Proceedings of the 5th WSEAS Int. Conf. on APPLIED INFORMATICS and COMMUNICATIONS, Malta, September 15-17, 2005 (pp330-334)

Multi sensor data fusion with filtering

Richard A. Wasniowski
Computer Science Department
California State University
Carson, CA 90747,USA

Abstract: - The purpose of data fusion is to produce an improved model or estimate of a system from a set of independent data sources. There are various multisensor data fusion approaches, of which Kalman filtering is one of the most significant. Methods for Kalman filter based data fusion include measurement fusion and state fusion. This paper gives a simple a review of fusion and state fusion, and secondly proposes new integrated method of state fusion based on fusion procedures at the prediction and update level. To illustrate application, a simple example is performed to evaluate the proposed method.

Key-Words: -Multisensor data fusion, sensors network, data fusion, filtering.

1 Introduction

The widespread distribution and availability of small-scale sensors, actuators, and embedded processors is transforming the physical world into a computing platform. Sensor networks that combine physical sensing capabilities such as temperature, light with networking and computation capabilities are becoming ubiquitous. Applications range from environmental control, warehouse inventory, and health care to scientific and military scenarios. The goal of these sensor networks is to collect information from the area and to relay it through the network. Since several years, research in telecommunication, wireless networks, and signal processing has focused on this topic that raises new challenges in wireless communication. Wireless sensors networks recently received tremendous attention because of its promise of a wide range of potential applications in both civil and military areas. Wireless sensor network consists of a large number of small sensor nodes with sensing, data processing, and communication capabilities, which are deployed in a region of interest and collaborate to accomplish a common task, such as environmental monitoring. Distinguished from traditional wireless networks, wireless sensors networks are characterized by dense node deployment, frequent topology change, and serious power, computation, and memory constraints. These unique characteristics and constraints present many new challenges to the design such as energy conservation, self-organization, efficient data dissemination, and fault tolerance. Sensor integration and fusion is a prerequisite to exploiting the inherent advantages of multi-sensor systems over single sensor systems. Using a single sensor, we can monitor objects with a precision and accuracy that depend on the sensor characteristics. By using multiple sensors to observe and monitor an object, we can obtain multiple viewpoints, extended coverage both spatially and temporally, reduce the ambiguity and obtain more precise estimate of object behaviour than that is possible through the best individual sensor. The use of sensory data from a range of disparate, multiple sensors are to automatically extract the maximum amount of information possible about the sensed environment under all operating conditions. Increased performance, reliability, data rates, and autonomy, coupled with increased complexity, diverse uncertain operating environments, requires the automated intelligent combination of data from multiple sensors to derive less ambiguous/uncertain information about the desired state. While the concept of data fusion is not new, the emergence of new sensors, advanced processing techniques, and improved processing hardware make real-time fusion of data increasingly possible. This makes the system less vulnerable to failures of a single component and generally provides more accurate information. In addition several readings from the same sensor are combined, making the system less sensitive to noise and anomalous observations. The objective of this paper is to develop a Kalman based fusion model to give a better, state estimate at each step of system operation.

2 Multisensor networks

Sensor networks consist of very large numbers of low-cost devices, each of which is a data source, measuring some quantity the object's location, or for example the ambient temperature and requiring some data fusion model for their operation
(See Fig 1)

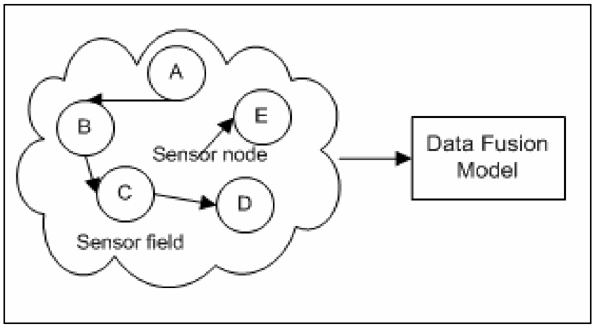

Fig.1 Sensor network.

These networks provide important data sources and create new data management requirements. For instance, these sensors are generally self powered, wireless devices. Such a device draws far more power when communicating than when computing. Thus, when querying the information in the network as a whole, it will often be preferable to distribute as much of the computation as possible to the individual nodes. In effect, the network becomes a new kind of database, whose optimal use requires operations to be pushed as close to the data as possible. Query execution on sensor networks requires a new capacity: the ability to adapt to rapidly changing configurations, such as sensors that die or disconnect from the network. In sensor networks individual sensor nodes are connected to other nodes in their neighborhood through a wireless network, and they use a multihop routing protocol to communicate with nodes that are spatially distant. Sensor nodes also have limited computation and storage capabilities: a node has a general-purpose CPU to perform computation and a small amount of storage space to save program code and data. A sensor node has one or more sensors attached that are connected to the physical world. Example sensors are temperature sensors or light sensors. Thus each sensor is a separate data source that generates records with several fields such as the id and location of the sensor that generated the reading, a time stamp, the sensor type, and the value of the reading. Records of the same sensor type from different nodes have the same schema, and collectively form a distributed table. The sensor network can be considered a large distributed database system consisting of multiple tables of different types of sensors. Sensor data might contain noise, and it is often possible to obtain more accurate results by data fusion from several sensors. For example, when monitoring the concentration of a dangerous bio-chemical in an area, one possible query is to measure the average value of all sensor readings in that region, and report whenever it is higher than some predefined threshold. We consider the sensor network as a large distributed database system, namely sensor database. Recent development of sensor database systems has attracted more and more interests in the querying performance for sensor network. Multiple sensor networks may be classified by how the sensors in the network interact. Three classes are defined as complementary, competitive and cooperative. Sensors are complementary when they do not depend on each other directly, but can be combined to give a more complete image of the environment. Complementary data can often be fused by simply extending the limits of the sensors. Sensors are competitive when they provide independent measurements of the same information. They provide increased reliability and accuracy. Because competitive sensors are redundant, inconsistencies may arise between sensor data, and care must be taken to combine the data in a way that removes the uncertainties. When done properly, this kind of data fusion increases the robustness of the system. Sensors are cooperative when they provide independent measurements, and when combined provide information that would not be available from any one sensor. Cooperative sensor networks take data from simple sensors and construct a new abstract sensor with data that does not resemble the readings from any one sensor.

3 Data fusion and filtering

The main task of a sensors network is to provide information about a process variable in the environment by taking measurements, and because these measurements are noisy and are taken at discrete points in time, it is necessary to fuse multiple measurements to reconstruct the parameter of interest. In general, given an observation vector corresponding to time, we want to estimate a process state vector. Manyika and Durrant Whyte distinguish the following three cases [14]: *Smoothing:* The change of a process entity shall be reconstructed after a series of measurements has been performed. For each instant of interest, several measurements from previous, actual, and following instants are used in order to estimate the value of the process variable. While the measurements have to be recorded in real time, the smoothing algorithm can be performed offline; *Filtering:* The actual state of a process entity shall be estimated by using an actual measurement and information gained from previous measurements. Usually, filtering is performed in real time; *Prediction:* The actual state of a process entity shall be estimated by using a history of previous measurements. The prediction problem requires an adequate system model in order to produce a meaningful estimation. Typically, prediction is performed in real time. Many filtering algorithms cover all three aspects. Filtering and prediction are fundamental elements of any tracking system The stochastic Kalman Filter uses a mathematical model for filtering signals using measurements with a respectable amount of statistical and systematical errors. The method was developed by Kalman and Bucy [11]. The conventional state-vector fusion and measurement fusion are two kinds of methods for Kaman filter based data fusion, and the conventional measurement fusion has lower estimation error but a higher computational cost. There is a growing interest in using Kalman filter models for data fusion. In turn, it is of considerable importance to represent Kalman filter in neural forms with local learning rules. To our best knowledge, Kalman filter has not been given such local representation. It seems that the main obstacle is the dynamic adaptation of the Kalman-gain. Here, a neural representation is presented, which is derived by means of the recursive prediction error method. We show that this method gives rise to attractive local learning rules and can adapt the Kalman gain. Multiple process models offer a number of important advantages over single model estimators. Multiple models allow a modular approach to be taken. Rather than develop a single model which must be accurate for all possible types of behaviour by the true system, a number of different models can be derived. Each model has its own properties and, with an appropriate choice of a data fusion algorithm, it is possible to span a larger model space. Four different strategies for multiple model estimation can be examined: multiple model switching, multiple model detection, multiple hypotheses testing, and multiple model fusion. Although the details of each scheme are different, the three first all use fundamentally the same approach. The designer specifies a set of models. At any given time only one of these models is correct and all the other models are incorrect. The different strategies use different types of test to identify which model is correct and, once this has been achieved, the information from all the other models is neglected. The latter strategy, model fusion utilize that process models are a source of information and their predictions can be viewed as the measurement from a virtual sensor. Therefore, multiple model fusion can be cast in the same framework as multisensor fusion and a Kalman update rule can be used to consistently fuse the predictions of multiple process models together. This strategy has many important benefits over the other approaches to multiple model management. It includes the ability to exploit information about the differences in behaviour of each model. As a result, the fusion is synergistic: the performance of the fused system can be better than that of any individual model.

3.1 Fusion models

The purpose of data fusion is to produce a model or representation of a system from a set of independent data sources, from which a single view or perception of some external environment or system is found or detected; therefore data fusion is the continuous process of assembling a model of the domain of the interest utilizing data from disparate sources. The algorithm uses a predefined linear model of the system to predict the state at the next time step. Added to this is a component to update for errors in the model using the actual observations of the system. The prediction and update are combined using the Kalman gain which is calculated to minimize the mean-square error of the state estimate. The Kalman filter has found widespread application in data fusion problems and track fusion problems, see [14], in particular Manyika and Durrant-Whyte [14] have applied it extensively to robot localization, guidance and navigation. Other areas of application include target detection, multisensor, multi-target tracking, automatic target recognition, collision avoidance, etc. (see [1, 5, 9, 12, 15]). The conventional state-vector fusion and measurement fusion are two kinds of methods for Kalman filter based data fusion, and the conventional measurement fusion [14] has lower estimation error but a higher computational cost.

3.2 Neural Based Kalman model

Let us consider the following linear dynamical system:

Proceedings of the 5th WSEAS Int. Conf. on APPLIED INFORMATICS and COMMUNICATIONS, Malta, September 15-17, 2005 (pp330-334)

$\mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \mathbf{n}_t$ observation process

$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \mathbf{m}_t$ dynamics of hidden variables

where

$\mathbf{m}_t \propto N(0,\Pi)$, $\mathbf{n}_t \propto N(0,\Sigma)$

are independent noise processes.

The above notation is shorthand to denote a stochastic variable of expectation value $\mathbf{m}$ and covariance matrix Σ. Our task is the estimation of

hidden variables

$\mathbf{x}(t) \in \mathbf{R}^n$ given the series of observations $\mathbf{y}(\tau) \in \mathbf{R}^p$, $\tau \le t$.

For estimations in squared (Euclidean) norm and Gaussian noise, the optimal solution was derived by Rudolf Kalman ([14, see also 16]).

4 Experiments

Using sensor network deployment in our lab, we collected (every 30 seconds) from multiple sensors the following data: temperature (deg C), light (lumens) humidity (percent), and the voltage level (V) of the batteries at each node. The data was collected in the following format:

Time	NodeID	Temp	Light	Hum	Vol

This is a "real" noisy dataset, with lots of missing data, noise, and failed sensors giving outlier values, especially when battery levels are low. In order to assists in understanding the process of estimating the state of a system based on noisy output signals we developed interactive software application that is based on the Kalman Filter. This consists of two main parts, a simple one-dimensional filter and a collating multi-dimensional filter (see Fig 2). The user can set all input parameters through a single interface or by following a series of guided steps.

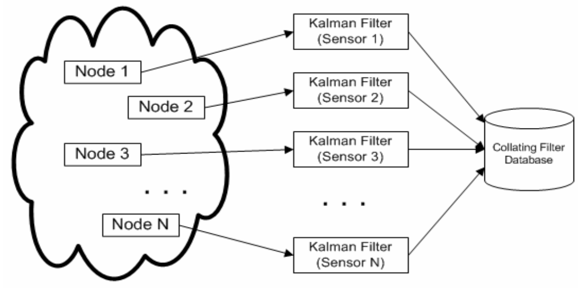

Fig. 2 System framework

Raw measurement data can be collected automatically or inserted manually by the user and the results of estimating the true system state are then presented as a series of graphs in real-time.

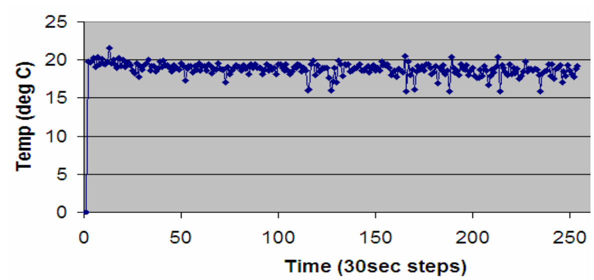

Fig. 3 Temperature data (filtered) from node 1

The typical graph of data is shown on Fig. 3. The examined architecture has some strengths and weaknesses for the modelling of sensor fusion applications. However, currently the Kalman Filter and Bayesian reasoning are the most frequently used tools in general data fusion models.

5 Discussion

This paper describes a novel software application that assists in understanding the process of estimating the state of a linear dynamic system based on noisy output signal measurements. An interactive Java tool, based on the Kalman Filter, is described. This consists of two main parts, a simple one-dimensional filter and a multi-dimensional filter tool. The user can set all input parameters through a single interface or by following a series of guided steps. Raw measurement data can be generated automatically or inserted manually by the user and the results of estimating the true system state are then presented as a series of graphs in real-time. The application is described through the use of two simple examples. Such an application could be used to teach signal and systems engineering.

Our future work will extend the current application to include signal filtering and prediction and will then look at the extended Kalman filter for non-linear systems.

References

[1] Y. Bar-Shalom, L. Campo, The common process noise on the two-sensor fused-track covariance, IEEE Transactions on Systems 22 (6) (1986) 803–805.

[2] A. Lorincz, B. Szatmary, G. Szirtes, Mystery of structure and function of sensory processing areas of the neocortex: A resolution, J. Comp. Neurosci. 13 (2002) 187-205.

[3] L. Ljung, T. Soderstrom, Theory and practice of recursive identification, MIT Press, Cambridge, MA, 1993.

[4] K. Chang, R. Saha, Y. Bar-Shalom, On optimal track-to-track fusion, IEEE Transactions on Systems 33 (4) (1997) 1271–1275.

[5] C. Harris, A. Bailey, T. Dodd, Multisensor data fusion in defense and aerospace, Journal of Royal Aerospace Society 162(1015) (1998) 229–244.

[6] C. Bell, V. Han, Y. Sugawara, K. Grant, Synaptic plasticity in a cerebellum-like structure depends on temporal order, Nature 387 (1997) 278–281.

[7] A. Bagchi, Optimal control of stochastic systems, Prentice Hall, New York, 1993.

[8] R. Elliott, A. Lakhdar, J. Moore, Hidden Markov models. Estimation and control, Springer Verlag, New York, 1995.

[9] K. Chang, Z. Tian, R. Saha, Performance evaluation of track fusion with information filter, in: FUSION'98 International Conference, 1998, pp. 648–655.

[10] C. Chui, G. Chen, Kalman filtering with real-time applications, in: Springer Series in Information Sciences, vol. 17, Springer-Verlag, New York, 1987, pp. 23–24.

[11] Q. Gan, C. Harris, Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion, Research report, ISIS Research Group, Department Computer Science, University of Southampton, 1999.

[12] M. Liggins, C. Chong, I. Kadar, M. Alford, V. Vannicola, S. Thomopoulos, Distributed fusion architectures and algorithms for target tracking, Proceedings of IEEE 85 (1) (1997) 95–107.

[13] R. Luo, M. Kay, Multisensor integration and fusion in intelligent systems, IEEE Transactions on Systems, Man and Cybernetics 19 (5) (1989) 901–931.

[14] J. Manyika, H. Durrant-Whyte, Data Fusion and Sensor Management: A Decentralized Information—Theoretic Approach, Ellis Horwood, London, 1994.

[15] J. Roecker, C. McGillem, Comparison of two-sensor tracking methods based on state vector fusion and measurement fusion, IEEE Transactions on Systems 24 (4) (1988) 447–449.

[16] R. Wasniowski, Interfacing of sensor networks to grid computing, report 2004.

[17] Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. SIGMOD Record 3(3), 2002.