รายงานวิจัยฉบับสมบูรณ์

โครงการ

การประยุกต์ใช้แบบจำลองชนิด สเปซิโอ เทมโพรอล สเตท สแปซ และตัวกรองคาลมาน ในระบบห่วงโซ่แบบพลวัต

โดย
ดร. พิสุทธิ์ อภิชยกุล

30 สิงหาคม 2562

สัญญาเลขที่ R2558C143

รายงานวิจัยฉบับสมบูรณ์

โครงการ

การประยุกต์ใช้แบบจำลองชนิด สเปซิโอ เทมโพรอล สเตท สแปซ

และตัวกรองคาลมาน ในระบบห่วงโซ่แบบพลวัต

โดย

ดร. พิสุทธิ์ อภิชยกุล

สนับสนุนโดย

งบประมาณรายได้มหาวิทยาลัยนเรศวร

ปีงบประมาณ พ.ศ. 2558

Spatio-Temporal State Space Model and Kalman Filter Application in the Dynamic Supply Chain

PHISUT APICHAYAKUL [1]

[1]Faculty of Engineering, Naresuan University, Thailand.

Sincerely thank you NU for funding support.

Contents

1 Introduction **1**
1.1 Introduction 1
1.2 Research Objectives 2
1.3 Conceptual Framework of the Research Project 2
1.4 Literature review - State of research 2
1.5 Research Methodology 6

2 Spatio-Temporal State Space **7**
2.1 Spatio-Temporal State Space Model 7
2.2 Neighbourhood Structure 9

3 Estimation **13**
3.1 Modelling of STSS Model and EM Algorithm 13
3.1.1 E-step 14
3.1.2 M-step 14
3.2 EM Algorithm 16

4 Spatio-Temporal State Space for Supply Chain System **19**
4.1 Spatio-Temporal State Space for supply chain system 19
4.2 Simulation 23
4.2.1 A toy Model, Supply Chain System Example 1 23
4.2.2 A toy Model, Supply Chain System Example 2 23
4.2.3 A case study 26

5 Conclusion **31**
Recommendations 31

Appendix A **35**

List of Figures

1.1 Supply Chain System 4

2.1 Three tiers supply chain system 9
2.2 Forward flow of materials 10
2.3 Backward flow of information 10
2.4 Model structure for three tiers supply chain system 10

4.1 Supply Chain System Example 1 20
4.2 Supply Chain System Example 2 22
4.3 Quantity of the product at location L_1, Example 1. 24
4.4 Quantity of the product at location L_3, Example 1. 24
4.5 Input, inventory level of supply chain system, Example 1. . . . 25
4.6 Q function, Example 1. 25
4.7 Estimated A Matrix, Example 1. 26
4.8 Quantity of the product at location L_1, Example 2. 26
4.9 Quantity of the product at location L_3, Example 2. 27
4.10 Input, inventory level of supply chain system, Example 2. . . 27
4.11 Q function, Example 2. 28
4.12 Estimated A Matrix, Example 2. 28
4.13 Quantity of the product at location L_1 29
4.14 Quantity of the product at location L_3 29
4.15 Input, inventory level of supply chain system 30
4.16 Estimated A Matrix 30

List of Tables

3.1 The EM algorithm for state space models. 17

1

Introduction

"All models are wrong; some are useful"
– George E. P. Box, William Hunter and Stuart Hunter, *Statistics for Experimenters, second edition, 2005, page 440*

1.1 Introduction

In developing country, the logistic and supply chain management is one of a key to success. The overall system of supply chain is a complex system since there are several factors involving in the system. In order to create a model of the overall system accurately, two main characteristics: flexibility and informativeness are needed. Some of the control theories (e.g. Robust Control Theory, Predictive Control Theory) are dealing with the complex system, for example, the control of robot, the control of refinery. Is it possible to adapt the control theories into the application of industrial engineering, supply chain management? Is the state space model or spatio-temporal state space model suitable to model the supply chain system since the basic information or data is collected as a time series? Those are some questions related to the application of state space model in the area of supply chain management. Furthermore, the estimation methods, which are used in the field of control engineering, can be adapted to utilize in the supply chain management field, for example, Kalman filter. Kalman filter is a mathematic tool, which is employed for state estimation. The underlying system state is estimated from noisy data. Time series data of the interest system is collected, and then is used in the estimation procedure. The Kalman filter operates recursively in order to estimate the system state. Kalman filter is widely used in various fields, such as navigation control of aircraft, guidance and sensorless control.
Inventory control is another active research area in the field of industrial engineering. The demand forecasting is the key to improve inventory control performance. There are several methods used in forecasting the demand,

for instance, exponentially weighted moving average (EWMA), EWMA with trend and seasonal corrections. The use of Kalman filter to correctly forecast the demand is an interested issue since the demand itself is consisted of various types of uncertainty.
The project is firstly planned to study various supply chain systems in various organizations. It will concentrate on the Lower North Region of Thailand, where is the area of Naresuan University located. The information, which is gathered from this phase, will be used to construct a Spatio-temporal State Space Model by simulating through the MATLAB program. Next, the time series of demand in the organizations in the same area will be collected. The information in this phase will be used in the estimated procedure. At this stage, various types of estimation methods will be employed in order to compare variety of performance in individual estimators. Similar to previous phase, the MATLAB program (or perhaps other relevant programs) will be employed to calculate the information.
The result of the study is firstly expected to provide a proposed model and estimation method, which is potentially giving more or deeper understanding through supply chain system. Secondly, it is expected to provide an accurately prediction of demand. This result will assist an organization a better control of inventory. The more accuracy the system is, the more an organization reduces its cost, time investment and resources in its production line. The results can be applied and developed in wider areas and regions.

1.2 Research Objectives

To develop a model with a better understanding of the dynamic characteristics in a supply chain system.

1.3 Conceptual Framework of the Research Project

The study will utilize a spatio-temporal state space model associated with Kalman filter technique in describing a dynamic characteristic of supply chain system. Inventory is managed based on demand forecasting at various points of the supply chain from incomplete data.

1.4 Literature review - State of research

There are several definitions of a supply chain system. Recently, [13] defined a supply chain as a system that consists of all parties, directly or indirectly, in fulfilling a customer demand. A supply chain system includes not only the manufacturer and suppliers, but also transporters, warehouses, retailers, and even customers themselves. A typical supply chain is characterized as a forward flow of material and a backward flow of information, [2]. Many companies, both global and local, plan their operation (production, stock and distribution systems) based on customer demands, supply conditions, sales and raw material prices, [14]. Various industries improve their supply chain management through an integration of information technology into a decision-making procedure. Since the supply system consists of several factors and the interaction between those factors, it can be called a complex system. According to [16], a typical supply chain can often be complex due to large mesh of interlinked suppliers, manufacturers and distributors, as shown in figure 1.1. The fact that each participant (supplier, manufacturer and distributor) may be a member of a large number of other supplies chains. Finally, the dynamic nature of the supply chain. In supply chain system, it can be categories the whole system into a number of subsystems. [2], supply chain is comprised of two basic, integrated processes, First, the production planning and inventory control process consists of raw material scheduling and inventory control process, manufacturing design and scheduling process and material handling design and control process. Second, the distribution and logistics process consists of the transportation process, distribution facilities design and control, management of inventory and final product delivery. Since the supply chain system consists of several stages of operation, in the past, in each echelon would operate independently. The decision making in each echelon is made individually based on objectives of their particular activity. As a result, each echelon attempted to optimize its own operations in separation. In general, a global optimum cannot guarantee from a sequence of locally optimized.

In the past decade, researchers pay more attention on the performance, design and analysis of the supply chain as a whole rather than investigated the various processes within supply chains individually. The other interested aspect is to reveal the dynamics of the process involve. Static models are insufficient when dealing with the dynamic characteristics of the supply chain, which are due to demand fluctuations, lead-time delays, sale forecasting, etc. In particular, they are not able to describe, analyze, and fine remedies for a major problem in supply chains, which recently became known as "the bullwhip effect", [6]. The orders at the upstream of a supply chain have been observed to exhibit a higher level of variability than those at the downstream, which is nearer to the customer, [14]. [5] identified four major causes of the bullwhip effect. First, Demand forecasting updating, Secondly, Order batching, Third, Price fluctuation, and Finally, Rationing and shortage gaming.

A model that can capture the dynamic of the supply chain system is a key to make a good decision in supply chain system management. According to [4], multi-stage models for supply chain design and analysis can be divided into four categories, by methodology, 1.) Continuous time differential equation models 2.) Discrete time differential equation models 3.) Discrete

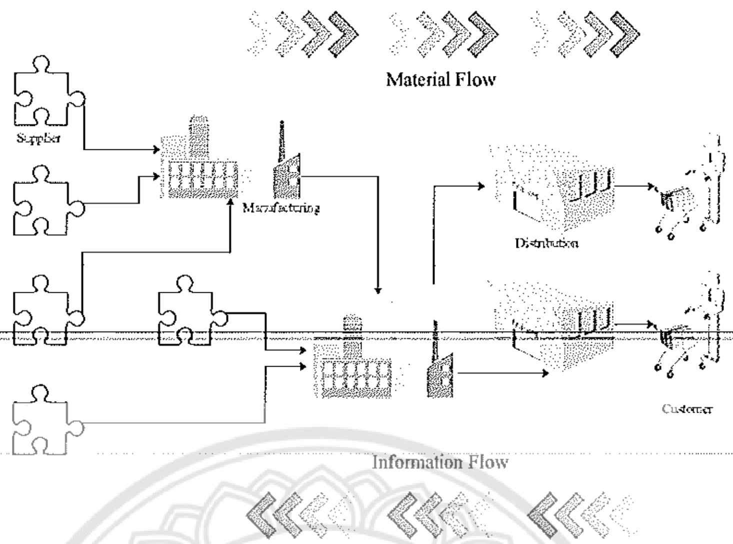

Figure 1.1: Supply Chain System

event simulation systems 4.) Operational research techniques

Control theory provides sufficient mathematical tools to analyze, design and simulate dynamic systems. [6] presented a review of advanced control methodologies to a supply chain management system. The majority of the paper focus on the application of classical control to supply chain management problem, where most of the analysis concerns linear systems and is performed in the frequency domain via Laplace transfer function and Z transfer function. The next session of the paper review the application of advanced control theory, optimal control, where the system dynamics are analyzed in the time domain and are described by state space models. The rest of the paper review the application of Model predictive control, robust control and approximate programming. State space model is a useful model in a dynamic control system. The state-space model or dynamic linear model, in its basic form, employs an order one, vector auto-regression as the state equation, [12],

$$x_t = Ax_{t-1} + w_t \tag{1.1}$$

where x_t is a state vector, A is pxp state matrix , for time points $t = 1, 2, \ldots, n$ Assume the w_t are px1 independent and identically distributed, zero-mean normal vectors with covariance matrix Q. The observation equation is

$$y_t = C_t x_t + v_t \tag{1.2}$$

where C_t is a qxp measurement or observation matrix. The additive observation noise v_t is assumed to be white and Gaussian with qxq covariance matrix

R. Aviv (2003), proposed a time-series framework for supply chain inventory management where demand process can be described in a linear state space form. Inventory is managed at various points of the chain (members), based on local information that each member observes and continuously updates.

Kalman filter technique is used to calculate minimum mean square error (MMSE) forecasts of future demands at each location of the supply chain. Inventory levels and the order sizes determine by updated forecast. A state space model of a multi-node supply chain is presented in [3]. The bullwhip effect (demand amplification) is characterized using the proposed state space model. A multi-node supply chain is controlled via local proportional inventory-replenishment policies. A state of supply chain is modeled by Augmented Trans-Net model, extend version of Tran Nets [16], with optimal estimation techniques of the extended Kalman filter. As a result, a less complex but more accurate model of state of supply chain is constructed, [15]. A supply chain consists of 7 place nodes and 3 transition nodes is modeled using the proposed method. The values of 2 place nodes are available from measurements that include the error variance and the measurement of one of the interest place node is missing. The Augmented Tran Net models with extend Kalman filter is used to estimate the value at those 3 interest place nodes. In conclusion, a proposed model is able to estimate the state of supply chain based on incomplete data and errors in data.

The Spatio-Temporal State Space (STSS) model is studied and presented by [7], since the state space framework is more convenient and widely used in control systems design. The spatio-temporal system can be observed as a set of spatially arranged and correlated time series, which motivates the use of a spatio-temporally indexed hidden, variable, [8]. In [9], the Spatio Temporal State Space (STSS) model, is used to model a smart structure, the smart beam. The defined neighborhood structure is introduced in order to describe the spatiotemporal neighborhood relationship. The proposed estimation method, the EM algorithm in association with the Kalman filter and smoother, is used to model smart structures based on the STSS model.

From above review, the Spatio-Temporal State Space model can be used as a supply chain model since the basic structure is match. In more details, the structure of each echelon in supply chain match with the neighborhood structure in STSS model and also, in each echelon, the demand is collected as a time series. The EM algorithm associated with Kalman filter and smoother is then used to estimate the parameters of the proposed model.

The structure of the paper is as follows. In section 2, spatio-temporal model is explained based on the work of [7], [8] and [9]. In section 3, introduction of neighbourhood structure associate in spatio-temporal state space model is presented. These can, later, adapt to use in modelling of a dynamic supply chain system. Finally, the parameter estmation procedure based on EM algorithm associate with Kalman Filtr and Smoother is proposed. The parameters of the model can be estimated despite an incomplete data.

1.5 Research Methodology

The research firstly developing a spatio-temporal state space model with Kalman filter technique and EM algorithm, then doing series of simulation through the MATLAB software. Secondly the examples of supply chain system, over the area of lower North Region of Thailand, are selected. The data is then collected from these systems. Finally, the collected data is applied into the model. It will be run and analyzed through the MATLAB software in order to generate inventory replenishment policy.

2

Spatio-Temporal State Space

In this chapter the detail of Spatio-Temporal State Space model is presented. Spatio-Temporal State Space model is developed from state space model which use to model the dynamic systems, for example, motor control system, chemical process system.

2.1 Spatio-Temporal State Space Model

In supply chain, the data can be collected at several locations throughout the structure of supply chain system. The overall system can be viewed as a spatio-temporal system since the relationship between each location and the time series data given by each location can be used to model the system. For these reasons, the supply chain system can be viewed as spatio-temporal system and the spatio-temporal based model is an alternative way of modelling supply chain system. In this section, the spatio-temporal state space model is introduced. The spatio-temporal system can be observed as a set of spatially arranged and correlated time series which motivates the use of a spatio-temporally indexed hidden variable, [7], [8]. Let n_t denote the maximum temporal autoregressive order of the process and let $x(s,t)$ be a hidden variable at a specific spatio-temporal location, $s \in \mathcal{S}$ and $t \in \mathcal{T}$ where $\mathcal{S}$ is spatial domain and $\mathcal{T}$ is the temporal domain. The dynamics of the hidden field are represented by

$$x_{t+1} = Ax_t + Ww_t \tag{2.1}$$

where $x_t \in \mathbb{R}^{n_x \times 1}$ denotes the state vector

$$\begin{aligned} x_t \;=\; & [x(s_1,t)\; x(s_2,t)\; \ldots\; x(s_{n_y},t) \\ & x(s_1,t-1)\; x(s_2,t-1)\; \ldots\; x(s_{n_y},t-1) \\ & \vdots \\ & x(s_1,t-n_t+1)\; x(s_2,t-n_t+1)\; \ldots \\ & x(s_{n_y},t-n_t+1)]^T \end{aligned} \tag{2.2}$$

where n_y is the number of observation locations. Partitioning the state vector into current and past hidden variables

$$x_t = \begin{bmatrix} \bar{x}_t \\ \tilde{x}_t \end{bmatrix} \tag{2.3}$$

where the partition

$$\bar{x}_t = [x(s_1, t)\ x(s_2, t) \dots x(s_{n_y}, t)]^T \tag{2.4}$$

and the remainder of the state vector is denoted as $\tilde{x}_t$ allowing the model to be written

$$\bar{x}_t = \bar{A} x_{t-1} + w_t \tag{2.5}$$
$$\tilde{x}_t = [I\ 0] x_{t-1} \tag{2.6}$$

The state matrix is arranged in the following canonical form

$$A = \begin{bmatrix} \bar{A} \\ I\ 0 \end{bmatrix} \tag{2.7}$$

where $\bar{A} \in \mathbb{R}^{n_y \times n_x}$ contains parameters and $I, 0$ denote the identity and zero matrices, respectively, such that $A \in \mathbb{R}^{n_x \times n_x}$. The $n_x \times n_y$ matrix $W = [I\ 0]^T$ maps the state disturbance $w_t \in \mathbb{R}^{n_y \times 1}$ onto the next state. The disturbance on the state is modelled using Gaussian white noise with distribution $w_t \sim \mathcal{N}(0, \Sigma_w)$ where $\Sigma_w \in \mathbb{R}^{n_y \times n_y}$.
The mapping between elements of the hidden field and the observed field is given by

$$y_t = C x_t + v_t \tag{2.8}$$

where $t \in \mathcal{T} \subset Z$ denotes discrete time. The $n_y \times n_x$ observation matrix $C = [I_{n_y \times n_y}\ 0_{n_y \times (n_x - n_y)}]$ is constructed so that the current output is a noise corrupted version of the hidden variables $\bar{x}_t$. The observation vector $y_t \in \mathbb{R}^{n_y \times 1}$ is formed from the current value of the time series associated with each observation location

$$y_t = [y(s_1, t) \dots y(s_{n_y}, t)]^T \tag{2.9}$$

where $s_i \in \mathcal{S}$ is a spatial location. Observation noise is denoted $v_t \in \mathbb{R}^{n_y \times 1}$ and is modelled by Gaussian white noise with distribution $v_t \sim \mathcal{N}(0, \Sigma_v)$ where $\Sigma_v \in \mathbb{R}^{n_y \times n_y}$.
The collection of states up to time n is defined as $X = \{x_1, \dots, x_n\}$, the collection of observation up to time n is defined as $Y = \{y_1, \dots, y_n\}$ and the collection of both the states and the observations is denoted $Z = \{X, Y\}$.

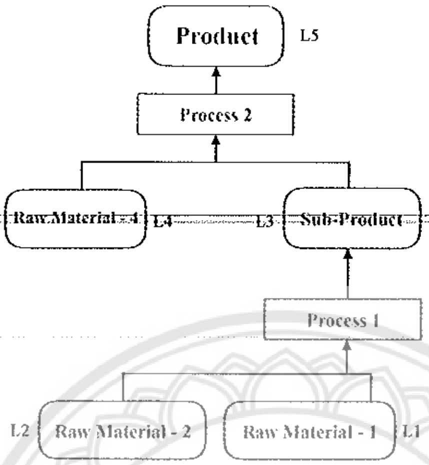

Figure 2.1: Three tiers supply chain system

2.2 Neighbourhood Structure

The relationship between locations in a neighbourhood is introduced into the model via the structure of the $\bar{A}$ matrix. This relationship structure amongst neighbourhoods inside the $\bar{A}$ matrix is then mapped to unknown parameter space, θ, [9].

$$vec(\bar{A}) = \Delta\theta \tag{2.10}$$

The principle of selection of neighbourhood structures is introduced here by configuring the order of the relationship in time and space domains and also the direction. Concerning the directional aspect, there are two directions, one is the forward direction and the other is the backward direction. The direction only applies to the domain of space. Let us consider the spatio-temporal state space model representation of the example supply chain, three tiers supply chain system, as shown in Fig. 2.1

From the example of three tiers supply chain system, the system can be represented using 1 dimension of locations arrangement, as shown in Fig. 2.2

The data are assumed to be correlated with space-time in either direction or both. The order is defined by how far the response from that particular location goes on to affect the neighbourhood in both space and time.

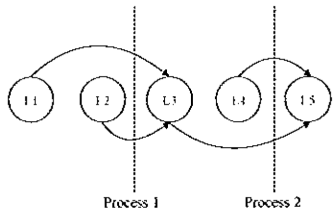

Figure 2.2: Forward flow of materials

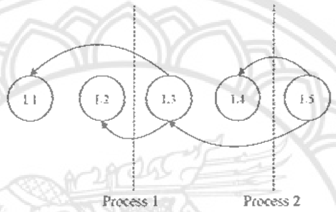

Figure 2.3: Backward flow of information

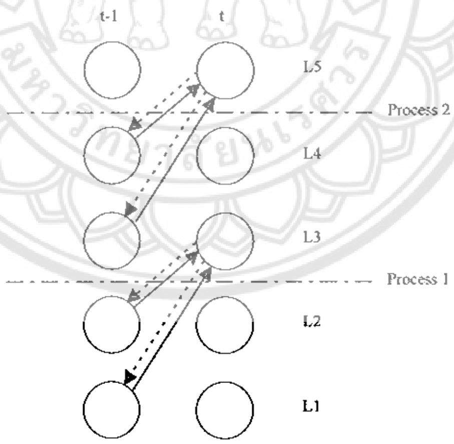

Figure 2.4: Model structure for three tiers supply chain system

The time order means how many steps back in time the response of that particular location has an effect on itself and its neighbourhood.

The configuration shown in Fig. (2.4), corresponds to the structure of the $\bar{A}$ matrix as,

$$\bar{A} = \begin{bmatrix} a_{11} & 0 & a_{13} & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & a_{35} \\ 0 & 0 & 0 & a_{44} & a_{45} \\ 0 & 0 & a_{53} & a_{54} & a_{55} \end{bmatrix} \tag{2.11}$$

The diagonal of matrix $\bar{A}$ corresponds to the autocorrelation effect of the time series at that particular location. The upper part is the result of the forward directional spatial correlation, which represent the information of materials. The lower part is the result of the backward directional spatial correlation, which represent the information of order.

3

Estimation

This chapter deals with the parameter estimation method for Spatio-Temporal State Space model.

3.1 Modelling of STSS Model and EM Algorithm

Considering the spatio-temporal state space model, represented by equation (2.1) and (2.8), the complete data log-likelihood can be written in terms of the model's component densities giving, [[7]],

$$L_\theta(X,Y) = ln\ p(x_0) + \sum_{t=1}^{T} ln\ p(y_t|x_t) + \sum_{t=0}^{T} ln\ p(\bar{x}_{t+1}|x_t) + \sum_{t=0}^{T} ln\ p(\tilde{x}_{t+1}|x_t) \quad (3.1)$$

where the component densities are written as

$$p(x_0) = \mathcal{N}(\mu_0, \Sigma_0),$$
$$p(\bar{x}_t|x_{t-1}) = \mathcal{N}(\bar{A}x_{t-1}, \Sigma_w),$$
$$p(\tilde{x}_t|x_{t-1}) = \delta(\tilde{x}_t - [I\ 0]x_{t-1}),$$
$$p(y_t|x_t) = \mathcal{N}(Cx_t, \Sigma_v).$$

Each term of the model's component densities can be evaluated, ignoring constants. The complete log-likelihood of the spatio-temporal state space model can be written as,

$$L_\theta(X,Y) = -\frac{1}{2}ln|\Sigma_0| - \frac{1}{2}(x_0-\mu_0)^T\Sigma_0^{-1}(x_0-\mu_0) - \frac{n}{2}\ ln|\Sigma_w| - \frac{1}{2}\sum_{t=1}^{n}(\bar{x}_t - \bar{A}x_{t-1})^T\Sigma_w^{-1}(\bar{x}_t - \bar{A}x_{t-1}) - \frac{n}{2}\ ln|\Sigma_v| - \frac{1}{2}\sum_{t=1}^{n}(y_t - Cx_t)^T\Sigma_v^{-1}(y_t - Cx_t) \quad (3.2)$$

3.1.1 E-step

In this step, the expectations for the state sequences are calculated given the current parameter set, θ', and the observation data, Y_n. From the above definition and the definition of the state error covariance equation, the expectations $\bar{\Xi}_{xx}^{(0,1)} \in \mathbb{R}^{n_y \times n_x}$, $\Xi_{xx}^{(1,1)} \in \mathbb{R}^{n_x \times n_x}$ and $\bar{\Xi}_{xx}^{(0,0)} \in \mathbb{R}^{n_y \times n_x}$ are calculated from the following equations,

$$\Xi_{xx}^{(1,1)} = \sum_{t=1}^{n} E_{\theta'}[x_{t-1}^{n} {x_{t-1}^{n}}^{T}] = \sum_{t=1}^{n} (P_{t-1}^{n} + \hat{x}_{t-1}\hat{x}_{t-1}^{T}) \tag{3.3}$$

$$\bar{\Xi}_{xx}^{(0,0)} = \sum_{t=1}^{n} E_{\theta'}[\bar{x}_{t}^{n} {x_{t}^{n}}^{T}] = \sum_{t=1}^{n} (\bar{P}_{t}^{n} + \hat{\bar{x}}_{t}\hat{x}_{t}^{T}) \tag{3.4}$$

$$\bar{\Xi}_{xx}^{(0,1)} = \sum_{t=1}^{n} E_{\theta'}[\bar{x}_{t}^{n} {x_{t-1}^{n}}^{T}] = \sum_{t=1}^{n} (\bar{P}_{t,t-1}^{n} + \hat{\bar{x}}_{t}\hat{x}_{t-1}^{T}) \tag{3.5}$$

where $\bar{P}_{t}^{n}$ denotes the first n_y rows of the smoothed state covariance, $\bar{P}_{t,t-1}^{n}$ denotes the first n_y rows of the lag-one covariance and $\bar{x}_{t}^{n}$ denotes the first n_y rows of smoothed state sequence.

3.1.2 M-step

After evaluating the expectations, the Q-function becomes a deterministic function of θ, which can be maximised, given an estimated state sequence. The Q-function for the spatio-temporal system represented in equations (2.5), (2.6) and (2.8) can be written in the following form,

$$Q(\theta^{(i)}|\theta^{(i-1)}) = E\{2\ L_{\theta}(X,Y)|Y_n, \theta^{(i-1)}\} \tag{3.6}$$

where the complete log-likelihood, $L_{\theta}(X,Y)$, is presented in equation (3.2). In order to express the log-likelihood in a more compact form, the trace operator is applied. Replacing equation (3.2) into (3.6), the Q function in equation (3.6) can be rewritten as

$$\begin{aligned} Q(\theta^{(i)}|\theta^{(i-1)}) \ = \ & -ln|\Sigma_0| - tr\{\Sigma_0^{-1}[P_0^n + (x_0^n - \mu_0)(x_0^n - \mu_0)^T]\} \\ & -n\ ln|\Sigma_w| \\ & -tr\{\Sigma_w^{-1}[\bar{\Xi}_{xx}^{(0,0)} - \bar{\Xi}_{xx}^{(0,1)}\bar{A}^T - \bar{A}(\bar{\Xi}_{xx}^{(0,1)})^T + \bar{A}\Xi_{xx}^{(1,1)}\bar{A}^T]\} \\ & -n\ ln|\Sigma_v| \\ & -tr\{\Sigma_v^{-1}\sum_{t=1}^{n}[(y_t - Cx_t^n)(y_t - Cx_t^n)^T + C_t P_t^n C_t^T]\} \end{aligned} \tag{3.7}$$

where the expectations $\Xi_{xx}^{(0,1)} \in \mathbb{R}^{n_x \times n_x}$, $\Xi_{xx}^{(1,1)} \in \mathbb{R}^{n_x \times n_x}$ and $\Xi_{xx}^{(0,0)} \in \mathbb{R}^{n_x \times n_x}$ are calculated by the following equations,

$$\Xi_{xx}^{(1,1)} = \sum_{t=1}^{n} E_{\theta'}[x_{t-1}^{n} {x_{t-1}^{n}}^{T}] = \sum_{t=1}^{n} (P_{t-1}^{n} + \hat{x}_{t-1}\hat{x}_{t-1}^{T}) \tag{3.8}$$

$$\Xi_{xx}^{(0,0)} = \sum_{t=1}^{n} E_{\theta'}[x_t^n x_t^{nT}] = \sum_{t=1}^{n}(P_t^n + \hat{x}_t\hat{x}_t^T) \tag{3.9}$$

$$\Xi_{xx}^{(0,1)} = \sum_{t=1}^{n} E_{\theta'}[x_t^n x_{t-1}^{n}{}^T] = \sum_{t=1}^{n}(P_{t,t-1}^n + \hat{x}_t\hat{x}_{t-1}^T) \tag{3.10}$$

where $\hat{x}$ denotes the expects value of x.

From the basic trace operator theorem, $tr(A) = tr(A^T)$ and the property of transpose $(AB)^T = B^T A^T$, it can be shown that

$$\begin{aligned} tr(\bar{\Xi}_{xx}^{(0,1)}\bar{A}^T) &= tr((\bar{\Xi}_{xx}^{(0,1)}\bar{A}^T)^T) \\ &= tr(\bar{A}(\bar{\Xi}_{xx}^{(0,1)})^T) \end{aligned}$$

Replacing $tr\{\bar{\Xi}_{xx}^{(0,1)}\bar{A}^T\}$ with $tr\{\bar{A}(\bar{\Xi}_{xx}^{(0,1)})^T\}$ in equation (3.7) then leads to

$$\begin{aligned} Q(\theta^{(i)}|\theta^{(i-1)}) &= -ln|\Sigma_0| - tr\{\Sigma_0^{-1}[P_0^n + (x_0^n - \mu_0)(x_0^n - \mu_0)^T]\} \\ &\quad -n\ ln|\Sigma_w| - tr\{\Sigma_w^{-1}[\bar{\Xi}_{xx}^{(0,0)} - 2\bar{A}(\bar{\Xi}_{xx}^{(0,1)})^T + \bar{A}\Xi_{xx}^{(1,1)}\bar{A}^T]\} \\ &\quad -n\ ln|\Sigma_v| \\ &\quad -tr\{\Sigma_v^{-1}\sum_{t=1}^{n}[(y_t - Cx_t^n)(y_t - Cx_t^n)^T + C_t P_t^n C_t^T]\} \end{aligned} \tag{3.11}$$

Assume that the observation matrix(C), observation noise covariance(Σ_v), state uncertainty covariance(Σ_w) and neighbourhood structure are known. The unknown parameter set is then defined by $\theta = \{\bar{A}\}$. Replacing the complete log-likelihood equation into the Q function equation gives

$$f_Q(\bar{A}) = E[\alpha - \sum_{t=1}^{T}(\bar{x}_t - \bar{A}x_{t-1})^T\Sigma_w^{-1}(\bar{x}_t - \bar{A}x_{t-1})] \tag{3.12}$$

where the quantities that do not depend on an unknown parameter, θ, are lumped into the constant α. Based on the properties of the trace operator and the definition of expectations, equation (3.12) can be written as,

$$\begin{aligned} f_Q(\bar{A}) &= \alpha - tr\{\Sigma_w^{-1}[-2\bar{A}(\bar{\Xi}_{xx}^{(0,1)})^T + \bar{A}\Xi_{xx}^{(1,1)}\bar{A}^T]\} \\ &= \alpha + 2tr\{(\bar{\Xi}_{xx}^{(0,1)})^T\Sigma_w^{-1}\bar{A}\} - tr\{\Xi_{xx}^{(1,1)}\bar{A}^T\Sigma_w^{-1}\bar{A}\} \end{aligned} \tag{3.13}$$

Further properties of the Kronecker product, the vectorise and trace operators are used to manipulate equation (3.13), [[?]]. The first component of the equation can be written as

$$tr((\bar{\Xi}_{xx}^{(0,1)})^T\Sigma_w^{-1}\bar{A}) = vec(I)^T(\bar{\Xi}_{xx}^{(0,1)} \otimes \Sigma_w^{-1})vec(\bar{A}) \tag{3.14}$$

and the second component can be written as

$$tr(\bar{\Xi}_{xx}^{(1,1)}\bar{A}^T\Sigma_w^{-1}\bar{A}) = vec(\bar{A})^T(\bar{\Xi}_{xx}^{(0,1)} \otimes \Sigma_w^{-1})vec(\bar{A}) \tag{3.15}$$

replace the equation (3.14) and (3.15) into equation (3.13)

$$\begin{aligned} f_Q(\bar{A}) &= \alpha + 2vec(I)^T(\bar{\Xi}_{xx}^{(0,1)} \otimes \Sigma_w^{-1})vec(\bar{A}) \\ &\quad -vec(\bar{A})^T(\Xi_{xx}^{(1,1)} \otimes \Sigma_w^{-1})vec(\bar{A}) \end{aligned} \tag{3.16}$$

where $\otimes$ denotes the Kronecker product, see Appendix (??) for more details, and all the independent parameters are lumped into the constant α. Replacing the term $\bar{A}$ with $vec(\bar{A}) = \Delta\theta$, gives

$$\begin{aligned} f_Q(\bar{A}) &= \alpha + 2vec(I)^T(\bar{\Xi}_{xx}^{(0,1)} \otimes \Sigma_w^{-1})\Delta\theta \\ &\quad -(\Delta\theta)^T(\Xi_{xx}^{(1,1)} \otimes \Sigma_w^{-1})\Delta\theta \end{aligned} \tag{3.17}$$

In order to maximise the function, differentiating the $f_Q(\bar{A})$ function with respect to θ gives

$$\frac{d\ f_Q(\bar{A})}{d\theta} = 2vec(I)^T(\bar{\Xi}_{xx}^{(0,1)} \otimes \Sigma_w^{-1})\Delta - 2\theta^T\Delta^T(\Xi_{xx}^{(1,1)} \otimes \Sigma_w^{-1})\Delta \tag{3.18}$$

With the above equation being set to zero, the estimate of the unknown parameter $\hat{\theta} = \{\bar{A}\}$ that locally maximises the Q-function of the spatio-temporal state space model is given by

$$\hat{\theta} = (\Delta^T(\Xi_{xx}^{(1,1)} \otimes \Sigma_w^{-1})\Delta)^{-1}\Delta^T((\bar{\Xi}_{xx}^{(0,1)})^T \otimes \Sigma_w^{-1})vec(I) \tag{3.19}$$

Note that the matrix $(\Delta^T(\Xi_{xx}^{(1,1)} \otimes \Sigma_w^{-1})\Delta)$ is invertible.

In some applications, the state uncertainty covariance and observation noise covariance are assumed unknown. The estimation of state uncertainty covariance can be performed by maximising the log-likelihood function with respect to state uncertainty covariance matrix, given the Q function of the form,

$$\begin{aligned} Q(\theta^{(i)}, \theta^{(i-1)}) &= \alpha - n\ ln|\Sigma_w| \\ &\quad -tr\{\Sigma_w^{-1}[\bar{\Xi}_{xx}^{(0,0)} - \bar{\Xi}_{xx}^{(0,1)}\bar{A}^T - \bar{A}(\bar{\Xi}_{xx}^{(0,1)})^T \\ &\quad +\bar{A}\Xi_{xx}^{(1,1)}\bar{A}^T]\} \end{aligned} \tag{3.20}$$

where α gathers all the terms that do not depend on state uncertainty covariance.

3.2 EM Algorithm

Given an incomplete data set, the estimation of dynamic systems can be done using the EM (Expectation-Maximisation) algorithm which first introduced by [1]. Parameter estimation problem for state space models which is solved by EM algorithm was presented by [11]. The EM algorithm in conjunction with the conventional Kalman smoothed estimators are used to estimate the parameters of the state space model by maximum likelihood. The EM algorithm for spatio-temporal state space, [9], is presented in Table. 3.1

Table 3.1: The EM algorithm for state space models.

Initial:
state sequence, x_0
parameter set $\theta^{(i)}$

EM algorithm:
E-step:
The calculation of estimated state sequence, given the observed data:
Kalman filter: Prediction
Calculation of a priori state estimate $\hat{x}_t^{t-1}$
Calculation of the prior state error covariance P_t^{t-1}

Kalman filter: Correction
Calculation of a posteriori state estimate $\hat{x}_t^t$
Calculation of the posterior state error covariance P_t^t

RTS smoother:
Calculation of the covariance of the smoothed state P_t^n

Lag-one covariance smoother:
Calculation of the lag one covariance of the smoothed state $P_{t,t-1}^n$

The calculation of the expectation of state sequence:
$\Xi_{xx}^{(1,1)}$, $\Xi_{xx}^{(0,0)}$, $\Xi_{xx}^{(0,1)}$

M-step:
New parameter set , $\theta^{(i+1)}$, is calculated by maximising the Q-function

Repeating until the stopping criteria is reached

4

Spatio-Temporal State Space for Supply Chain System

In this section, the Spatio-Temporal State Space model for supply chain system is discussed in detail. Two simulated example of simple supply chain system are explored and analyse. Finally, the spatio-temporal state Space model is utilised based on collected data.

4.1 Spatio-Temporal State Space for supply chain system

The concept of dynamic programming has been use in formulating, analysing and solving the inventory problem. For a single-echelon, single product sytem where the problem is to optimally select orders $u(t)$ of the product (control variable) in order to meet uncertain demand $d(t)$ (exogeneous disturbance), while minimising the total expected purchasing, inventory and shortage cost. The dynamics of the system are described by the following stat space equation[6]:

$$x(t+1) = x(t) + u(t) - d(t) \tag{4.1}$$

where $x(t)$ is the inventory level at time t^{th}. In this study, based on state space approach for supply chain system, space domain is considered and included into the mathematical model. The proposed model capture the dynamic of supply chain system in both time and space domain.

Consider supply chain system example, as shown in Fig. 2.4, which can be represented using spatio-temporal state space model as

$$x_i(t) = Ax_i(t-1) + Bu_i(t) - D_d d(t) \tag{4.2}$$

where state matrix (A), which capture the dynamic of the system, and state vertor $(x_i(t))$ at location i is arranged in the form of eq. (2.7) and (2.3)

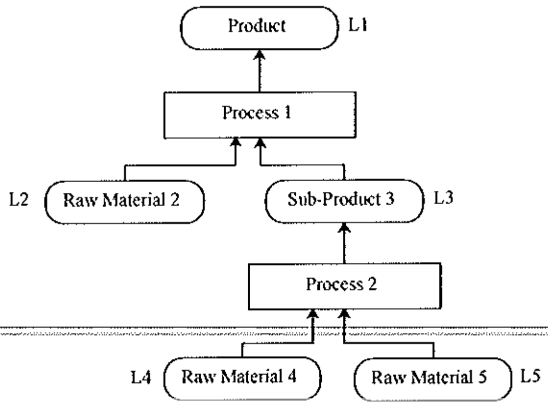

Figure 4.1: Supply Chain System Example 1

respectively, $u_i(t) = [u_1(t,\tau_1), u_2(t,\tau_2), u_3(t,\tau_3), u_4(t,\tau_4), u_5(t,\tau_5)]^T$ is the order plan at time t at each locations, τ is defined as a delayed and $d(t)$ is the demand of the product at time t. In this study, time delay is not taken into account. From the matrix $\bar{A}$ in equation 2.11, neighbourhood structure formulation, the A matrix can be written as follow,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 \\ a_{31} & 0 & a_{33} & a_{34} & a_{35} \\ 0 & 0 & a_{43} & a_{44} & 0 \\ 0 & 0 & a_{53} & 0 & a_{55} \end{bmatrix} \tag{4.3}$$

The matrix B and D_d is shown as follow, respectively,

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.4}$$

$$D_d = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.5}$$

Considering at each location the spatio-temporal state space equations are presented as follow, respectively,

$$x_1(t) = a_{11}x_1(t-1) + a_{12}x_2(t-1) + a_{13}x_3(t-1) + u_1(t) - d_1(t) \quad (4.6)$$

$$x_2(t) = a_{21}x_1(t-1) + a_{22}x_2(t-1) + u_1(t) \quad (4.7)$$

$$x_3(t) = a_{31}x_1(t-1) + a_{33}x_3(t-1) + a_{34}x_4(t-1) + a_{35}x_5(t-1) + u_3(t) \quad (4.8)$$

$$x_4(t) = a_{43}x_3(t-1) + a_{44}x_4(t-1) + a_{45}x_5(t-1) + u_3(t) \quad (4.9)$$

$$x_5(t) = a_{53}x_3(t-1) + a_{55}x_5(t-1) + u_3(t) \quad (4.10)$$

The measurements, with noise, available at location 1 and 3. The mapping between elements of the hidden field and the observed field is given by

$$y_i(t+1) = Cx_i(t) + v(t) \quad (4.11)$$

where matrix C is shown as follow,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.12)$$

As a results,

$$y_1(t) = x_1(t-1) + v_1(t) \quad (4.13)$$

$$y_3(t) = x_3(t-1) + v_3(t) \quad (4.14)$$

Another example, the supply chain system with more complex structure as shown in figure 4.2. A matrix, with the consideration of space and time, can be written as follow,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 & 0 \\ a_{31} & 0 & a_{33} & a_{34} & a_{35} & a_{36} \\ 0 & 0 & a_{43} & a_{44} & 0 & a_{46} \\ 0 & 0 & a_{53} & 0 & a_{55} & a_{56} \\ 0 & 0 & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix} \quad (4.15)$$

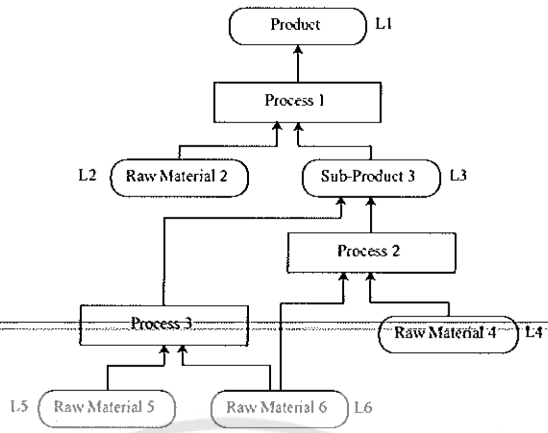

Figure 4.2: Supply Chain System Example 2

Considering at each location the spatio-temporal state space equations are presented as follow, respectively,

$$x_1(t) = a_{11}x_1(t-1) + a_{12}x_2(t-1) + a_{13}x_3(t-1) + u_1(t) - d_1(t) \quad (4.16)$$

$$x_2(t) = a_{21}x_1(t-1) + a_{22}x_2(t-1) + u_1(t) \quad (4.17)$$

$$x_3(t) = a_{31}x_1(t-1)+a_{33}x_3(t-1)+a_{34}x_4(t-1)+a_{35}x_5(t-1)+a_{36}x_6(t-1)+u_3(t) \quad (4.18)$$

$$x_4(t) = a_{43}x_3(t-1)+a_{44}x_4(t-1)+a_{45}x_5(t-1)+a_{46}x_6(t-1)+u_3(t) \quad (4.19)$$

$$x_5(t) = a_{53}x_3(t-1) + a_{55}x_5(t-1) + a_{56}x_6(t-1) + u_3(t) \quad (4.20)$$

$$x_6(t) = a_{63}x_3(t-1)+a_{64}x_4(t-1)+a_{65}x_5(t-1)+a_{66}x_6(t-1)+u_3(t) \quad (4.21)$$

The measurements, with noise, available at location 1 and 3. The mapping between elements of the hidden field and the observed field is given by

$$y_i(t+1) = Cx_i(t) + v(t) \quad (4.22)$$

where matrix C is shown as follow,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.23}$$

As a results,

$$y_1(t) = x_1(t-1) + v_1(t) \tag{4.24}$$

$$y_3(t) = x_3(t-1) + v_3(t) \tag{4.25}$$

4.2 Simulation

The simulation was done using MATLAB software, source code available in Appendix A. There are two sections in simulation step. First, the simulation of toy model, in this section the toy model which mimic the supply chain system is used to simulate the ability of proposed technique. Secondly, the collected data from a case study is used to simulate the performance of proposed technique.

4.2.1 A toy Model, Supply Chain System Example 1

In this section, a duplicated supply chain system is generated to mimic the behaviour of the real system, a toy model. Measurements of the quantity of the product are measured in location L_3 and L_5, as shown in Figure 4.3 and 4.4 respectively. Measurement at location L_3 is generated using $\mathcal{N}(50, 3)$ with white noise and location L_3 is generated using $\mathcal{N}(50, 2)$ with white noise.

Inventory levels at location L_2, L_4, L_5 are input of the system, shown in figure 4.5,

Parameters in A matrix, estimated A matrix, calculation is done by using the algorithm in section 3. Q function is shown in figure 4.6 and estimated A matrix is shown in figure 4.7.

4.2.2 A toy Model, Supply Chain System Example 2

In this section, a duplicated supply chain system is generated to mimic the behaviour of the real system, a toy model. Measurements of the quantity of the product are measured in location L_3 and L_5, as shown in Figure 4.8 and 4.9 respectively. Measurement at location L_3 is generated using $\mathcal{N}(140, 4)$

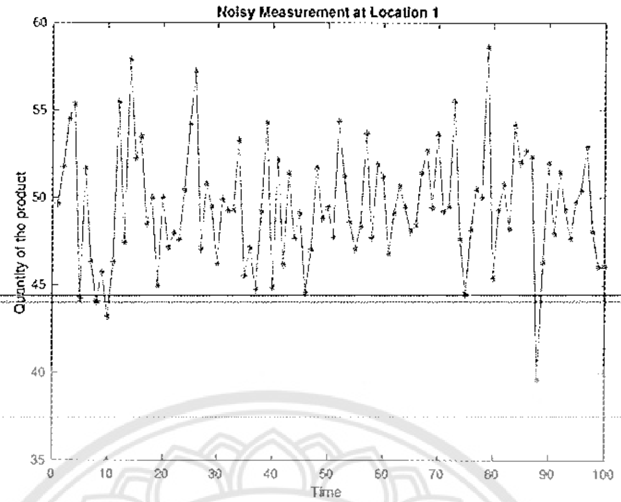

Figure 4.3: Quantity of the product at location L_1, Example 1.

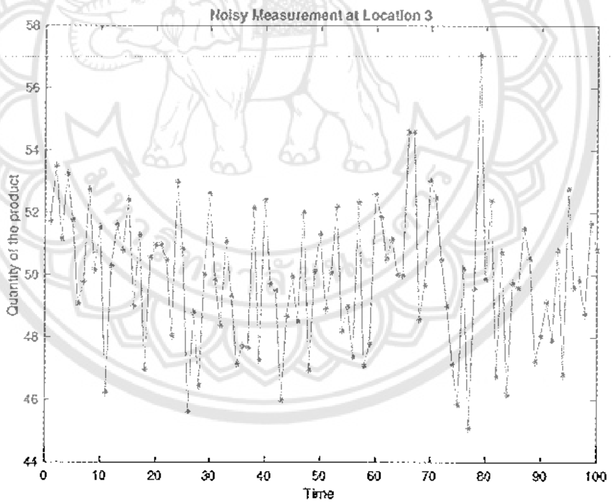

Figure 4.4: Quantity of the product at location L_3, Example 1.

with white noise and location L_3 is generated using $\mathcal{N}(160, 2)$ with white noise.

Inventory levels at location L_2, L_4, L_5 are input of the system, shown in figure 4.10,

Parameters in A matrix, estimated A matrix, calculation is done by using

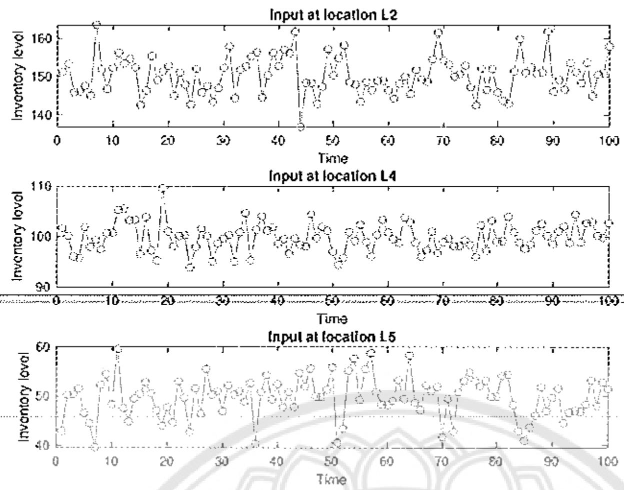

Figure 4.5: Input, inventory level of supply chain system, Example 1.

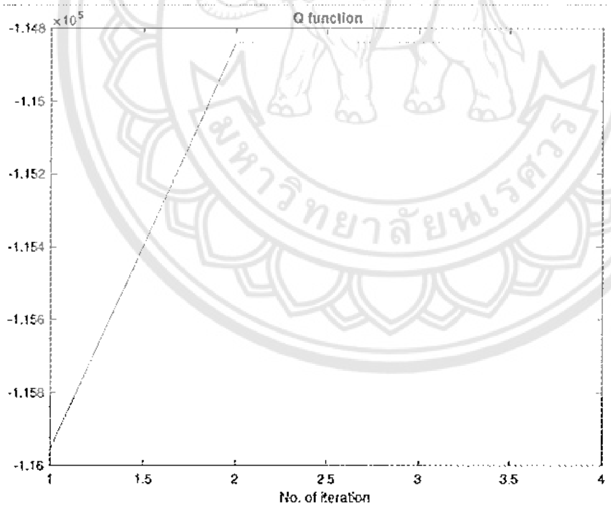

Figure 4.6: Q function, Example 1.

the algorithm in section 3. Q function is shown in figure 4.11 and estimated A matrix is shown in figure 4.12.

```
estimated A (EM algorithm)
    0.7854         0    0.1811         0         0
         0    0.6030    0.3580         0         0
    0.1198    0.1126   -0.0081         0    0.8372
         0         0         0   -0.0193    0.8956
         0         0    0.0345    0.1278    0.8516

>> |
```

Figure 4.7: Estimated A Matrix, Example 1.

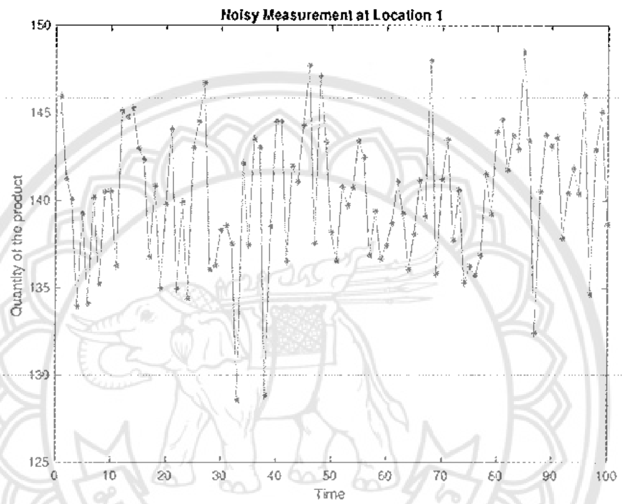

Figure 4.8: Quantity of the product at location L_1, Example 2.

4.2.3 A case study

In this section, data collections are done within the selected electronic part assembly company. Selected electronic part, a final product, are comprised of 3 raw materials and 1 sub-part. Inventory level for each material is collected and use as an input of the system. Number of part both final product and sub-part are recorded and use as a measurement of the system. All collected data can be shown in figure 4.13, 4.14 and 4.15.

Inventory levels at location L_2, L_4, L_5 are input of the system, shown in figure 4.10,

Parameters in A matrix, estimated A matrix, calculation is done by using the algorithm in section 3. The result is shown in figure 4.16.

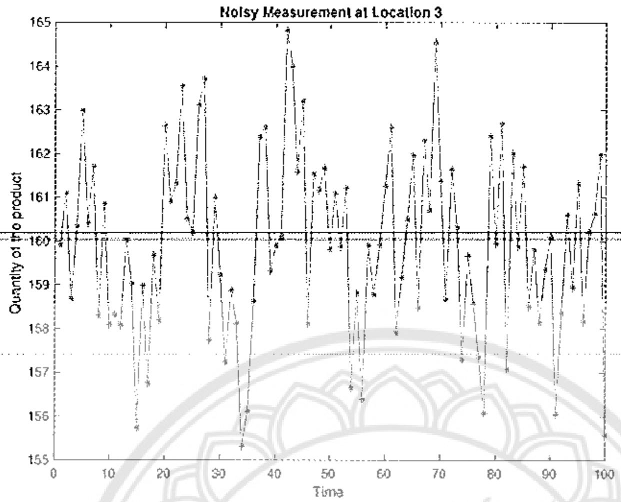

Figure 4.9: Quantity of the product at location L_3, Example 2.

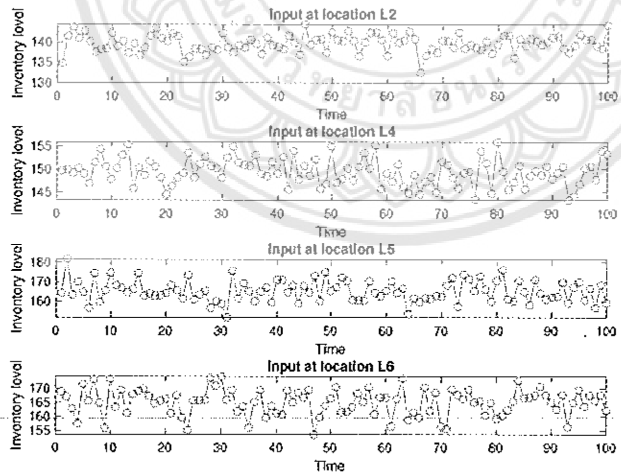

Figure 4.10: Input, inventory level of supply chain system, Example 2.

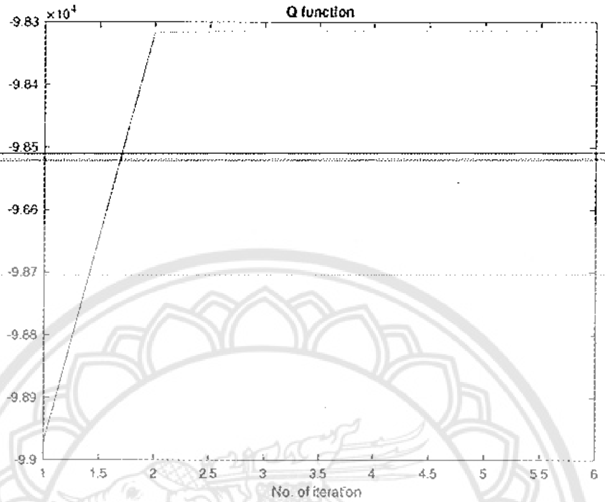

Figure 4.11: Q function, Example 2.

```
estimated A (EM algorithm)
   -0.0012   -0.1170    0.9792         0         0         0
    0.2368    0.7628         0         0         0         0
   -0.0467         0    1.0228    0.0833   -0.1443    0.0862
         0         0    0.6976    0.0297         0    0.2022
         0         0    1.2030         0   -0.1693    0.0047
         0         0    0.8103    0.2546   -0.1902    0.1758

>>
```

Figure 4.12: Estimated A Matrix, Example 2.

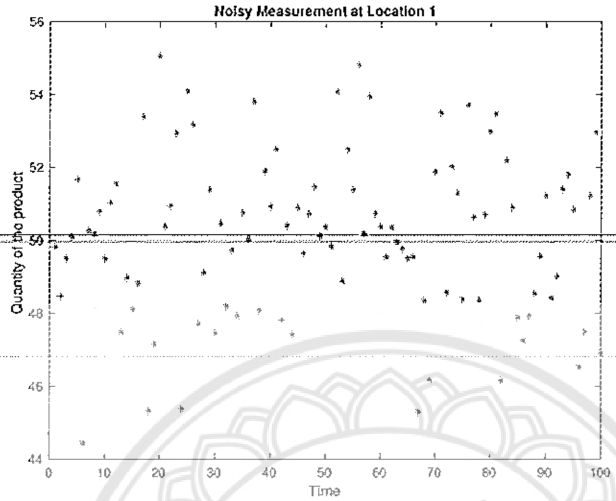

Figure 4.13: Quantity of the product at location L_1

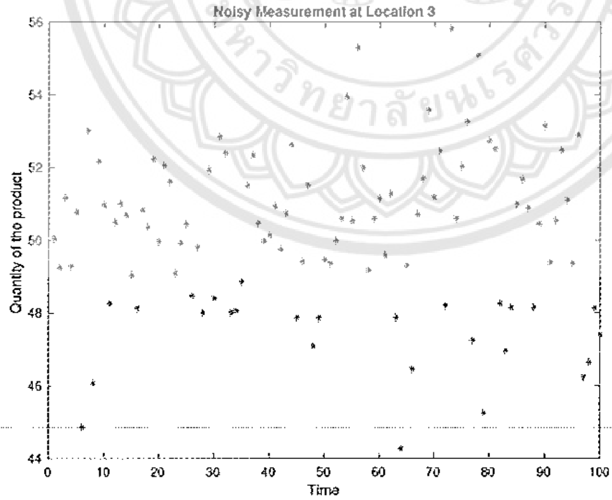

Figure 4.14: Quantity of the product at location L_3

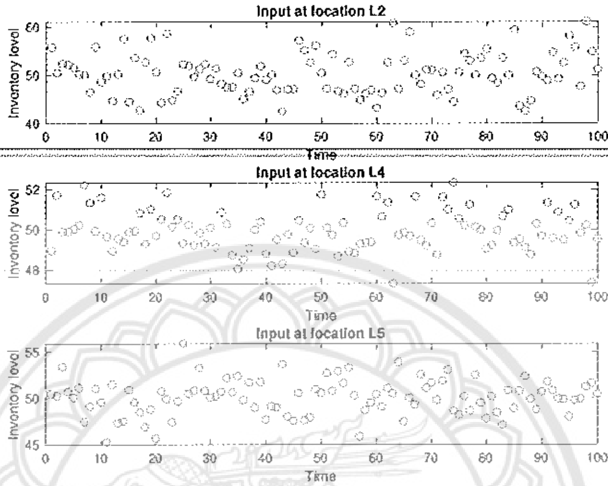

Figure 4.15: Input, inventory level of supply chain system

```
estimated A (EM algorithm)
    0.5572         0    0.4405         0         0
         0    0.4274    0.5715         0         0
    0.4819    0.0980    0.1728         0    0.2482
         0         0         0    0.9326    0.0671
         0         0    0.0884    0.7494    0.1639

>>
```

Figure 4.16: Estimated A Matrix

5

Conclusion

The spatio-temporal state space model is proposed as a dynamic model of supply chain system since the basic structure is match. The information at each location can be integrated into a model via the structure of the spatio-temporal state space model. The model structure integrates flow of information in one direction and material flow in another direction. These can be represented in upper section and lower section of A matrix. In the diagonal of A matrix, the relationship between inventory level at previous time step and current situation is captured. The EM algorithm associate with Kalman Filter use to estimate the parameter in A matrix. The results show that the algorithm able to estimated parameters even the data, inventory level and/or quantity of parts, is contaminated with noise.

Recommendations

In this study, the relationship of inventory level and quantity of part and/or sub-part are considered. In estimation step, the estimated A matrix can become unstable. This problem arise time by time. The further study can explore, how to guarantee the stability of A matrix during estimation step. Another issue, since the process time associated with time delay is not considering in this study. Further explore area is that, how can we integrate such kind of information into spatio-temporal state space model? To completely capture the dynamic of supply chain system all information and uncertainty have to be captured, analyse and able to use those information in order to control the system. The mathematic model that can mimic the real system can be used as a power tool for supply chain management.

Bibliography

[1] A. P. Dempster, N.M. Laird, and D.B. Rubin. *Maximum likelihood from incom- plete data via the EM algorithm*, Journal of the Royal Statistical Society. Series B (Methodology), Vol. 39, No. 1, 1977

[2] Benita M. Beamon, *Supply chain design and analysis: Models and methods*, Internation Journal of Production Economics, Vol. 55, No. 3, 1998, pp. 281-294

[3] C. Papanagnou and G. D. Halikias, *Supply chain modeling and control under proportional inventory-replenishment policies*, International Journal of Systems Science, Vol. 39, No. 7, 2008, pp. 699-711

[4] C. E. Riddalls, S. Bennett and N. S. Tipi, *Modelling the dynamics of supply chains*, International Journal of Systems Science, Vol. 31 No. 8, 2000, pp. 969-976

[5] H. L. Lee, V. Padmanabhan and S. Whang, *The bullwhip effect in supply chains*, Sloan Management Review, Vol. 38, No. 3, 1997, pp. 93-102

[6] H. Sarimveis, P. Patrinos, C. D. Tarantilis and C. T. Kiranoudis, *Dynamic modeling and control of supply chain systems: A review*, Computers & Operations Research, Vol. 35, 2008, pp. 3530-3561

[7] Michael A. Dewar, *A framework for dynamic modelling of spatiotemporal systems*, PhD thesis, The University of Sheffield, 2007.

[8] Michael A. Dewar and Visakan Kadirkamanathan, *A canonical space-time state space model: State and parameter estimation*, IEEE Transactions on Signal Processing, Vol. 55 , No. 10, 2007, pp. 4862-4870

[9] Phisut Apichayakul, *Spatio-temporal State Space Model Estimation for Smart Structures*, PhD thesis, The University of Sheffield, 2010.

[10] Phisut Apichayakul and Visakan Kadirkamanathan, *Spatio-temporal dynamic modeling of smart structures using a robust expectation-maximization algorithm*, Smart Materials and Structures, Vol. 20, No.4, 2011

[11] Robert H. Shumway and David S. Stoffer *An approach to time series smoothing and forecasting using the EM algorithm* Journal of Time Series Analysis, Vol. 3, No. 4, 1982, pp. 253-264

[12] Robert H. Shumway and David S. Stoffer, *Time Series Analysis and Its Applications with R Examples*, Springer, 2006.

[13] Sunil Chopra and Peter Meindl *Supply Chain Management, Strategy, Planning&Operations, 3rd Edition*, Pearson, 2007.

[14] Suresh P. Sethi, Houmin Yan and Hanqin Zhang, *Inventory and Supply Chain Management with Forecast Updates*, Springer, 2005.

[15] T. Wu and P. O'Grady, *An extended Kalman filter for collaborative supply chain management*, Internet Lab Technical Report TR 2001-11, 2001.

[16] T. Wu and P. O'Grady, *A network-based approach to integrated supply chain design*, Production Planning & Control: The Management of Operations, Vol. 16, No. 5, 2005, pp. 444-453

[17] Yossi Aviv, *A time-series framework for supply chain inventory management*, Operations Research, Vol. 51, No. 2, 2003, pp. 210-227

Appendix A

MATLAB Code

stsssc1.m

```
clear all
close all
clc

resY=realdd;
resYY=outpp(resY);
%load resYYf
%ploty(resY);

icomp=1;

mystss=stss(resYY);
mystss=testmodelr(mystss,icomp);

%mystss.A

%EM algorithm
[nstored mystssem]=emc(mystss);

disp('estimated A (EM algorithm)')
disp(mystssem.A)

%mystssem
%finalsw=mystssem.Sw
%finalsv=mystssem.Sv

yr=mystssem.Y;
```

emc.m

```
function [lstore, sys]=emc(sys)

% function to calculate the maximum likelihood parameter estimates of the
% state space model object sys. Returns the sys object with updated state
% and parameter estimates.
%
% sys is expected to have at least nx and Y defined. Will overwrite
% anything else.

% store initial system for debugging
sysinit=sys;

lf(1)=0;
qf(1)=0;
lold=0;

threshold=1*10^(-5);

% iterate
it=1;
count=1;
criteria=1;

%Initial state sequence and its covariance
P0f = sys.P;
P0=P0f(:,:,1);
P0ini=P0;

%put measurement into x0
%yx=sys.Y/(10^(3));
%xhat0=[yx(:,1); yx(:,2)];

%random generation x0
xhat0=randn(sys.nx,1);   %X0 - N(0,1)

xhat0ini=xhat0;

while criteria

   %it
   Abef=sys.A;
   %swt=eig(sys.Sw)
   %svt=eig(sys.Sv)

   % E step
   [sys,lfilter]=ksmooth(sys,P0,xhat0);
```

```
    lincom(it+1)=lfilter;
    if lincom(it+1) < lincom(it), warning(['negative change in l: ' num2str(lincom(it+1)-lincom(it))]),
end


    %update P0 x0 for next iteration.
    %P0f = sys.P;
    %P0=P0f(:,:,1);
    %xhat0f = sys.X;
    %xhat0 = xhat0f(:,1);

    % M step
    [sys,qfunc]=maximise(sys,P0,xhat0,P0ini,0);

    qf(it+1)=qfunc;
    if qf(it+1) < qf(it), warning(['negative change in q function: ' num2str(qf(it+1)-qf(it))]), end

    Aaft=sys.A;

    %figure(3)
    %hold on
    %plot(it,qfunc,'*')

    l=complikelihood(sys,xhat0ini,P0ini,xhat0ini);
    lstore(count)=l;
    if lold > l, warning(['negative change in comp likelihood: ' num2str(l-lold)]), end
    lold=l;

    cunstable=max(abs(eig(sys.A)));
    if cunstable>1
       error('unstable A matrix')
       %estimated state lead to unstable A matrix
    end

    count=count+1;

    normbf=norm(Abef);
    normaf=norm(Aaft);
    ccheck=abs(normbf-normaf);
    %Abef-Aaft

    %criteria=abs(l-lold)>threshold;
    %criteria=abs(lf(it+1) - lf(it))>threshold;
    %criteria=abs(qf(it+1) - qf(it))>threshold;
    criteria=ccheck>threshold;

    it=it+1;
```

```
    % put a limit on the number of iterations
    if it>50, criteria=0; end

end

%plotres(qf,lstore,lincom);
%plotql(qf,lstore);
end

function plotql(qf,lstore)

figure
for i=2:length(qf)
    qfp(i-1)=(qf(i));
end
plot(qfp)
title('Q function')
xlabel('No. of iteration')
figure
for i=2:length(lstore)
    lcp(i-1)=(lstore(i));
end
plot(lcp)
title('The complete log-likelihood function')
xlabel('No. of iteration')
end

function plotres(qf,lstore,lincom)

figure(5)
subplot(3,1,1)
for i=2:length(qf)
    qfp(i-1)=(qf(i));
end
plot(qfp)
title('Q function')

subplot(3,1,2)
for i=2:length(lstore)
    lcp(i-1)=(lstore(i));
end
plot(lcp)
title('L complete')

subplot(3,1,3)
for i=2:length(lincom)
    lfp(i-1)=(lincom(i));
```

```
end
plot(lfp)
title('L')

save ldata.mat lstore;
end
```

testmodelr.m

```
function sys=testmodelr(toy,icomp)


%% set up the testing model
swt=1*10^(-1);
svt=1*10^(-3);

%icomp
switch icomp

   case 1

%nt=input('Enter nt :=');
nt=1;
ny=toy.ny;
nx=nt*ny;

%test=stss(nx,ny);
test=stss(toy.Y);
test.nx=nx;

%test.Y=toy.Y;
test.U=zeros(0,toy.T);
test.B=toy.B;

 %test.A=afixbeamb(icomp,nt,ny);
 test.A=[1 0 1 0 0;0 1 1 0 0;1 1 1 0 1;0 0 0 1 1;0 0 1 1 1];

 test=mappingd(test);
 test.A=[];
 test.P=1000*eye(nx);
 %test.Sw=1*10^(-1)*eye(ny);
 %test.Sv=2*10^(-4)*eye(ny);
 test.Sw=swt*eye(ny);
 test.Sv=svt*eye(ny);
 test.W=[eye(ny,ny); zeros(nx-ny,ny)];
 test.C=[eye(ny) zeros(ny,nx-ny)];

 %test.C=[0 0 0 0 0;0 0 0 0 0;0 0 0 0 0;0 0 0 0 0;0 0 0 0 1];
 test=myinitialise(test);


   %test.A=[ab ; eye(nx-ny) zeros(nx-ny,ny)];
   Astab=test.A;

   disp(['max eig of test model: ' num2str(max(abs(eig(Astab))))])
```

```
nx=nt*ny;

%test=stss(nx,ny);
test=stss(toy.Y);
test.nx=nx;

%test.Y=toy.Y;
test.U=zeros(0,toy.T);
test.B=toy.B;

 test.A=afixbeamb(icomp,nt,ny);
 test=mappingd(test);
 test.A=[];
 test.P=1000*eye(nx);
 %test.Sw=2*10^(-2)*eye(ny);
 %test.Sv=8*10^(-3)*eye(ny);
 test.Sw=swt*eye(ny);
 test.Sv=svt*eye(ny);
 test.W=[eye(ny,ny); zeros(nx-ny,ny)];
 test.C=[eye(ny) zeros(ny,nx-ny)];

 test=myinitialise(test);


   %test.A=[ab ; eye(nx-ny) zeros(nx-ny,ny)];
   Astab=test.A;

   disp(['max eig of test model: ' num2str(max(abs(eig(Astab))))])

   while max(abs(eig(Astab))) > 1
     error('unstable ini system')
   end

   case 4

%nt=input('Enter nt :=');
nt=3;
ny=toy.ny;
nx=nt*ny;

%test=stss(nx,ny);
test=stss(toy.Y);
test.nx=nx;

%test.Y=toy.Y;
test.U=zeros(0,toy.T);
test.B=toy.B;
```

```
test.A=afixbeamb(icomp,nt,ny);
test=mappingd(test);
test.A=[];
test.P=1000*eye(nx);
%test.Sw=2*10^(-2)*eye(ny);
%test.Sv=8*10^(-3)*eye(ny);
test.Sw=swt*eye(ny);
test.Sv=svt*eye(ny);
test.W=[eye(ny,ny); zeros(nx-ny,ny)];
test.C=[eye(ny) zeros(ny,nx-ny)];

test=myinitialise(test);

  %test.A=[ab ; eye(nx-ny) zeros(nx-ny,ny)];
  Astab=test.A;

  disp(['max eig of test model: ' num2str(max(abs(eig(Astab))))])

  while max(abs(eig(Astab))) > 1
    error('unstable ini system')
  end

  case 5

%nt=input('Enter nt :=');
nt=4;
ny=toy.ny;
nx=nt*ny;

%test=stss(nx,ny);
test=stss(toy.Y);
test.nx=nx;

%test.Y=toy.Y;
test.U=zeros(0,toy.T);
test.B=toy.B;

test.A=afixbeamb(icomp,nt,ny);
test=mappingd(test);
test.A=[];
test.P=1000*eye(nx);
%test.Sw=2*10^(-2)*eye(ny);
%test.Sv=8*10^(-3)*eye(ny);
test.Sw=swt*eye(ny);
test.Sv=svt*eye(ny);
test.W=[eye(ny,ny); zeros(nx-ny,ny)];
test.C=[eye(ny) zeros(ny,nx-ny)];
```

```
test=myinitialise(test);



  %test.A=[ab ; eye(nx-ny) zeros(nx-ny,ny)];
  Astab=test.A;

  disp(['max eig of test model: ' num2str(max(abs(eig(Astab))))])

  while max(abs(eig(Astab))) > 1
    error('unstable ini system')
  end

  case 6

%nt=input('Enter nt :=');
nt=5;
ny=toy.ny;
nx=nt*ny;

%test=stss(nx,ny);
test=stss(toy.Y);
test.nx=nx;

%test.Y=toy.Y;
test.U=zeros(0,toy.T);
test.B=toy.B;

test.A=afixbeamb(icomp,nt,ny);
test=mappingd(test);
test.A=[];
test.P=1000*eye(nx);
%test.Sw=2*10^(-2)*eye(ny);
%test.Sv=8*10^(-3)*eye(ny);
test.Sw=swt*eye(ny);
test.Sv=svt*eye(ny);
test.W=[eye(ny,ny); zeros(nx-ny,ny)];
test.C=[eye(ny) zeros(ny,nx-ny)];

test=myinitialise(test);



  %test.A=[ab ; eye(nx-ny) zeros(nx-ny,ny)];
  Astab=test.A;

  disp(['max eig of test model: ' num2str(max(abs(eig(Astab))))])
```

```
    while max(abs(eig(Astab))) > 1
      error('unstable ini system')
    end

    otherwise
       error('select 1-5!')
end

%test.P=100*eye(nx);
%test.Sw=2*10^(-2)*eye(ny);
%test.Sv=8*10^(-3)*eye(ny);
```

```
%% return test model
sys=test;
```

myinitialise.m

```
function sys=myinitialise(sys)

% this function initialises the system object, depending on the contents of
% sys

if isempty(sys.Y);

   % if no data exists (sys.Y is empty), then we should generate a toy
   % model and populate the Y and X sequences.

   disp('no data exists');

   n_flag=1; % there should be a neighbourhood defined
   u_flag=0; % there should be no input

   % generate toy model
   %sys=toy(sys,n_flag,u_flag);
   sys=mytoy(sys,n_flag,u_flag);

   %simulate
   sys=simulate(sys);

elseif isempty(sys.A)

   % if data exists but no parameters we should use the data to initialise the system
   % object.

   % old method: run least squares to get initial estimate, then smooth
   % sys=leastsquares(sys);
   % smooth to get state estimates, P and K
   % sys=ksmooth(sys);
   % better method, populate the state, fake a P and a K, then get
   % parameter estimates with maximise using the neighbourhood, then
   % smooth to get a proper P and K

   disp('data exist sys.Y and no parameters sys.A');

   Y=sys.Y;
   T=size(Y,2);
   sys.T=T;
   nx=sys.nx;
   ny=sys.ny;
   dt=nx/ny;

   X=[];
```

```
    for i=dt:-1:1;
       X=[X ; Y(:,i:T) zeros(ny,i-1)];
    end

    sys.X=X;

    % generate a P matrix by just replecating the intial P
    P=sys.P;
    sys.P=repmat(P,[1,1,T]);

    % use the zero matrix for an initial K
    sys.K=zeros(nx,ny,T);

    %calculate Aini
    sys=maximiseini(sys);

    % error catching
    if sum(abs(eig(sys.A))>1), error('unstable A matrix'), end

else
    % if data exists, and the parameters exist then just set up X and P and
    % K without running maximise

    disp('parameter exist')

    Y=sys.Y;
    T=size(Y,2);
    sys.T=T;
    nx=sys.nx;
    ny=sys.ny;
    dt=nx/ny;

    X=[];

    for i=dt:-1:1;
       X=[X ; Y(:,i:T) zeros(ny,i-1)];
    end

    sys.X=X;

    % generate a P matrix by just replecating the intial P
    P=sys.P;
    sys.P=repmat(P,[1,1,T]);

    % use the zero matrix for an initial K
    sys.K=zeros(nx,ny,T);

end
```