engineering [13], [6], [7]. The solution of the Riccati equation is essential in such fields as optimal estimation and control, game theory, integral equations, and power systems. In view of the importance of the Riccati equation, there exists considerable literature both on the algorithmic as well as on the numerical aspects of its solution [13], [5], [6], [8]. The basic Riccati equation of interest in this paper arises in linear estimation, namely in the implementation of state space estimation algorithms such as, the widely studied Kalman filter and the Lainiotis linear partitioning filters [13], [7], [11]. The applicability of the various Riccati equation solutions to practical problems was until recently limited by the computing power available. Real problems in application areas, such as estimation and stochastic control require fast and accurate computation of large volumes of data as a consequence of the necessity to deal with complex more realistic models. The recent advances in the computer technology allows the use of distributed data

---

\*Author for correspondence. Digital Signal/Image Processing Laboratory, Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, Ontario, Canada, M5S 1A4, Tel: (416) 978-6845, Fax: (416) 978-4425, E-mail: kostas@dsp.toronto.edu

<sup>†</sup>E-mail: papanikm@prometheus.hol.gr

<sup>‡</sup>E-mail: parask@ee.fit.edu

fusion systems, consisting of networks of sensor nodes, each one with its own processing and communication capabilities. This motivated research toward more computationally efficient algorithms which can take full advantage of the new technological capabilities [3], [10], [12].

We consider here new distributed, computationally efficient algorithms for the solution of the discrete Riccati equation arising in estimation and stochastic control. A linear, dynamic, multisensor state space model is considered here [2], [1], [10]. Such systems arising naturally when data are obtained by different subsystems located at dispersed locations or when large data volumes must be processed in real-time. In realistic application scenarios the computational requirements of a given estimation algorithm are more sensitive to the variations of the measurement vector dimension than to the variations of the state vector dimension. The increase in the number of sensors would require either the use of faster and more expensive processing units, or more efficient algorithmic approaches [3].

There are two major approaches to obtain the Riccati equation solution required for state estimation in such multisensor systems:

- **The centralized approach**, where all the sensors are considered part of the same measurement model. The solution of the RE in this case involves calculations with matrices of high dimensionality resulting in heavy computational load at the processing facility.
- **The distributed approach**, where the information gathered by the sensors are processed locally. After this first local processing step, interim results are transmitted to a central processor. At this central processor the solution of the RE can be obtained using only the matrices of the state equation.

Two new solutions of the *RE* based on optimal distributed estimation algorithms are discussed here. The paper is organized as follows: In section II, the mathematical description of the problem is presented. In section III, the new algorithms for the distributed solution are introduced and analyzed. Motivation and implementation details are discussed in this section. In section IV, the algorithms are compared in terms of their computational complexity. Finally, section V summarizes our conclusions.

## II. PROBLEM FORMULATION

The linear, multisensor, discrete state space model from where the RE emanates has the following form:

$$x(k + 1) = \Phi(k + 1, k)x(k) + w(k) \quad (1)$$

$$z(k + 1) = H(k + 1)x(k + 1) + v(k + 1) \quad (2)$$

for  $k = 1, 2, \dots$ , where  $x(k)$  is the  $n$ -dimensional state vector,  $z(k)$  is the  $m$ -dimensional measurement vector,  $\Phi(k + 1, k)$  is the  $n \times n$  state transition matrix,  $H(k)$  is the  $m \times n$

output matrix,  $[w(k)]$  and  $[v(k)]$  are  $n$ -dimensional, and  $m$ -dimensional Gaussian zero mean white noises with noise covariances  $Q(k)$  and  $R(k)$  respectively. The initial value  $x(0)$  of the state vector  $x(k)$  at time  $t_0 = 0$  is considered Gaussian random variable with mean  $\hat{x}(0|0)$  and variance  $P(0|0)$ . It is assumed independent of both the plant noise  $w(k)$  and measurement noise  $v(k)$ .

Under the distributed scenario, first introduced in [14], [2], the observation vector  $[z(k)]$  can be partitioned into  $r$  subvectors, each corresponding to a local sensor system with local dimensionality  $m_i$ .

$$z^T(k) = [z_1^T(k), z_2^T(k), \dots, z_r^T(k)] \quad (3)$$

$$H^T(k) = [H_1^T(k), H_2^T(k), \dots, H_r^T(k)] \quad (4)$$

$$v^T(k) = [v_1^T(k), v_2^T(k), \dots, v_r^T(k)] \quad (5)$$

Consequently, the local measurement equation based on the state vector is given by:

$$z_i(k+1) = H_i(k+1)x(k+1) + v_i(k+1) \quad (6)$$

where,  $z_i(k+1)$  is the  $m_i$ -dimensional measurement vector at the local processor system  $i$ ,  $H_i(k+1)$  is the  $m_i \times n$  local output matrix, and  $v_i(k+1)$  is the  $m_i$  local measurement noise vector. It is assumed here that:

$$m = \sum_{i=1}^r m_i \quad (7)$$

In the distributed estimation problem, it is required to obtain the minimum mean square estimate (mmse) of the system state based on the measurement records available at the associated local processors. The task involves the solution of the Riccati equation for the prediction error covariance matrix.

### III. DISTRIBUTED RICCATI EQUATION SOLUTIONS

The discrete Riccati equation arising in the estimation of the system state  $x(k)$  can be solved using the Kalman filter equations [13]. Using the partitioning approach to estimation and control, Lainiotis has also introduced fast, robust and stable computational algorithms for the solution of the Riccati equation [4], [5], [6], [11].

The main drawback, however, of the centralized approaches to the *RE* solution is that they require large communication bandwidth for raw measurement transmission between the local subsystems and the processor where the estimation algorithm is implemented. Moreover, in the case of large measurement dimensionality ( $m$ ), which is exactly what happens in a typical multisensor scenario, there is a tremendous computational burden associated with the processing of all measurement information in a

single processing facility. To overcome such problems, and to achieve robustness and redundancy, distributed approaches to the solution of the Riccati equation are introduced here. Following the distributed formulation analyzed in the previous section, data obtained from different sensors subsystems are processed locally and the results are transmitted to the central processor (data fusion center). The state space model is given in the form of equations (1), (6). The new *RE* solutions are based on the distributed Kalman and the distributed linear Lainiotis filters.

### A. Kalman Filter-based Distributed Riccati Solution.

The Riccati equation solution based on the distributed Kalman filter is summarized in the following theorem:

#### THEOREM III-A.1: *Kalman Filter-based Distributed Riccati Solutions*

Consider the system (1), (6); For this linear discrete time system, the solution of the Riccati equation is given as follows:

- central processor calculations:

$$P^{-1}(kk) = P^{-1}(kk-1) + \sum_{i=1}^r [P_i^{-1}(kk) - P_i^{-1}(kk-1)] \quad (8)$$

$$P(kk-1) = \Phi(k, k-1)P(k-1k-1)\Phi^T(k, k-1) + Q(k-1) \quad (9)$$

with  $k = 1, 2, \dots$  and initial condition  $P(0|0)$

- local processor calculations (for the  $i^{th}$  local subsystem):

$$P_i(k+1|k+1) = [I - K_i(k+1)H_i(k+1)]P_i(k+1|k) \quad (10)$$

$$P_i(k+1|k) = \Phi(k+1, k)P_i(k|k-1)\Phi^T(k+1, k) + Q(k+1) - \Phi(k+1, k)K_i(k)P_{zoi}^{-1}(kk-1)K_i^T(k)\Phi^T(k+1, k) \quad (11)$$

for  $k = 1, 2, \dots$  with initial condition  $P_i(0| - 1) = P_i(0|0)$  and

$$K_i(k) = P_i(k|k-1)H_i^T(k)P_{zoi}^{-1}(kk-1) \quad (12)$$

$$P_{zoi}(kk-1) = H_i(k)P_i(k|k-1)H_i^T(k) + R_i(k) \quad (13)$$

The solution is based on the equations of the distributed Kalman filter [2].

### B. Lainiotis Filter-based Distributed Riccati Solution.

The distributed Riccati equation solution based on the distributed Lainiotis filter is summarized in the following theorem:

**THEOREM III-B.1: Lainiotis Filter-based Distributed Riccati Solution**

Consider the system (1), (6). For this linear discrete time system, the solution of the Riccati equation is given as follows:

- central processor calculations:

$$P(k+1|k+1) = P_n(k+1|k+1) + \Phi_n(k+1, k) P_r^{-1}(k|k+1) \Phi_n^T(k+1, k) \quad (14)$$

$$P_r(k|k+1) = [O_n(k+1) + P^{-1}(k|k)] \quad (15)$$

$$P_n^{-1}(k+1|k+1) = Q^{-1}(k) + B_i(k+1) \quad (16)$$

$$B_i(k+1) = \sum_{i=1}^r [P_{ni}^{-1}(k+1|k+1) - Q^{-1}(k)] \quad (17)$$

$$F_n(k+1) = Q^{-1}(k) \Phi(k+1, k) \quad (18)$$

$$\Phi_n(k+1, k) = P_n(k+1|k+1) F_n(k+1) \quad (19)$$

with initial condition  $\Phi(0, 0) = I_{n \times n}$

$$O_n(k+1) = F_n(k+1)^T [Q(k) - P_n(k+1|k+1)] F_n(k+1) \quad (20)$$

- local processor calculations (for the  $i^{th}$  local subsystem):

$$P_{ni}(k+1|k+1) = [I - K_{ni}(k+1) H_i(k+1)] Q(k) \quad (21)$$

$$K_{ni}(k+1) = Q(k) H_i^T(k+1) P_{zni}^{-1}(k+1|k) \quad (22)$$

$$P_{zni}(k+1|k) = H_i(k+1) Q(k) H_i^T(k+1) + R_i(k+1) \quad (23)$$

The proof is given in the appendix.

**C. Remarks**

1. It must be noted that the above given solutions correspond to the general time varying, linear, dynamic, discrete time state space model.
2. The Riccati equation solution which is based on the distributed form of the linear Lainiotis filter at the different local subsystems requires only nonrecursive simple multiplications. These memoryless operations can be implemented using a very simple processor or even an optical device.
3. There is no two way communication between the local and central processors. Because bidirectional communication between slower local stations and the central

agent is not desirable in a highly parallel processing environment, both the new distributed solutions have been designed with the lowest possible interprocessor communication [12].

4. The solutions that are based on the linear Lainiotis filter lead to elemental solutions totally decoupled from sample to sample. It was pointed out by Lainiotis [5], [6] that the solution is determined in terms of system matrices  $Q(\cdot)$ ,  $R(\cdot)$ ,  $H(\cdot)$ , and  $\Phi(\cdot)$  which can be calculated independently on a per sample basis. The distributed Riccati solution of theorem 2 retains the property of the completely decoupled nominal solutions of the partitioning approach. Moreover, the distributed solution has a more impressive aspect. It allows the calculation of the Riccati equation to be done in a totally distributed, parallel form. Specifically,
  - (a) The Riccati equation solution can be computed in a distributed form, using decentralized distributed algorithms for covariance calculations. Thus, the calculations in the different local models can be performed in parallel.
  - (b) In addition, the Riccati solution is decomposed into a set of nominal solutions decoupled from one sample to another in a parallel mode. They can be performed in parallel since they are naturally decoupled and no filtered quantity at the present step requires recursive quantities from a previous step. To the best of the authors' knowledge the solution given in this work is the only algorithm capable to provide this form of parallel processing capability.
5. Finally, we observe that the Riccati solutions that are based on the linear Lainiotis filter are more robust from a numerical point of view. As it was mentioned above the matrices  $P(k|k)$ ,  $P(k + 1|k)$  are error covariance matrices thus, must be nonnegative definite [6], [11]. In both centralized and distributed form, the Lainiotis filter based approaches use quadratic terms in the calculations of the solution preserving the nonnegative nature of the final output.

The two forms of the distributed solution to the *RE* introduced in this work have similar performance since they involve the same system matrices. However, these solutions have different computational and communication requirements, especially when they are applied to multisensor problems [10], [12].

#### IV. COMPUTATIONAL REQUIREMENTS OF THE DISCRETE RE SOLUTIONS

Apart from the numerical behavior of any proposed algorithm, its computational complexity is a realistic measure of its practicality and usefulness since it determines the required computing power and processing (execution) time. A general framework to evaluate the computational requirements of recursive algorithms is given in [9], [3], [11]. In this work, the rules defined there are used to evaluate our distributed Riccati solutions.

Due to the fact that the algorithms provide the solution in a recursive manner, the algorithms' total execution time is equal to the product of its per recursion computational complexity times the number of recursions required to obtain a solution. The comparisons introduced in this paper are on a per recursion basis. Two more assumptions are introduced in order to have a meaningful comparison among the different algorithms.

1. First, the overall computational requirements for a distributed algorithm, per recursion, are computed as the sum of the per recursion calculation requirements in one of the local processors plus the computational requirements in the central processor since the calculations in all local models are performed in parallel.
2. Secondly, the fundamental operations involved in the solution of the Riccati equation are matrix and vector operations. A detailed analysis of the computations involved in the evaluation in such operations is provided in [3], [12]. The interested reader can refer to them for more information on the subject. In this context, the total time required to complete an operation (or a sequence of operations) is proportional to the normalized total number of equivalent scalar operations, defined as:

$$Time = kX(4X(MULTS) + (ADDS) + 6X(DIVS) + 25X(SQRTS))$$

where *MULTS* is the number of scalar multiplications required, *ADDS* is the number of scalar additions required, *DIVS* is the number of scalar divisions required and *SQRTS* is the number of the scalar square roots. The weights used in the above formula do not refer to any particular machine. Rather than that, they can be considered mean values of those coefficients commonly encountered. All the qualitative results presented in the sequence hold even if the weighting coefficients in the above formula are different for a specific computing platform [3], [12].

In rest of the paper the following notation is used for convenience:

- CLRES: Centralized Lainiotis-filter based Riccati Equation Solution
- CKRES: Centralized Kalman-filter based Riccati Equation Solution
- DLRES: Distributed Lainiotis-filter based Riccati Equation Solution
- DKRES: Distributed Kalman-filter based Riccati Equation Solution

### A. Centralized Approaches: Computational Requirements

The computational requirements for the sequential implementation of the centralized approaches (in terms of normalized operations), when a time varying linear model is used, are summarized in the following tables [12]: (See Tables I, II).

For a time invariant or a periodic system the computational requirements of the CLRES approach are divided into two parts. The first part, called preliminary, summarizes the computational requirements for the operations that are calculated only once (or during the system's period) and the second part, named per step operations, summarizes the requirements for the per step calculations. (See Table III).

**Table I** Centralized Approaches. Time Varying Linear Model

| <i>CLRES</i> | <i>normalized operations</i>  |
|--------------|---|
|              | $12.5n^3 + 14.5n^2 + 58n + 2.5m^3 + 10m^2 + 8.5nm^2 + 20n^2m + 14.5nm + 32.5m - 18$ |

**Table II** Centralized Approaches. Time Varying Linear Model

| <i>CKRES</i> | <i>normalized operations</i>   |
|--------------|--|
|              | $7.5n^3 + 6.5n^2 + 2.5n^2m + 7.5nm^2 + 3mn + 6m^2 + 2.5m^3 + 31.5m - 4n - 6$ |

**Table III** Centralized Approaches. Time Invariant or Periodic Linear Model

| <i>CLRES</i>  | <i>normalized operations</i>  |
|---------------|---|
| 1 prelim. op. | $2.5m^3 + 10m^2 + 8.5nm^2 + 20nm^2 + 4.5nm - 0.5n^2 + 32.5m - 0.5n - 6$ |
| per step op.  | $12.5n^3 + 14n^2 + 58n - 12$  |

**B. Distributed Approaches: Computational Requirements**

The computational requirements (in terms of normalized operations) of the distributed Riccati solutions, when a time varying linear model is used, are shown in the sequence. (See Tables IV, V).

For a time invariant or a periodic system the computational requirements of the DLRES approach are divided into two parts. As before, the first part, called preliminary, summarizes the computational requirements for the operations that are calculated only once (or during the system’s period) and the second part, named per step operations, summarizes the requirements for the per step calculations. (See Table VI).

*Remarks*

1. The time required to solve the Riccati equation for a time invariant or periodic model by direct use of the Distributed Kalman filter is identical to that required to solve the Riccati equation for the time varying system due to the fact that the Distributed Kalman filter equations have exactly the same form for time varying, time invariant or periodic state space models.
2. The computational complexity of the RE solution for a time invariant or periodic model using the Distributed Lainiotis filter is much less than the complexity associated with the solution of the Riccati equation for a time varying system. If a time invariant system is assumed, the matrices  $P_n(\cdot)$ ,  $O_n(\cdot)$ ,  $\Phi_n(\cdot)$  are time invariant so their calculation is required only once. If the actual system is periodic the above mentioned matrices become periodic in nature, with period equal to the period of the model RP. In this case, their calculation is required only within the

**Table IV** Distributed Approaches. Time Varying Linear Model

| <i>DKRES</i>      | <i>normalized operations</i>  |
|-------------------|---|
| central processor | $10n^3 + 6.5n^2 + 32.5n - 6 + r(n^2 + n)$   |
| local processor   | $7.5n^3 + 2n^2 + 0.5n + 2.5m_i^3 + 10n^2m_i + 7.5nm_i^2 + 2nm_i + 7.5m_i^2 + 32.5m_i - 6$ |

**Table V** Distributed Approaches. Time Varying Linear Model

| <i>DLRES</i>      | <i>normalized operations</i>   |
|-------------------|--|
| central processor | $25n^3 + 11n^2 + 33n - 6 + 0.5r(n^2 + n)$  |
| local processor   | $2.5m_i^3 + 6m_i^2 + 32.5m_i + 5nm_i^2 + 2.5n^2m_i + 1.5m_i n - 0.5n^2 - 0.5n - 6$ |

**Table VI** Distributed Approaches. Time Invariant Linear Model

| <i>DLRES</i>                | <i>normalized operations</i>  |
|-----------------------------|---|
| central proc. (per step)    | $10n^3 + 4n^2$  |
| central proc. (preliminary) | $15n^3 + 7n^2 + 33n - 6 + 0.5r(n^2 + n)$  |
| local processor             | $2.5m_i^3 + 6m_i^2 + 32.5m_i + 5nm_i^2 + 2.5n^2m_i + 1.5m_in - 0.5n^2 - 0.5n - 6$ |

time interval  $[0, RP]$  and not within the whole processing interval. It must be emphasized that in the case of a time invariant or periodic system there is no per step calculations at the local level using the DLRES approach. The required calculations have to be performed once in the case of the time invariant models or only during the period interval in the case of periodic systems. In addition, the results can be transmitted to the central processor and stored there. In this way, there is no need for further processing at the local level or communication between the central agent and the local systems. Further, the vast majority of the calculations in the central processor becomes non recursive and it is sufficient to be done only once (for time invariant systems) or only during the systems period (for periodic models).

The distributed algorithm of theorem 2 is the only one which possesses such attractive computational characteristics. The algorithm is designed to take full advantage of the new parallel processing capabilities available. There are two ways in which we achieve parallelism using the new Riccati equation solution developed in theorem 2. Namely,

- parallelism in system's level
- parallelism in computations during one sampling period

Parallelism in system's level is based on the fact that every local processor can execute the calculations required by the central agent in parallel with the other local processors. In this way, in one step of the algorithm  $r$ -operations are executed in parallel at the different local processors.

Parallelism in the computations' level during one sampling period (one step) is based on the fact that every solution involves during each step the evaluation of a set of arithmetic expressions. From theorem 2, it can be easily seen that the calculations in equations (21)–(23) are not recursive and can of course be made in parallel. Thus, even if the model is time varying these calculations can be executed in parallel in one step using a parallel machine.

Up until now we have discussed the computational requirements of the different approaches to the discrete Riccati equation solution when implemented in a centralized or decentralized fashion. An answer however, to a fundamental question posed by filter designers and users was not given. Is the difficulty of implementation inherent in a distributed realization compensated by the gain in computing speed? To answer this question, a typical multisensor example is used.

In this typical multisensor problem, seismic signal processing for oil exploration, a time invariant wavelet is utilized to describe the signal received by the seismic sensors [14], [9], [12]. Usually a convolution summation of order  $n = 4$  describes the seismic wavelet.

**Table VII** Seismic Wavelet Estimation Problem

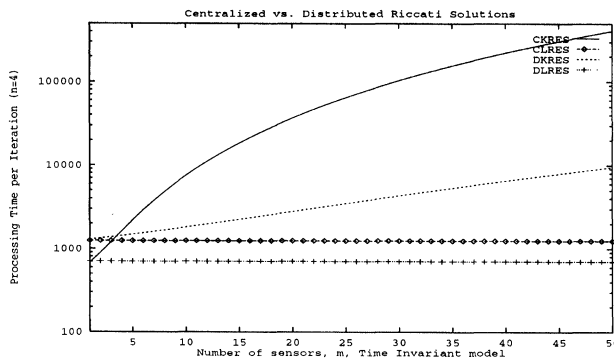
| <i>method</i> | <i>normalized operations</i> |
|---------------|------------------------------|
| CKRES         | $2.4e + 09$                  |
| CLRES         | $4.24e + 04$                 |
| DKRES         | $2.61e + 06$                 |
| DLRES         | $3.37e + 03$                 |

The number of sensors (geophones) utilized to capture the seismic reflection varies from a few hundred to several thousand. For this time invariant seismic wavelet of order  $n = 4$ , assuming  $m = 1000$  sensors divided in  $r = 10$  local subsystems (geophone clusters), the normalized operations (computational requirements) for the solution of the Riccati equation which is required in the source estimation process are summarized in the following table. (See Table VII).

In order to compare the algorithms' efficiency from a computational standpoint, the total processing time (total normalized operations) required by centralized and distributed algorithms was plotted against the total dimension of the measurement vector  $m$ . The total processing time for the distributed algorithms is obtained adding the computational requirements of one local processor to those of the central processing facility. In Fig. 1, the complexity for a time invariant system with state dimensionality  $n = 4$  is depicted. In Fig. 2, the requirements for a time varying system with  $n = 1$  are plotted. In both cases the number of local subsystems is considered fixed at  $r = 5$ . The efficiency of the to the computing facility at each time step.

### V. CONCLUSIONS

The centralized and the distributed approaches to the solution of the discrete Riccati equation for a multisensor linear model were discussed in this paper. State variable concepts and optimal estimation algorithms have been utilized to obtain the solution in a distributed parallel fashion. A Riccati equation solution based on the distributed form of the linear Lainiotis filter was compared, in terms of computational efficiency, with the



**Figure 1** Centralized vs. Distributed Riccati Solutions. Time Invariant model.

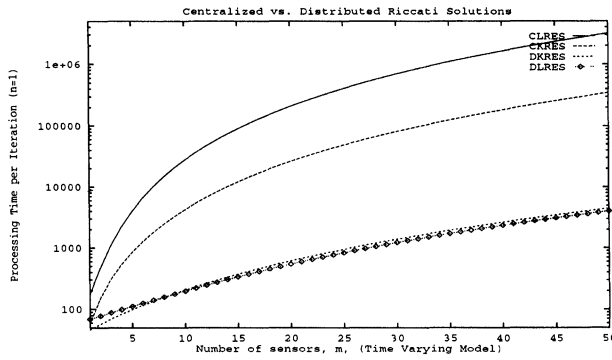


Figure 2 Centralized vs. Distributed Riccati Solutions. Time Varying model.

RE solution based on the distributed Kalman filter. The comparative evaluation of these approaches have been made using a typical multisensor scenario. The results indicate that the RE solution based on the distributed Lainiotis filter leads to orders of magnitude decrease in processing time while maintains the same level of performance.

## VI. APPENDIX

### Proof of Theorem 2.

Our objective is to eliminate the measurement related matrices from the solution. Thus, we modify the equations of the solution based on the centralized Lainiotis filter. The most important step is to obtain an expression for the nominal observability matrix  $O_n(k + 1)$  which will not include any matrix associated with the measurement equation.

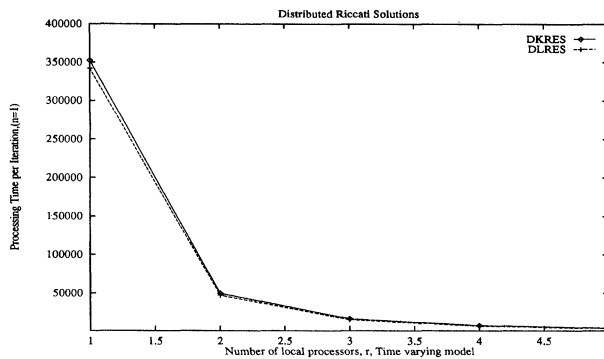


Figure 3 Distributed Riccati Solutions. Time varying model.

We start with an interim result, the expression for the filter transition matrix  $\Phi_n(k+1, k)$ . The filter transition matrix in the centralized linear Lainiotis filter is defined as:

$$\Phi_n(k+1, k) = [I - K_n(k+1)H(k+1)]\Phi(k+1, k) \quad (24)$$

According to this formula the filter transition matrix is calculated in terms of the filter gain and the overall output matrix  $H(k+1)$ . The equations of the standard Kalman filter are used to eliminate these quantities from the expression for the filter transition matrix. From the Kalman filter the estimation covariance is given as:

$$P(k|k) = [I - K(k)H(k)]P(k|k-1) \quad (25)$$

Since covariance matrices are square matrices, the inverse always exist thus, multiplying both terms of (25) from the right with  $P^{-1}(k|k-1)$  the following expression can be obtained:

$$P(k|k)P^{-1}(k|k-1) = [I - K(k)H(k)] \quad (26)$$

In the nominal Kalman filter used in the derivation of the centralized linear Lainiotis filter it is assumed that:

$$P(k|k-1) = Q(k-1)$$

since we initialize the nominal filter at each step with zero nominal initial conditions. If we combine the above two equations, we can claim that:

$$P_n(k|k)Q^{-1}(k-1) = [I - K_n(k)H(k)] \quad (27)$$

Therefore, the filter transition matrix for the distributed filter can be written as:

$$\Phi_n(k+1, k) = P_n(k|k)Q^{-1}(k-1)\Phi(k+1, k) \quad (28)$$

The nominal observability matrix in the centralized linear Lainiotis filter is defined as:

$$O_n(k+1) = K_m(k+1)H(k+1)\Phi(k+1, k) \quad (29)$$

$$K_m(k+1) = \Phi^T(k+1, k)H^T(k+1)P_z^{-1}(k+1|k) \quad (30)$$

Once again we aim to eliminate all the measurement related matrices from the calculations. Simple algebraic calculations of the standard Kalman filter gain equation lead us to the following solution:

$$H^T(k)P_z^{-1}(k|k-1) = P^{-1}(k|k-1)P(k|k)H^T(k)R^{-1}(k) \quad (31)$$

Thus, the nominal Kalman gain the distributed algorithm can be re-written as:

$$K_m(k+1) = \Phi^T(k+1, k)Q^{-1}(k)P_n(k+1|k+1)H^T(k+1)R^{-1}(k+1)H(k+1)\Phi(k+1, k) \quad (32)$$

In the information form of the Kalman filter, the expression for the estimation covariance is as follows:

$$P_n^{-1}(k|k) - P_n^{-1}(k|k-1) = H^T(k)R^{-1}(k)H(k) \quad (33)$$

Taking into consideration the special form of the nominal Kalman filter with the zero initial conditions, we can claim that the equivalent update formula is:

$$P_n^{-1}(k|k) - Q^{-1}(k-1) = H^T(k)R^{-1}(k)H(k) \quad (34)$$

Utilizing this result the nominal observability matrix for the distributed solution can be written after some simple manipulations as follows:

$$F_n(k+1) = Q^{-1}(k)\Phi(k+1, k) \quad (35)$$

$$O_n(k+1) = F_n(k+1)^T[Q(k) - P_n(k+1|k+1)]F_n(k+1) \quad (36)$$

Equation (16) which provides the nominal estimation covariance used in our distributed solution emanates from the distributed form of the Kalman filter taking into consideration that:

$$H^T(k)R^{-1}(k)H(k) = \sum_{i=1}^r H_i^T(k)R_i^{-1}(k)H_i(k) \quad (37)$$

from the formulation and the assumptions of the distributed estimation problem and that:

$$P_n^{-1}(k|k) - Q^{-1}(k-1) = H^T(k)R^{-1}(k)H(k) \quad (38)$$

In the *DLRES* based solution equation (14) which provides the estimation covariance is identical to the expression used in the centralized linear Lainiotis filter.

Finally, the expressions in (21)–(23) are obtained using a Kalman filter with zero initial conditions at each local sub-system.

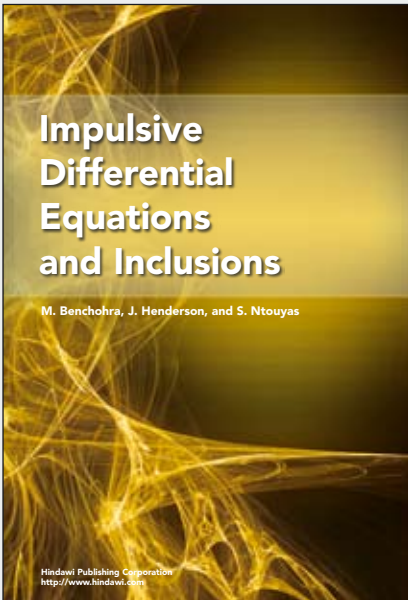
## References

1. N.A. Carlson, 'Federated Square Root Filter for Decentralized Parallel Processes,' IEEE Trans. on Aerospace and Electronic Systems, vol. AES-26, 1990, pp. 517–525.
2. H.R. Hashemipour, S. Roy, A.J. Laub, 'Decentralized Structures for Parallel Kalman Filtering,' IEEE Trans. on Automatic Control, vol. AC-33, 1988, pp. 88–94.

3. S.K. Katsikas, S.D. Likothanassis, D.G. Lainiotis, 'On the parallel implementation of linear Kalman and Lainiotis filters and their efficiency,' *Signal Processing*, vol. 25, 1991, pp. 289–306.
4. D.G. Lainiotis, 'Optimal Nonlinear Estimation,' *Int. Journal of Control*, 1971, pp. 1137–1148.
5. D.G. Lainiotis, 'Discrete Riccati Equation Solution: Partitioned Algorithms,' *IEEE Trans. on Automatic Control*, vol. AC-20, 1975, pp. 255–257.
6. D.G. Lainiotis, 'Partitioned Riccati Solutions and Integration Free Doubling Algorithms,' *IEEE Trans. on Automatic Control*, vol. AC-21, 1976, pp. 677–688.
7. D.G. Lainiotis, 'Partitioning: A unifying framework for adaptive systems I: Estimation,' *Proceedings of IEEE*, vol. 64, 1976, pp. 1126–1143.
8. D.G. Lainiotis, K.S. Gonvidaraj, D. Andrisanni II, 'Nonsymmetric Riccati Equation: Partitioned Algorithms,' *Computer and Electrical Eng.*, vol. 5, 1978, pp. 109–122.
9. D.G. Lainiotis, S.K. Katsikas, S.D. Likothanassis, 'Optimal seismic deconvolution,' *Signal Processing*, vol. 15, 1988, pp. 375–404.
10. D.G. Lainiotis, K.N. Plataniotis, C.J. Charalampous, 'Distributed Computing Filters: Multisensor Marine Applications,' *IEEE OCEANS'92 Proceedings*, 1992, pp. 265–270.
11. D.G. Lainiotis, N.D. Asimakis, S.K. Katsikas, 'New Doubling Algorithms for the Discrete Periodic Riccati Equation,' *Applied Mathematics and Computation*, vol. 1, 1993, pp. 467–476.
12. K.N. Plataniotis, *Distributed Parallel Processing State Estimation Algorithms*, Ph.D. Dissertation, Florida Institute of Technology, Melbourne, Florida, 1994.
13. K. Watanabe, *Adaptive Estimation and Control: The Partitioning Approach*, Prentice Hall, 1992.
14. A.S. Willsky, M.G. Bello, D.A. Castanon, 'Combining and updating of local estimates and regional maps along sets of one dimensional tracks,' *IEEE Trans. on Automatic Control*, vol. AC-27, 1982, pp. 799–812.

# Impulsive Differential Equations and Inclusions

M. Benchohra, J. Henderson, and S. Ntouyas



Functional differential equations and inclusions have assumed an increasing relevance, mainly due to the great variety of problems arising in sciences, engineering, and technology which can be described by such types of equations and inclusions. This book is devoted to this rapidly developing theory of differential equations and inclusions with special emphasis on such equations and inclusions subject to impulses. In addition, the book includes a wide account of various applications to physical problems.

Limited-Time Promotional Offer. Buy this title NOW at **20% discount plus Free Shipping.**

The book consists of twelve chapters. Initial value problems and boundary value problems for both impulsive differential equations and differential inclusions, as well as for both impulsive functional differential equations and functional differential inclusions, or for both neutral functional differential equations and neutral functional differential

inclusions, are studied. Moreover, results on impulsive semilinear ordinary differential inclusions and functional differential inclusions satisfying nonlocal boundary conditions are given extensive attention. Positive solutions and multiple positive solutions for impulsive ordinary differential equations and functional differential equations, as well as impulsive differential inclusions satisfying periodic boundary conditions, are also studied.

Some of these results are extended to functional differential equations and functional differential inclusions with variable times, to semilinear functional differential equations and functional differential inclusions for operators that are nondensely defined on a Banach space, to impulsive hyperbolic differential inclusions, and to impulsive dynamic equations on time scales.

Each chapter concludes with a section devoted to notes and bibliographical remarks. This book is addressed to a wide audience of specialists such as mathematicians, engineers, biologists, and physicists.

**CMIA Book Series, Volume 2, ISBN: 977-5945-50-X**

Please visit <http://www.hindawi.com/cmia.2.html> for more information about the book. To place an order while taking advantage of our current promotional offer, please contact [books.orders@hindawi.com](mailto:books.orders@hindawi.com)