Hierarchical Multi-rate Measurement Fusion for Estimation of Dynamical Systems

Manoranjan Majji*, Jeremy J. Davis †
John L. Junkins ‡
Texas A&M University, College Station, TX, 77843-3141, United States

Fusion algorithms for optimal estimation of dynamical systems are presented. The two algorithms presented, focus on fusion in the presence of asynchronous measurements that are obtained at differing sampling instances and rates. First of these Kalman filter based schemes is based on multiple measurement model switching. The second scheme essentially propagates all the measurements until the update time. The decentralization of the filtering scheme, in which each measurement subsystem is equipped with a dedicated simple estimator and the outputs are combined by a central filter is a characteristic feature of the schemes presented. Features of the algorithms presented are discussed with application on a two wheel differentially driven robot equipped with redundant sensors in a decentralized framework.

I. Introduction

With increasing complexity of mechanical systems and advancement of sensor technology, there is an imminent need to enhance the theoretical tools for estimation and inference in the presence of redundancy in sensed information. The Kalman filter theory in the past 45 glorious years, has taken numerous forms and shapes, being tailor made to suit the problem at hand. Recently, the application of these algorithms to highly distributed and redundant sensor information processing has received the interest of engineers. The availability of cheap and reliable sensors has also contributed to this in-progress revolution.

I.A. Why Redundant Measurements?

In the ideal world where everything is linear (with Gaussian noise and with small number of degrees of freedom), we just need so many sensors as to satisfy the observability conditions for a dynamical system. But unfortunately most physical systems and measurement processes are nonlinear, we see models with higher dimensions and improved filtering algorithms are needed to optimize the architecture. Hence the so called observability (about a nominal) becomes a function of the physics of the measurement process, strongly coupled with the evolving “plant” dynamics. It is to overcome this ignorance, we would require more probes in to various internal parts of the plant, whose measurements help us make the required control decisions. Therefore, in many practical situations, redundancy of sensors may be required, thereby motivating the need for development of algorithms aimed at incorporating redundancy explicitly, implicitly and at several levels.

I.B. Data Fusion

Fusion can be perceived as a method of estimating information from varied sources. Ideally a single filter would take all these measurements and give a best estimate of the state of the over all system. In practice, this is undesirable because of the sheer redundancy of the information that the central filter receives. So, for fast and more efficient computations, we would like to use processed measurements at the central filters.

*Graduate Research Assistant, Department of Aerospace Engineering, Student Member, AIAA.
†Graduate Research Assistant, Department of Aerospace Engineering, Student Member, AIAA.
‡Distinguished Professor, Holder of Royce E. Wisenbaker Chair in Engineering, Regents Professor, Department of Aerospace Engineering Texas A&M University, College Station, TX, 77843-3141, Fellow, AIAA.
would require us to delegate the processing job to several filters located/embedded in the sensors themselves. This is an attractive scheme for parallelization. Such a vision for sensor networks has been put forward by many researchers in modern system theory.\(^1\) One such hierarchical setup is shown in fig. 1. We present two different workings in this framework. Any practical fusion approach has many challenges. Some of them are listed below

- Alignment of measurements
- Filter network resetting/tuning
- Sensor fault isolation/ repair /recalibration
- Delay in the subordinate filters / sensors
- Bias estimation and reduction
- Error propagation in the network
- Local observability based sensor weighting/ ignorance

Clearly, several issues remain to be addressed in this area and steps have been taken up in this direction.\(^1\) Some theoretical developments to address many of such challenges are in progress. For instance, Sinopoli et. al.\(^2\) develop critical arrival rate bounds for convergence of filters.\(^2\) There are similar bodies of literature addressing the issues above and more.

I.C. Scope and Layout

Focus of the present paper is the presentation of a hierarchical approach to multi rate filtering problems for state estimation. This is discussed in section 2. In most dynamical systems the measurement sets do not arrive at necessarily at the time steps when Kalman filter is updated. This has come to be known as the multi rate filtering problem.\(^3\) There is an increasing body of literature in this direction,\(^3, 4\) Therefore, the hierarchical filters developed in this paper are designed to align\(^5\) and handle multirate measurements. These results are elaborated on a two wheel differentially driven robot model with mice odometry and wheel encoder based position and orientation determination sub-systems.

![General Layout of the Proposed Schemes](image_url)
II. Hierarchical Filter Schemes

We propose the following architectures and evaluate the merits and demerits of each of them.

II.A. Architecture : Multiple Model Switching

Consider a sensor network in which the components provide measurements at different rates and at different time instances. By simple modifications to the original discrete time Kalman filter we obtain the following algorithm. Notice that the original Kalman Filter derivation does not place any restriction on uniform time updates. Therefore, this is the optimal solution to the asynchronous fusion problem.

II.A.1. Algorithm

<table>
<thead>
<tr>
<th>Discrete time Model</th>
<th>( x(t_{k+1}) = A(t_k)x(t_k) + B(t_k)u(t_k) + G(t_k)w(t_k) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement Model</td>
<td>( \hat{y}_j(t_p) = H_j(t_p)x(t_p) + v(t_p) ), At ( t_p ), the ( j^{th} ) sensor model is used</td>
</tr>
<tr>
<td>Noise Characteristics</td>
<td>( w(t_k) \sim N(0,R(t_k)), v(t_k) \sim N(0,Q(t_k)) )</td>
</tr>
<tr>
<td>Initialization</td>
<td>( \hat{x}(t_0) = \hat{x}_0, P_0 = E(\tilde{x}_0\tilde{x}_0^T) ), where, ( \tilde{x}_0 := \hat{x}_0 - x_0 ) is estimation error.</td>
</tr>
<tr>
<td>Propagation (State)</td>
<td>( \hat{x}^-(t_{p+1}) = A(t_p)\hat{x}^+(t_p)B(t_p)u(t_p) )</td>
</tr>
<tr>
<td>Propagation (Covariance)</td>
<td>( P^-(t_{p+1}) = A(t_p)P^+(t_p)A^T(t_p) + G(t_p)Q(t_p)G^T(t_p) )</td>
</tr>
<tr>
<td>Kalman Update (State)</td>
<td>( \hat{x}^+(t_p) = \hat{x}^-(t_p) + K_j(t_p)(\tilde{y}_j(t_p) - H_j(t_p)\hat{x}^-(t_p)) )</td>
</tr>
<tr>
<td>Covariance Update</td>
<td>( P^+(t_p) = (I - K_j(t_p)H_j(t_p))P^-(t_p) )</td>
</tr>
<tr>
<td>Kalman Gain</td>
<td>( K_j(t_p) = P^-(t_p)H_j^T(t_p)(H_j(t_p)P^-(t_p)H_j^T(t_p) + R_j(t_p))^{-1} )</td>
</tr>
</tbody>
</table>

II.A.2. Features

We observe the following features of the algorithm.

- Notice that the optimality of this algorithm comes at an (often heavy) price of book keeping and extensive computations.
- The algorithm is extremely convenient for inclusion and deletion of sensors from the network and continuing Kalman updates from the remainder of the network.
- This feature actually allows one to choose the most favorable of the sensors and discard/discount the measurements from the most unfavorable of the sensors, depending on the operating conditions in the state space.

Application of this scheme to a realistic hierarchical scheme is presented in the next section.

II.B. Architecture : Multirate Propagation Update Scheme

In this scheme, a central filter with a fixed update rate is implemented with accompanying subsystems providing the propagated measurements to the central filter at its update time. The architecture is as follows.
II.B.1. Algorithm

<table>
<thead>
<tr>
<th>Discrete time Model</th>
<th>( x(t_{k+1}) = A(t_k)x(t_k) + B(t_k)u(t_k) + G(t_k)w(t_k) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurement Model</td>
<td>( \hat{y}<em>{\text{comp}}(t_p) = H</em>{\text{comp}}(t_p)x(t_p) + v(t_p) )</td>
</tr>
<tr>
<td>Noise Characteristics</td>
<td>( w(t_k) \sim N(0, R(t_k)), v(t_k) \sim N(0, Q(t_k)) )</td>
</tr>
<tr>
<td>Initialization</td>
<td>( \hat{x}(t_0) = \bar{x}_0, P_0 = E(\bar{x}_0\bar{x}_0^T) ), where, ( \bar{x}_0 := \hat{x}_0 - x_0 ) is estimation error.</td>
</tr>
<tr>
<td>Propagation (State) (Central Filter)</td>
<td>( \dot{\hat{x}}(t_{k+1}) = A(t_k)\hat{x}(t_k) + B(t_k)u(t_k) )</td>
</tr>
<tr>
<td>Propagation (Covariance) (Central Filter)</td>
<td>( P^{-1}(t_{k+1}) = A(t_k)P^{-1}(t_k)A^T(t_k) + G(t_k)Q(t_k)G^T(t_k) )</td>
</tr>
<tr>
<td>Kalman Update (State)</td>
<td>( \hat{x}^+(t_k) = \hat{x}^-(t_k) + K_{\text{comp}}(t_k)(\hat{y}<em>{\text{comp}}(t_k) - H</em>{\text{comp}}(t_k)x^-(t_k)) )</td>
</tr>
<tr>
<td>Covariance Update</td>
<td>( P^+(t_k) = (I - K_{\text{comp}}(t_k)H_{\text{comp}}(t_k))P^-(t_k) )</td>
</tr>
<tr>
<td>Kalman Gain</td>
<td>( K_{\text{comp}}(t_k) = P^{-1}(t_k)H_{\text{comp}}(t_k) )</td>
</tr>
<tr>
<td>( j^{th} ) Measurement System Propagation</td>
<td>( \hat{y}(t_k) ) from ( \hat{y}(t_p) ) is determined as ( \tilde{y}(t_k) = H_j(t_k)\hat{x}^-(t_k) = H_j(t_k)(A'(t_p)\hat{x}^+(t_p)B'(t_p)u(t_p)) )</td>
</tr>
</tbody>
</table>

II.B.2. Features

At \( t_k \), the Composite sensor model is used. In the last part of the algorithm, the propagation from \( t_p \) to \( t_k \), until the next update so as to obtain an predicted measurement \( \hat{y}(t_k) \) is done with the calculation of appropriate \( A', B' \) discrete time system matrices.

- There is no switching between models. The update is scheduled at fixed time steps and occurs that way. The sensors that are not in sync with the filter are propagated.
- The algorithm is more decentralized and suited for applications involving filter networks with comparable computational facilities.
- Ideal for a continuous discrete Kalman Filter implementation.
- The sensor system propagation equations can be replaced with some curve fitting/approximation techniques in favor of computational cost and better accuracy in the propagation, introducing sub-optimality in the algorithm.

Equivalence of both the schemes is dependent on parameters such as the central kalman filter update rate and individual sensor system update rates. These are essentially two different implementations of the same filtering strategy.

### III. Application to Mobile Robots

To estimate their position and orientation, or pose, mobile robots utilize a variety of sensors for dead-reckoning, sometimes augmenting them with inertial measurement updates. Typically, dead-reckoning is accomplished through the use of wheel odometry which integrates measurements from incremental encoders mounted on the wheels. More recently, the drop in price of optical mice has made them a viable alternative for dead-reckoning with similar precision to wheel odometry. Since both methods are inexpensive and simple to implement on the same robot, we look at combining both measurements using our multi-rate Kalman filter to provide an improved pose estimate.

#### III.A. Dead-reckoning

Wheel odometry has been studied extensively to develop algorithms for different wheel configurations and to reduce systematic and non-systematic errors. Most of these methods are heuristically based, but nonetheless well-developed, and most mobile robots already have some form of these algorithms implemented. Thus, a decentralized approach makes the most sense for this sensor, since current algorithms can reduce wheel encoder measurements to reliable pose estimates. Since most systematic errors can be removed through calibration, and most non-systematic errors result in a reading that the wheel has travelled further than it actually has (such as slipping), we modelled the wheel encoders to produce readings with up to +2% error.
Optical mice have just recently begun to receive attention as dead-reckoning devices.\textsuperscript{10–14} Whereas they are not subject to problems such as slipping or creep of the wheels, new non-systematic errors occur due to their sensitivity to the height off the floor and varying calibrations when moving diagonally.\textsuperscript{12} In our model, we represent these errors with a $\pm 2\%$ on the mouse readings. Since each optical mouse provides two measurements, readings from two mice are necessary to estimate the three position and orientation values, but we have one extra piece of information. Bonardini developed a method for correcting non-systematic dead-reckoning errors using this fourth piece of information.\textsuperscript{14} However, since in practice more than two mice could be used to provide improved estimates, we use a simple least squares algorithm to estimate the pose of the robot from all four readings.

### III.B. Simulation Set-up

We look at a typical mobile robot configuration, similar to the one used by Bonardini.\textsuperscript{14} Our robot has two wheels that are differentially driven with free spinning castors providing additional support. Two mice are placed along a diagonal, as seen in Fig. 2. The wheels are located at $(\pm b/2, 0)$ with radius $r_w$, and the mice are located at $(\pm D/2, \pm L/2)$. The wheel encoders operate at a resolution of 1200 pulses per revolution and update their pose estimate at a frequency of $f_w$, while the optical mice update their pose estimate at a frequency of $f_m$. The speed of the robots over the test track is $v$. The values used for each of these constants is found in Table 1.

![Figure 2. Robot setup](image)

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b$</td>
<td>0.3 m</td>
</tr>
<tr>
<td>$D$</td>
<td>0.3 m</td>
</tr>
<tr>
<td>$L$</td>
<td>0.3 m</td>
</tr>
<tr>
<td>$v$</td>
<td>0.2 m/s</td>
</tr>
<tr>
<td>$f_w$</td>
<td>25 Hz</td>
</tr>
<tr>
<td>$f_m$</td>
<td>75 Hz</td>
</tr>
</tbody>
</table>

Borenstein developed the UMBmark test as a means for comparing and analyzing dead-reckoning systems.\textsuperscript{15} This test involves driving the robot counter-clockwise and then clockwise, five times each, around a 4 m by 4 m square. The robot drives the length of the side, stops, completes a 90 degree turn, and continues along the next side. As this is an established method for comparing results, we simulated our robot following the UMBmark test track as seen in Fig. 3.

### III.C. Fusing through Model Switching Approach

We implemented the model switching approach described above for our simulated mobile robot. Since both wheel odometry and optical mice provide updates in the form of incremental position and orientation, the best estimate from our filter at the time of the previous measurement by that same instrument was
incremented to provide a current state estimate. We assumed a constant jerk model. Figures 4 through 6 show the errors resulting from our filter compared to the errors sustained by either of the dead-reckoning systems alone.

Since both sensors provide only position and orientation information, we expect the Kalman filter to provide essentially a weighted average between the two measurement systems. We weighted the measurements from the mice higher than the measurements from the wheel encoders due to the inherent redundancy in the mice estimation as discussed in Section III.A. Figures 4 through 6 clearly show that such a weighted averaging is taking place. Compensating the pose information gathered by optical mice and wheel odometry with readings from a rate gyro - as is often done on mobile robots - would reduce error even further through our filter.
Figure 5. Error in y-position in mm.

Figure 6. Error in heading angle, $\Psi$ in rads.
III.D. Fusing through Multirate propagation approach

The multirate propagation approach, is computationally as expensive as the multimodel switching approach in its optimal implementation. Therefore, efforts are underway to implement high fidelity, multi-resolution approximation techniques\(^7\) and incorporate them for the measurement propagation to be updated by the central filter. Computational cost and other performance criteria of the alternative algorithms can be made only upon the successful implementation of these alternative propagation schemes. Hence a discussion on this algorithm and associated results need a little more time to mature.

IV. Conclusion and Future Directions

Two hierarchical multi-rate data fusion algorithms, based on the discrete time Kalman filter are presented. The algorithms essentially differ in the way they deal with the measurement alignment problem. The advantages of multiple model switching and multirate propagation update schemes are discussed and the multiple model switching architecture is demonstrated on a realistic mobile robot example. It was found from the example problem that for homogeneous sensor subsystems (filter subsystems), the centralized Kalman filter essentially produces a weighted average of the estimates of the subsystems, validating the architecture. In this scheme it is simple to add and remove sensors without altering the structure of the network. The multirate propagation scheme, where, the measurements are propagated by the subfilters before updating with the central filter is more decentralized and suited in filter networks where individual components of the network are endowed with comparable computational capabilities. However this scheme could use curve fitting techniques for this prediction step.\(^7\) Extensions of these algorithms in to continuous discrete schemes and extended Kalman filters is obvious and hence not mentioned. The mobile robot test simulations need to be demonstrated. Future efforts involve refinement and development of similar algorithms for optimal data fusion. Test examples offer a great deal of optimism for implementation of these filtering strategies.

Acknowledgments

The first author wishes to acknowledge the support of Texas Institute of Intelligent Bio Nano Materials and Structures for Aerospace Vehicles funded by NASA Cooperative Agreement No. NCC-1-02038.

References