ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

RECURSIVE MODELING OF SWITCHED LINEAR SYSTEMS: A BEHAVIORAL APPROACH

M.Sc. THESIS

Mert BAŞTUĞ

Department of Electronics and Communication Engineering

Electronics Engineering Programme

JULY 2012

ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

RECURSIVE MODELING OF SWITCHED LINEAR SYSTEMS: A BEHAVIORAL APPROACH

M.Sc. THESIS

Mert BAŞTUĞ
(504101214)

Department of Electronics and Communication Engineering

Electronics Engineering Programme

Thesis Advisor: Prof. Dr. M. K. Külmiz Çevik

JULY 2012

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DOĞRUSAL ANAHTARLI SİSTEMLERİN ARDIŞIL MODELLENMESİ: DAVRANIŞSAL YAKLAŞIM

YÜKSEK LİSANS TEZİ

Mert BAŞTUĞ
(504101214)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. M. K. Külmiz Çevik

TEMMUZ 2012

Mert BAŞTUĞ, a M.Sc. student of ITU Institute of Science and Technology 504101214 successfully defended the thesis entitled **"RECURSIVE MODELING OF SWITCHED LINEAR SYSTEMS: A BEHAVIORAL APPROACH"**, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. M. K. Külmiz Çevik** Istanbul Technical University	....
Co-advisor :	**Dr. Mihály Petreczky** École des Mines de Douai	....
Jury Members :	**Prof. Dr. Leyla Gören** Istanbul Technical University	....
	Assoc. Prof. Dr. Neslihan Serap Şengör Istanbul Technical University	....
	Asst. Prof. Dr. Özkan Karabacak Istanbul Technical University	....

Date of Submission : 11 June 2012
Date of Defense : 4 July 2012

FOREWORD

I would like to present my very special thanks to Prof. Dr. M. K. Külmiz Çevik and Dr. Mihály Petreczky for always directing me in the best way and their encouragement, and for their valuable suggestions throughout my process of learning and preparing this thesis.

July 2012 Mert BAŞTUĞ

TABLE OF CONTENTS

Page

ABBREVIATIONS

DTLSS	**:** Discrete-Time Linear Switched System
LTI	**:** Linear Time-Invariant
SISO	**:** Single Input Single Output
MPUM	**:** Most Powerful Unfalsified Model
C-MCUM	**:** Controllable - Minimal Complexity Unfalsified Model

LIST OF FIGURES

RECURSIVE MODELING OF SWITCHED LINEAR SYSTEMS: A BEHAVIORAL APPROACH

SUMMARY

Switched systems are hybrid systems which result from interaction of continuous or discrete time dynamical systems with discrete events. In such systems, discrete events triggered by the changes in the switching signal lead the system to operate in different modes. In this thesis, a recursive method for modeling and identifying a finite dimensional discrete time switched system from its input/output signals will be proposed. Recursive partial realization of a discrete time linear switched system (DTLSS) is a special case of this problem and it is treated seperately. The fact that the system model is updated as new data samples are received provides a way to detect the mode changes if the orders of modes are assumed to be known. Thus, a solution to a basic problem in the literature regarding this subject is given. In addition, for correctly identifying each mode, a condition on the dwell times of modes, which is the time between two consecutive changes in the switching signal, is given for both cases. The procedure gives the kernel representations of the local modes of a DTLSS from its partial input/output sequence for both problems. Lastly, problem of constructing state space representations consistent with the data from acquired kernel representations is discussed. For this purpose, a global viewpoint for realization theory of DTLSSs existing in the recent literature is briefly explained. In this work, behavioral approach to system theory, developed by J. C. Willems, is used for modeling dynamical systems. In this approach, a dynamical system is defined by the set of all possible trajectories it can generate.

The research plan is as follows: A recursive procedure is applied to the identification of switched linear systems from impulse response. This problem, known as partial realization problem, is studied in recent literature for discrete time linear systems and results acquired there is modified by taking the mode changes of a switched system into account. For the identification of each mode, the input-output sequence generated by each mode must be sufficiently rich to exhibit all characteristic features of it. In this thesis, necessary and sufficient conditions for the sequence generated by the switched system to be sufficiently rich is derived. This problem is separately studied for the cases of partial realization and identification from arbitrary input/output sequences. By the help of these conditions the minimum needed value of the dwell times of the modes are found. In addition, for the partial realization problem, ways for testing these conditions recursively are examined. Then, the recursive method is generalized for the identification of the switched systems from its arbitrary input-output sequences. The necessary changes are made in identifiability conditions and the recursive procedure is modified accordingly. For making various theoretical predictions and comments and to test the results obtained, the recursive procedure is realized in Matlab for both problems. The constructed codes will hopefully contribute to the comparison of other

works in the literature regarding this topic with our work. Finally the problem about state space representations is discussed.

For the general case, the recursive procedure mainly consists of five steps being "initialization", "error computation", "event detection", "model update" and "identification of the mode" respectively. In the initialization step, the algorithm is initialized by defining the necessary initial conditions for the procedure and giving the order of modes of the DTLSS assumed to be a priori known to the algorithm as inputs. In the error computation phase, the error is computed at each step by applying the found kernel representation in the previous step to the newly acquired data. In the event detection step, a criterion based on this error and the dwell time is checked and the information about whether there is a mode change or not is acquired by the help of this criterion. If there is no event detected, in the model update step, the old representation is updated by multiplying it by the kernel representation of the "most powerful unfalsified model" for the error sequence. If there is an event detected, the procedure gives the kernel representation of the MPUM for the mode and then turns back to the initialization step to identify the new mode.

The results acquired in this thesis are for identification from one observed partial trajectory of a single input single output DTLSS only. Future work can be done to generalize the recursive procedure for application to multi input multi output systems and for the case when there are more than one observed trajectory. In addition, the problem can be considered for the case of known switching signal. The dwell time assumptions may be modified accordingly for that case. Lastly, a persistency of excitation test may be added to the recursive procedure for the identification from arbitrary input/output sequences case and subspace methods can be merged into the recursive procedure.

DOĞRUSAL ANAHTARLI SİSTEMLERİN ARDIŞIL MODELLENMESİ: DAVRANIŞSAL YAKLAŞIM

ÖZET

Anahtarlı sistemler ayrık zamanlı veya sürekli zamanlı dinamik sistemlerin ayrık olaylarla etkileşimi sonucu ortaya çıkan karma sistemlerdir. Bu tür sistemlerde anahtar işaretinin değişimine bağlı olarak tetiklenen ayrık olay, sistemin farklı modlarda çalışmasın ı sağlar. Anahtarlı sistemler denetleyici kontrol sistemlerinin modellenmesi, analizi ve tasarımında kullanılabilir. Darbe etkili mekanik sistemler, röleli veya ideal diyodlu devreler de bu tür sistemlere örnek olarak verilebilir. Bu örnekler, ve anahtarlı sistemlerin sistem-kuramsal özellikleri son yıllarda detaylı olarak çalışılmaktadır. Bu tezde ayrık zamanlı anahtarlı sistemlerin modellenmesi problemi incelenmiştir. Bu amaç için iki farklı yöntem kullanılabilir. Birinci yöntem, dinamik sistemi daha küçük alt sistemlere ayırıp, fiziksel yasaları ve temel prensipleri kullanarak sistemin uygun bir temsilini (bir diferansiyel denklem veya fark denklemi takımı gibi) bulmaktır. Bu çalışmada kullanılan bir diğer yöntem ise sistemin gözlenmiş giriş/çıkış çiftlerinden yararlanarak sistemin davranışını tam ya da yaklaşık olarak açıklayan bir model bulmaktır. Bu yaklaşım, literatürde "sistem tanıma" olarak adlandırılır. Genelde bu yaklaşım, pratik durumlar için daha uygun, ölçümlerin üzerinde stokastik bir gürültünün var olduğu durumlar için kullanılsa da bu çalışmada daha temel bir problem olan, ideal veriden sistemin tam olarak tanınması problemi ele alınmıştır.

Bu tezde giriş ve çıkış işaretlerinden sonlu boyutta ayrık zamanlı anahtarlı sistemlerin modellenmesi ve tanınması için ardışıl bir yöntem önerilmiştir. Ayrık zamanlı doğrusal anahtarlı sistemlerin kısmi gerçeklemesi bu problemin özel bir halidir ve ayrı olarak incelenmiştir. Sistem modelinin veriler geldikçe güncellenmesi, modların mertebelerinin bilinmesi halinde, mod değişimlerinin sezilmesine olanak sağlamaktadır. Böylece literatürde bu konuda yapılan çalışmalarda karşılaşılan temel bir soruna çözüm getirilmektedir. Ayrıca bu iki problem için, anahtarlı sistemin her bir modunun tek olarak tanınabilmesi için modların sağlaması, ve art arda gelen iki anahtarlama anı arasında geçen zaman olan bekleme süresinin sağlaması gereken koşullar çıkarılmıştır. İki problem için de, ardışıl prosedür, anahtarlı sistemin ürettiği kısmi giriş/çıkış çifti dizisinden her bir modunun sıfır gösterilimini elde etmektedir. Son olarak, elde edilen sıfır gösterilimlerinden, gözlenen veriyle uyumlu durum gösterilimlerinin elde edilmesi problemi tartışılmıştır. Bunun için, ayrık zamanlı anahtarlı sistemlerin gerçekleme kuramına ilişkin yakın zamanda literatürde sunulmuş daha global bir bakış açısı kısaca açıklanmıştır. Uyumlu durum gösterilimleri elde edebilmek için literatürde bulunan bir yöntem önerilmiştir.

Tezde, J. C. Willems tarafından geliştirilen sistem kuramına davranışsal yaklaşım, dinamik sistemlerin modellenmesi için kullanılmıştır. Bu yaklaşımda, bir dinamik sistem, üretebileceği her yörüngeden (çözümden) oluşan bir küme (davranış kümesi)

ile tanımlanır. Dinamik sistemin bir parametre kümesi ile değil, bu şekilde bir fonksiyon (ayrık zamanlı sistemler için dizi) kümesi ile tanımlanması sistem kuramındaki bazı temel kavramların belirli sistem temsillerinden bağımsız olarak verilmesini sağlamaktadır (parametreye dayalı tanımlar "yönetilebilirlik" gibi herhangi bir dinamik sisteme has özelliklerin, sistemin kendi özelliği değil, sistemin "durum gösterilimi" gibi belirli bir "temsili"nin özelliği olduğu yanılgısına yol açabilir).

Tezin araştırma planı şu şekildedir: Öncelikle tezde kullanılan davranışsal yaklaşımdaki modelleme amacına ilişkin temel kavramlar ve tanımlar araştırılmış ve kısaca açıklanmıştır. Bu yaklaşım kullanılarak, bir ardışıl yöntem, anahtarlı doğrusal sistemin sonlu sayıda birim dürtü (impulse) yanıtı verisinden tanınmasına uygulanmıştır. Literatürde "kısmi gerçekleme" olarak bilinen bu problem üzerine ayrık zamanlı doğrusal sistemler için yakın zamanda çalışılmıştır. Elde edilen sonuçlar bu tezde anahtarlı sistemin mod değişimleri göz önüne alınarak genişletilmiştir. Her bir modun tanınabilmesi için, her bir mod tarafından üretilen giriş/çıkış çiftleri dizisinin o modun bütün karakteristik özelliklerini yansıtması, başka bir deyişle yeterince zengin olması gerekmektedir. Bu çalışmada, anahtarlı sistemin ürettiği giriş/çıkış dizisinin yeterince zengin olabilmesi için gerekli ve yeterli koşullar çıkarılmıştır. Bu problem, kısmi gerçekleme ve keyfi giriş/çıkış dizisinden modları tanıma amaçları için ayrı ayrı ele alınmıştır. Bu koşullar yardımıyla, anahtarlı sistemin modlarının bekleme sürelerinin sağlaması gereken minimum süreler bulunmuştur. Ayrıca, kısmi gerçekleme problemi için, bu koşulları ardışıl olarak her aşamada kontrol etme yolları incelenmiştir. Sonra, ardışıl yöntem, anahtarlı sistemin modlarının sistemin ürettiği keyfi giriş/çıkış çifti dizisinden tanınması amacıyla genelleştirilmiştir. Tanınabilme koşullarında gerekli değişiklikler yapılmıştır ve ardışıl yöntem de buna göre yeniden düzenlenmiştir. Her bir problem için anahtarlama işaretinin bilinmediği fakat anahtarlı sistemin modlarının derecesinin bilindiği varsayılmış ve tek bir çözüm aranmıştır. Anahtarlı sistemlerin tanınması konusunda literatürde yapılan çalışmalarda, anahtarlama olayının sezilmesi problemi birçok zorluğa yol açmaktadır. Tezde modların bekleme sürelerine ilişkin yapılan belli varsayımlar ve yöntemin her yeni veri örneği geldiğinde ardışıl olarak modeli güncellemesi sayesinde bu problem çözülmüştür.

Genel durum için, ardışıl yöntem ana hatlarıyla "başlangıç", "hata bulma", "anahtarlama sezme", "model güncelleme" ve "modun tanınması" olmak üzere beş aşamadan oluşmaktadır. Başlangıç aşamasında prosedürün başlaması için gereken ilk koşullar ve "a priori" bilindiği varsayılan mod mertebeleri algoritmaya verilir. Hata bulma aşamasında her adımda bir önceki adımda bulunan sistem temsilinin yeni elde edilen veriye uygulandığında ortaya çıkan hata hesaplanır. Anahtarlama sezme aşamasında, bu hataya ilişkin belirli bir kriter kontrol edilerek bir anahtarlama olup olmadığı anlaşılır. Anahtarlama yoksa eski model, hataya ilişkin bulunan sıfır gösterilimiyle çarpılarak güncellenir. Anahtarlama var ise prosedür, çıkış olarak moda ilişkin modelin temsilini vererek başlangıç aşamasına geri döner.

Çalışma sürecinde çeşitli kuramsal tahminler ve yorumlar yapabilmek ve elde edilen sonuçları deneyebilmek için ardışıl yöntem, her iki ana problem için de Matlab ortamında gerçeklenmiştir. Oluşturulan kodların, literatürde bu konuda yapılan diğer çalışmalarla bu çalışmanın karşılaştırılmasını sağlaması ve gelecek çalışmalara destek olması umulmaktadır. Son olarak modların elde edilen sıfır gösterilimlerinden, uyumlu durum gösterilimleri elde edilmesi problemi tartışılmıştır. Bu bölümde ayrıca doğrusal anahtarlı sistemin yerel doğrusal sistemlerin uç uca eklenmesi olarak yorumlanmasının kısmi gerçekleme probleminde getirdiği bazı sorunlar, konu hakkında yeni yapılan çalışmalar kullanılarak gösterilmeye çalışılmıştır. Sonuç bölümünde ise çalışmanın ileride nasıl geliştirilebileceğine ilişkin bazı önerilerde bulunulmuştur.

Tezin son aşamasında, ardışıl yöntem yardımıyla modlara ilişkin bulunan sıfır gösterilimlerinden, aynı modlara ilişkin gözlenmiş giriş/çıkış verisiyle tutarlı olacak durum gösterilimlerinin elde edilmesi problemi tartışılmıştır. Bu problem, çözümü apaçık olan bir problem değildir. Doğrusal sistemler için böyle bir problem söz konusu değildir fakat anahtarlı sistemlerde, anahtarlama anından önceki modun, sonraki moda ilişkin bir ilk koşul yaratması böyle bir probleme yol açar. Bu bölümde, tezde kullanılan, anahtarlı sistemi tek tek doğrusal sistemlerin uç uca eklenmiş hali olarak yorumlamanın yol açtığı kavram bulanıklıkları da M. Petreczky'nin çalışmalarndan yararlanılarak tartışılmıştır (Söz gelimi, anahtarlı bir sistemin minimal bir gerçekleme olması, yerel modlarının her birinin minimal olması anlamına gelmez). Bölüm sonunda bahsedilen probleme ilişkin bir çözüm önerisi, varolan literatür kaynak gösterilerek verilmiştir.

Tezde elde edilen sonuçlar, tek giriş tek çıkışlı bir ayrık zamanlı doğrusal anahtarlı sistemin, anahtarlama işaretinin bilinmediği varsayıldığında, tek bir giriş çıkış çifti dizisinden tanınması için verilmiştir. Çalışmanın olası zenginleştirilmesi, ardışıl yöntemin çok giriş çok çıkışlı sistemler için de kullanılabilir hale getirilmesi veya elde gözlenmiş birden çok giriş çıkış çifti dizisi olduğu durumlarda da uygulamaya elverişli olması çabalanarak gerçekleştirilebilinir. Ek olarak, tezdeki problem, anahtarlama işaretinin bilinmesi durumu için de ele alınabilir. Bekleme süresi üzerine yapılan varsayımlar, bu duruma göre değişiklik gösterebilir. Son olarak, keyfi giriş çıkış çifti dizisinden sistemi tanıma problemi için, ardışıl prosedür içine, girişin her aşamada yeterince uyarıcı olup olmadığını kontrol eden bir kısım eklenebilir. Ayrıca, altuzay yöntemleri, ardışıl prosedür içine yerleştirilebilir.

1. INTRODUCTION

Switched systems are models of dynamical phenomena whose behavior changes among a number of submodels depending on a logical decision mechanism. Such systems are used in modeling, analysis and design of supervisory control systems, mechanical systems with impact, circuits with relays or ideal diodes for instance. These examples and some system theoretic properties of switched systems have been studied in [1, 2] and the references therein. In this thesis, we consider the problem of modeling a discrete-time dynamical switched linear system and finding an appropriate representation for the model is considered. For this purpose, one can use two different ways of approach. First approach is to separate the dynamical system into subsystems and elements and, by using physical laws and first principles, to find a representation (like a set of equations) that defines the system. Second approach, which is used in this work, is to find a representation for the behavior of the system by using observed input-output measurements. This approach is known as system identification in the literature. Although this approach generally used with the existence of stochastic noise on the measurements, in this work, the problem of identification from exact data is considered. In [3] extensive information for system identification from noisy measurements is given. In references like [4], [5] and [6] it is argued that the problem of exact identification is a more basic one and it should preceed the problem of stochastic identification. In this thesis, the behavioral approach for system theory (see [7,8]) is adopted to develope a recursive method for exact identification of discrete time switched linear systems (abbreviated as DTLSSs).

First, the problem of identification of a DTLSS from impulse response sequences (which is known as the partial realization problem) is considered. This is a classical problem which has been extensively studied in [9]. For linear systems the solution of the problem using the generating system approach has been described in [4]. This method gives the possible orders and parametrization of all linear systems which have a given sequence of impulse response samples. In the present work the recursive method given in [4, 10] for the construction of the generating system is appropriately modified and applied to the partial realization of switched systems. Then, the problem of identification of a DTLSS from arbitrary input-output sequences is considered. The solution to the problem of recursively modeling a linear system from continuous-time polynomial-exponential time series is given in [4]. In the same reference, it is pointed out that the modeling of a discrete time linear system from arbitrary input-output data is a special case of that problem. Necessary modifications for the identification of switched systems are made and a procedure is given.

For both problems, it is assumed that the switching signal is unknown but the orders of the modes are known and a unique solution is sought. Identifiability conditions for both cases are derived and explained. The problem studied in the present paper is similar to those in [11–13] which address identification of switched or hybrid systems. The detection of mode changes causes various difficulties in these works. By adopting certain assumptions on dwell times of the modes of the DTLSS, this problem is easily resolved by the recursive method described in this work since the model derived is tested and updated as new data samples are received.

In the recent related work [14] realizability of a family of input-output maps by a switched linear system has been considered and minimality of the realizations has been characterized. The related concepts and definitions are briefly given in the last chapter, and ways of finding state space representations for the modes consistent with the observed data are discussed.

The thesis is organised as follows: The preliminaries of behavioral approach for the context of modeling are given in Chapter 2. Basic definitions are given and different system representations are presented. At the end of the chapter, the problem of exact identification is stated. In Chapter 3, recursive partial realization problem is considered. Identifiability conditions and the recursive procedure for this case is given. In Chapter 4, the problem is generalized for recursive identification from arbitrary input-output data. In Chapter 5 state space realization problem is considered and ways of finding state space representations for the modes of the DTLSS are discussed. The thesis is concluded with the summary of results and ideas for possible future work.

2. PRELIMINARIES OF BEHAVIORAL APPROACH FOR MODELING CONTEXT

Here, we present some preliminaries about "the behavioral approach to linear system theory" which is developed by Jan C. Willems. The motivation for this approach is the following: First, in many practical cases (for example in electrical circuits) the distinction between inputs and outputs is not *a priori* clear; instead it should follow as a consequence of the modeling. Second, it is desirable to have representation-free definitions for classical conceptes in system theory like controllability or observability. This kind of concepts are not properties of a particular representation of a system, rather they are properties of the dynamical system itself. In addition, one should be able to treat the different representations of a given system (for example: input-output or state space representations) in a unified way [4]. In behavioral approach, a dynamical system is defined as a collection of trajectories rather than a collection of parameters. Therefore, it creates the possibility to define classical concepts in system theory in a representation free way. In the next sections, some basic definitions in behavioral framework needed for the purpose of this thesis will be given. The chapter will end with the definitions of exact identification problem and the central object of this problem called "the most powerful unfalsified model (MPUM)".

2.1 Linear Time-Invariant Dynamical Systems

Definition 2.1.1. (Dynamical System [15]) A dynamical system Σ is a 3-tuple $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$, with $\mathbb{T} \subseteq \mathbb{R}$ the time axis, $\mathbb{W}$ the signal space, and $\mathfrak{B} \subseteq \mathbb{W}^{\mathbb{T}}$ the behavior.

In Definition 2.1.1, $\mathbb{T}$ is the time axis which is $\mathbb{R}$ or $\mathbb{R}_+$ for continuous time case and $\mathbb{Z}$ or $\mathbb{N}$ for discrete time case. The set $\mathbb{W}$ is called the signal space where the signals take on their values. The set of all functions (trajectories) $w : \mathbb{T} \to \mathbb{W}$ is denoted by $\mathbb{W}^{\mathbb{T}}$ in the definition. A subset of this set $\mathbb{W}^{\mathbb{T}}$ is called the behavior of the system and denoted by $\mathfrak{B}$. Behavior set $\mathfrak{B}$ consists of all possible trajectories $w \in \mathbb{W}^{\mathbb{T}}$ that the system can generate and this is the set that defines a particular dynamical system. It can be seen that a dynamical system is defined with the three sets in Definition 2.1.1 where the behavior $\mathfrak{B}$ is a collection of trajectories. For the purpose of this work, we take the time axis as either $\mathbb{Z}$ or $\mathbb{N}$ since we deal with discrete time systems.

Definition 2.1.2. (Linearity) A dynamical system $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ is linear if and only if

$$w_1(t), w_2(t) \in \mathfrak{B} \Rightarrow (\alpha w_1(t) + \beta w_2(t)) \in \mathfrak{B} \quad \textbf{(2.1)}$$

where $w_1(t)$, $w_2(t)$ are any two trajectories in $\mathfrak{B}$ and $\alpha, \beta \in \mathbb{R}$ are arbitrary constants.

Definition 2.1.2 implies that a system is called linear if and only if any linear combination of trajectories in $\mathfrak{B}$ is also an element of $\mathfrak{B}$. In other words, a system $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ is linear when the signal space $\mathbb{W}$ is a vector space and $\mathfrak{B}$ is a linear subspace of $\mathbb{W}^{\mathbb{T}}$.

Definition 2.1.3. (Time-Invariance) Let $w(t+\Delta)$ denote the Δ times backward shifted trajectory $w(t)$ in the time axis $\mathbb{T}$. A dynamical system $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ is time-invariant if and only if

$$w(t) \in \mathfrak{B} \Rightarrow w(t+\Delta) \in \mathfrak{B}, \forall \Delta \in \mathbb{T} \quad \textbf{(2.2)}$$

where $w(t)$ is any trajectory of $\mathfrak{B}$.

Definition 2.1.3 implies that if a time series w is a trajectory of a time-invariant system, then all its shifts are also trajectories of that system.

In the following, the class of all discrete-time, linear time-invariant (LTI) systems with q variables (q is the dimension of the signal space $\mathbb{W}$) will be denoted by $\mathfrak{L}^q$.

2.2 Dynamical System Representations

In the classical theory, a property of the system is defined as a property of a particular representation (For example, controllability is defined as a property of a state space representation). This implicitly allows that such a definition might be representation dependent (it might only hold for that particular representation) and therefore not a specific property of the system itself. In behavioral approach, a certain property is first defined in terms of the behavior $\mathfrak{B}$, then the implications of it on the parameters of a particular system representation is found. It is important to emphasize that a system is *defined* as a collection of trajectories and then can be *represented* in a particular way (For example by using differential or difference equations). In this section, different representations of dynamical systems and their properties are discussed briefly.

2.2.1 Kernel representation

Consider the difference equation

$$R_0 w(t) + R_1 w(t+l) + ... + R_l w(t+l) = 0, \text{ where } R_i \in \mathbb{R}^{g \times q} \text{ for } i = 0, 1, ..., l. \quad \textbf{(2.3)}$$

This vector equation shows the recurrence relation between the consecutive samples of the time series w. Assuming that $R_l \neq 0$, the maximum number of shifts l is called the *lag* of the equation. Since (**2.3**) is a vector equation consisting of g rows, l is the largest lag among the lags $l_1, ..., l_g$ of all scalar equations.

The equation (**2.3**) induces a dynamical system whose behavior set is defined as

$$\mathfrak{B} = \{w \in (\mathbb{R}^q)^{\mathbb{Z}} | \text{ (2.3) holds}\}. \quad \textbf{(2.4)}$$

It means that the behavior of discrete time linear dynamical system can be represented and analyzed with the use of a vector difference equation. It turns out, however, it is more advantageous to use polynomial matrix algebra for analyzing $\mathfrak{B}$. Let σ be the backward shift operator whose operation is defined as $(\sigma w)(t) = w(t+1)$. Therefore, (**2.3**) can be written in a more compact form in terms of the polynomial matrix with the indeterminate s:

$$R(s) := R_0 + R_1 s^1 + R_2 s^2 + ... + R_l s^l \in \mathbb{R}^{g \times q}[s] \quad \textbf{(2.5)}$$

as

$$R(\sigma) w = 0. \quad \textbf{(2.6)}$$

Consequently, operations on the system of difference equations can be represented by operations on the polynomial matrix $R(s)$. The behavior of the system induced by (**2.3**) is

$$\mathfrak{B} = ker R(\sigma) := \{w \in (\mathbb{R}^q)^{\mathbb{N}} | R(\sigma) w = 0\}. \quad \textbf{(2.7)}$$

(**2.7**) is called the kernel representation of the system.

In [Wil86a] it is proven that without loss of generality one can assume the existence of a kernel representation $\mathfrak{B} = \ker R(\sigma)$ of an LTI complete system $\mathfrak{B} \in \mathfrak{L}^q$. Briefly, the linearity of the system induced by (**2.3**) follows from the linearity of (**2.3**) with respect to w. The time-invariance follows from the constant coefficient matices $R_0, ..., R_l$, and the finite dimensionality of the system follows from the fact that (**2.3**) involves a finite number l of shifts of the time series.

Note that a kernel representation for a given $\mathfrak{B}$ is not unique. The nonuniqueness is due to the possible existence of linearly dependent equations and equivalence of some representations. We need the following important definition and theorems for clearly explaining these facts.

Definition 2.2.1.1. (Unimodular Matrix) [8] A matrix $U(s) \in \mathbb{R}^{g\times g}[s]$ which represents elementary row operations on a matrix $R(s) \in \mathbb{R}^{g\times q}[s]$ is called unimodular. These operations can be defined as following:

(i) Permute any two rows of $R(s)$.

(ii) Multiply a row of $R(s)$ by a constant.

(iii) Multiply row i of $R(s)$ by s^d and add it to row j, where $d \in \mathbb{N}$ and $i, j \in 1, 2, ..., g$ with $i \neq j$.

Let the matrices M, C and $Q(s)$ be the matrices representing any number of these operations defined in *(i)*, *(ii)* and *(iii)* respectively. $U(s)$ can always be factored as

$$U(s) = MCQ(s). \tag{2.8}$$

Furthermore, $U(s)$ is a matrix with nonzero constant determinant i.e., $\det U(s) = c$ where c is a nonzero constant.

Notice that applying these operations to a set of difference equations, does not change the corresponding behavior, but it changes the representation. This fact will be stated in Theorem 2.2.1.4. The following theorem states that linearly dependent equations can always be destroyed by left multiplication with unimodular matrices. For stating the theorem we need one more definition.

Definition 2.2.1.2. (Matrix of full row rank) [16] Polynomial vectors $r_i(s) \in \mathbb{R}^{1\times q}[s]$, $i = 1,2,...,g$ ($\mathbb{R}^{1\times q}[s]$ denotes the set of all $1 \times q$ polynomial vectors in real coefficients) are called linearly independent over the field $\mathbb{R}$ if and only if

$$a_1 r_1(s) + a_2 r_2(s) + ... + a_g r_g(s) = 0 \Leftrightarrow a_i = 0, i = 1,2,...,g \tag{2.9}$$

where $a_i \in \mathbb{R}$, $i = 1,2,...,g$ are real coefficients. A matrix $R(s) \in \mathbb{R}^{g\times q}[s]$ is called to have full row rank (or "of rull row rank") if and only if all row vectors of $R(s)$ is linearly independent.

Theorem 2.2.1.3. *Every behavior $\mathfrak{B}$ defined by $R(\sigma)w = 0$, $R(s) \in \mathbb{R}^{g\times q}[s]$ (where $R(s)$ is not of full row rank) admits an equivalent full row rank (minimal) representation, that is, there exists a representation $\widetilde{R}(\sigma)w = 0$ of $\mathfrak{B}$ with $\widetilde{R}(s) \in \mathbb{R}^{\widetilde{g}\times q}[s]$, $\widetilde{g} < g$ of full row rank where $\begin{pmatrix} \widetilde{R}(s) \\ 0 \end{pmatrix} = U(s)R(s)$ and the matrix $U(s)$ is unimodular.*

Proof. For the proof, see [8]. □

Theorem 2.2.1.3 implies that for a given system $\mathfrak{B}$ there always exists a kernel representation in which $R(s)$ has full row rank [Wil91]. Such a representation is called a *minimal kernel representation*.

Theorem 2.2.1.4. *Two polynomial matrices $R_1 \in \mathbb{R}^{g\times q}[s]$ and $R_2 \in \mathbb{R}^{g\times q}[s]$ of full row rank, represent the same behavior if and only if there exists a unimodular matrix $U(s)$ such that $R_1(s) = U(s)R_2(s)$.*

Proof. For the proof, see [8]. □

Theorem 2.2.1.4 states that also the minimal representation is nonunique. The representation changes under unimodular transformations.

In a minimal kernel representation, the number of equations is minimal among all possible kernel representations of $\mathfrak{B}$ and this number is defined as $p :=$row dim $R(s)$. We define the degree of row i of $R(s)$ as the highest power of s in that row and denote it by l_i or degr_i. Also define the maximum row degree of $R(s)$ (maximum lag among all scalar equations) as $l := \max_{i=1,...,p} l_i$, and the sum of the lags as $n := \sum_{i=1}^{p} l_i$. In [15] it is stated that there exists a minimal kernel representation $\mathfrak{B} = \ker R(\sigma)$, in which the numbers p,l and n are simultaneously minimal over all possible kernel representations. Such a representation is called *shortest lag representation* and it is achived when $R(s)$ is row reduced. Below, the definition of row reducedness is given. From now on, the numbers p,l and n in the shortest lag representation of a system will be denoted by $p(\mathfrak{B})$, $l(\mathfrak{B})$ and $n(\mathfrak{B})$ respectively.

Definition 2.2.1.5. (Row reducedness) Let the polynomial matrix $R(s)$ be written row by row

$$R = \begin{bmatrix} r_1 & \dots & r_p \end{bmatrix}^T, \ \deg(r_i) = l_i. \tag{2.10}$$

$R(s)$ is row reduced if the leading row coefficient matrix (i.e., the matrix of which the (i,j)th entry is the coefficient of the term with power l_i of $R_{ij}(s)$) is full row rank.

The minimal number of equations $p(\mathfrak{B})$, the lag $l(\mathfrak{B})$ and the total lag $n(\mathfrak{B})$ in a hortest lag representation are therefore invariants of $\mathfrak{B}$. It turns out $p(\mathfrak{B})$ is equal to the number of outputs in an input/output representation (It is called output cardinality). Correspondingly, the integer $m(\mathfrak{B}) = q - p(\mathfrak{B})$ is also an invariant of $\mathfrak{B}$ and it is equal to the number of inputs (It is called input cardinality). The total lag $n(\mathfrak{B})$ is equal to the state dimension in a minimal state space representation of $\mathfrak{B}$.

2.2.2 Input/output representation

Definition 2.2.2.1. Input/Output Partition) [8] Let $\mathfrak{B}$ be a behavior with signal space $\mathbb{R}^q$. Partition the signal space as $\mathbb{R}^q = \mathbb{R}^m \times \mathbb{R}^p$ and correspondingly as $w = \text{col}(w_1, w_2)$, $w_1 \in \mathbb{R}^{m\mathbb{T}}$ and $w_2 \in \mathbb{R}^{p\mathbb{T}}$. This partition is called an *input/output partition* if:

1. w_1 is *free*; i.e., for all $w_1 \in (\mathbb{R}^m)^{\mathbb{T}}$ there exists a $w_2 \in (\mathbb{R}^p)^{\mathbb{T}}$ such that $w = \text{col}(w_1, w_2) \in \mathfrak{B}$.

2. w_2 does not contain any further free components; i.e., given w_1, none of the components of w_2 can be chosen freely. Stated differently, w_1 is *maximally free*.

If 1 and 2 hold then w_1 is called an *input variable* and w_2 is called an *output variable*.

Definition 2.2.2.2. (Proper matrix) A matrix of rational functions (i.e., each of the entries is the ratio of two polynomials) is called *proper* if in each entry, the degree of the numerator does not exceed the degree of the denominator.

Theorem 2.2.2.3. *Let $R(s) \in \mathbb{R}^{p\times q}[s]$ be of full row rank with $p < q$. then there exists a partition of $R(s)$ of the form:*

$$R(s) = \begin{bmatrix} -N(s) & D(s) \end{bmatrix} \tag{2.11}$$

where $D(s) \in \mathbb{R}^{p\times p}$ is composed of the columns of $R(s)$ which makes deg det$D(s)$ maximal among all $p \times p$ submatrices of $R(s)$, and where $N(s)$ is composed of the remaining columns. There also exists a corresponding partitioning of w:

$$w = \begin{bmatrix} u \\ y \end{bmatrix}, \tag{2.12}$$

where the elements of w corresponding to the columns of $R(s)$ which comprises $-N(s)$ are chosen as inputs u, and elements of w corresponding to the columns of $R(s)$ which comprises $D(s)$ are chosen as outputs y. This partitioning is in the sense of Definition 2.2.2.1 and the corresponding input/output behavioral equations can be written as

$$D(\sigma)y = N(\sigma)u. \tag{2.13}$$

Also, $N(s)$ and $D(s)$ satisfy:

- *det $D(s) \neq 0$*
- *$D^{-1}(s)N(s)$ is a matrix of proper rational functions.*

Proof. For a proof, see [8]. □

Definition 2.2.2.4. (Transfer matrix) Let the signal space $\mathbb{R}^q$ of a behavior partitioned as in Theorem 3.2, i.e.

$$R(s) = \begin{bmatrix} -N(s) & D(s) \end{bmatrix}, \Pi w = \begin{bmatrix} u \\ y \end{bmatrix} \tag{2.14}$$

where Π is a suitable $q \times q$ permutation matrix, $D(s) \in \mathbb{R}^{p\times p}[s]$, $N(s) \in \mathbb{R}^{p\times m}[s]$ and $u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$. The $p \times m$ matrix $D^{-1}(s)N(s)$ is called the *transfer matrix* of the behavior.

Theorem 2.2.2.3 and Definition 2.2.2.4 imply that one can always find an input/output partitioning in the sense of Definition 2.2.2.1 such that the behavior is defined by the equation

$$D(\sigma)y = N(\sigma)u. \tag{2.15}$$

The system $\mathfrak{B}$ induced by an input/output equation with parameters (D,N) and input/output partitioning defined by Π can be formally defined as

$$\mathfrak{B}_{i/o}(D,N,\Pi) := \{\Pi w := \mathrm{col}(u,y) \in (\mathbb{R}^q)^{\mathbb{N}} | D(\sigma)y = N(\sigma)u\}. \tag{2.16}$$

The representation (**2.16**) is called an input/output representation of the system $\mathfrak{B}$. The matrix $D^{-1}(s)N(s)$ is called the transfer matrix of the behavior defined by (**2.16**) [8].

2.2.3 State space representation

In the modeling procedure, there are variables whose relation wanted to be defined. Those variables are called the *manifest variables* and they were previously denoted by w. However, in the process of modeling from first principles, calculations involve different variables than the model aims to describe. These variables are called the *latent variables* (For example when modeling the port behavior of a one port electrical circuit, port voltage and current are the manifest variables and voltages and currents among all other branches are latent variables). *State variables* are special latent variables that specify the memory of the system. Below, a definition is given.

Definition 2.2.3.1. (State variables [7]) Latent variables x are called state variables if they satisfy the following axiom of state:

$$(w_1,x_1),(w_2,x_2) \in \mathfrak{B}, t \in \mathbb{N}, \text{ and } x_1(t) = x_2(t) \Rightarrow (w,x) \in \mathfrak{B}, \tag{2.17}$$

where

$$(w(\tau),x(\tau)) := \begin{cases} (w_1(\tau),x_1(\tau)) \text{ for } \tau < t, \\ (w_2(\tau),x_2(\tau)) \text{ for } \tau \geq t. \end{cases} \tag{2.18}$$

It turns out any LTI system $\mathfrak{B}$ admits a representation by an input/state/output equation

$$\sigma x = Ax + Bu, \quad y = Cx + Du, \quad \Pi w = \mathrm{col}(u,y), \tag{2.19}$$

in which both the input/output and the state structure of the system are explicitly displayed [15]. The system $\mathfrak{B}$ induced by an input/state/output equation with parameters (A,B,C,D) and Π, is formally defined as

$$\begin{aligned}\mathfrak{B}_{i/s/o}(A,B,C,D,\Pi) := \{\Pi w := \mathrm{col}(u,y) \in (\mathbb{R}^q)^{\mathbb{N}} | \exists x \in (\mathbb{R}^n)^{\mathbb{N}},\\ \text{such that } \sigma x = Ax + Bu, y = Cx + Du\}.\end{aligned} \quad \textbf{(2.20)}$$

(**2.20**) is called an input/state/output representation of the system $\mathfrak{B}$.

2.3 Autonomous Systems

Definition 2.3.1. (Autonomous System) A system $\mathfrak{B}$ is *autonomous* if for any trajectory $w \in \mathfrak{B}$ the past

$$w_- := (\ldots, w(-2), w(-1)) \quad \textbf{(2.21)}$$

of w completely determines its future

$$w_+ := (w(0), w(1), \ldots). \quad \textbf{(2.22)}$$

A system $\mathfrak{B}$ is autonomous if and only if its input cardinality m equals zero. That means there are no external *free* variables (inputs). Every trajectory's future is completely determined by its past. Therefore an autonomous LTI system is parameterized by the pair of matrices A and C via the state space representation

$$\sigma x = Ax, \quad y = Cx, \quad w = y. \quad \textbf{(2.23)}$$

The system induced by the state space representation with parameters (A,C) is

$$\mathfrak{B}_{i/s/o}(A,C) := \{w \in (\mathbb{R}^p)^{\mathbb{N}} | \exists x \in (\mathbb{R}^n)^{\mathbb{N}}, \text{ such that } \sigma x = Ax, w = Cx\}. \quad \textbf{(2.24)}$$

Since there are no free variables, behavior of an autonomous system is finite dimensional where dim $(\mathfrak{B}) = n$. An autonomous LTI system can also be parameterized in a minimal kernel representation $\mathfrak{B} = \ker R(\sigma)$ by a square nonsingular polynomial matrix $R(s)$ i.e., $R(s) \in \mathbb{R}^{p\times p}[s]$, $\det R(s) \neq 0$. Note that this is a special case of an input/output representation, a behavior with outputs only [8] (An autonomous system can be interpreted as an input/output system with zero inputs).

2.4 Controllability

Definition 2.4.1. (Controllability) The system $\mathfrak{B}$ is *controllable* if for any two trajectories $w_1, w_2 \in \mathfrak{B}$ there exists a third trajectory $w \in \mathfrak{B}$ and a time instant $t' \geq 0, t' \in \mathbb{Z}$, such that $w(t) = w_1(t)$ for all $t \leq 0$ and $w(t) = w_2(t - t')$ for all $t \geq t'$.

Controllability implies that we can steer any trajectory to another one within the behavior provided we allow a delay [8]. A test for controllability of the system $\mathfrak{B}$ in terms of the parameter $R(s) \in \mathbb{R}^{g \times q}[s]$ in a kernel representation $\mathfrak{B} = \ker R(\sigma)$ is given in [7]: $\mathfrak{B}$ is controllable if and only if the matrix $R(s)$ has a constant rank for all $s \in \mathbb{C}$ (This equivalently means $\mathfrak{B}$ is controllable if and only if the matrix $R(s)$ is left prime). In terms of input/output representation $\mathfrak{B} = \mathfrak{B}_{i/o}(D,N)$, $D(s)$ and $N(s)$ must be left coprime for $\mathfrak{B}$ to be controllable.

2.5 Complexity of a Linear Time-Invariant System

The complexity of an LTI system is parameterized by the ordered pair $c(\mathfrak{B}) := (m(\mathfrak{B}), n(\mathfrak{B}))$ where $m(\mathfrak{B})$ and $n(\mathfrak{B})$ shows the input cardinality and total lag of the system respectively. The parameter $c(\mathfrak{B})$ is called the complexity of the system. Define the lexicographic ordering as follows: Given the vectors of n real numbers a, b we write $a \geq b$ if $a = b$ or if for some $j \in 1, 2, ..., n$, $a_i = b_i$, $i < j$, and $a_j > b_j$. By using this ordering we call a system $\mathfrak{B}_2$ more complex than $\mathfrak{B}_1$ if $c(\mathfrak{B}_1) \leq c(\mathfrak{B}_2)$. Notice since there are no inputs in an autonomous system complexity of an autonomous system is always less than the complexiy of a nonautonomous system [4].

2.6 Exact Identification

The exact identification problem is defined as follows: Given a trajectory w of a discrete time LTI system $\mathfrak{B}$, find a representation of $\mathfrak{B}$. In this thesis problem of finding a kernel representation is considered. The exact identification is a basic and important system theoretic problem. It includes the classical impulse response realization (partial realization) problem and it is a prerequisite for the study of more complicated approximate, stochastic, and stochastic/approximate identification problems [17].

In this thesis, the problem is considered for a finite number of data and although the behavioral setting is used, the results are given in input/output setting. A central object for the discussion of exact identification problem is the "Most Powerful Unfalsified Model (MPUM)" and a definition is given below.

Definition 2.6.1. (MPUM) Assume we have a data set $D = \{w_0, w_1, ..., w_N\}$ where $w_i \in (\mathbb{R}^q)^{\mathbb{N}}$ for $i = 0, 1, ..., N$ are observed trajectories. A behavior $\mathfrak{B}$ is called an unfalsified model for D if $D \subseteq \mathfrak{B}$. A model $\mathfrak{B}_1$ is called more powerful than $\mathfrak{B}_2$ if $\mathfrak{B}_1 \subseteq \mathfrak{B}_2$. A model $\mathfrak{B}^*$ is called the *most powerful unfalsified model (MPUM)* for D, if $\mathfrak{B}^*$ is unfalsified for D and $D \subseteq \mathfrak{B} \Rightarrow \mathfrak{B}^* \subseteq \mathfrak{B}$.

Let us define the restriction of the behavior $\mathfrak{B}^* \subseteq (\mathbb{R}^q)^{\mathbb{N}}$ to the time interval $[1, T]$ where $T \in \mathbb{N}$ as $\mathfrak{B}^*|_{[1,T]}$ i.e., $\mathfrak{B}^*|_{[1,T]}$ consists of the parts of all trajectories in $\mathfrak{B}$ between the time interval $[1, T]$ for any $T \in \mathbb{N}$. It is shown in [5] that a MPUM for D exists and the system $\mathfrak{B}^* \subseteq (\mathbb{R}^q)^{\mathbb{N}}$ is an MPUM of the set D in the model class $\mathfrak{L}^q$ if it is

1. finite dimensional LTI,

2. unfalsified, i.e., $D \subseteq \mathfrak{B}^*|_{[1,T]}$

3. most powerful among all finite dimensional LTI unfalsified systems, i.e.,

$$\mathfrak{B} \in \mathfrak{L}^q \text{ and } D \subseteq \mathfrak{B}|_{[1,T]} \Rightarrow \mathfrak{B}^*|_{[1,T]} \subseteq \mathfrak{B}|_{[1,T]} \tag{2.25}$$

Thus the dynamical system with behavior $\mathfrak{B}^*$ explains the observed signal set and as little else as possible. Hence it has the most predictive power. Note that because of finite dimensionality, MPUM for a data set is always an autonomous system. The main results of the thesis, methods defined in Chapter 3 and Chapter 4, recursively finds a representation for the MPUM for each newly acquired impulse response or arbitrary input/output data for single input single output (SISO) DTLSSs. Then, from the representation of the MPUM, a representation for the kernel representation of the system is derived and the transfer function of the zero state behavior of the system is acquired. For the uniqueness of solution, some assumptions are used.

3. RECURSIVE PARTIAL REALIZATION OF DTLSSs

Switched linear systems are most conveniently described by the state equations (recall the definition of backward shift operator from Section 2.2.1)

$$\begin{aligned}(\sigma x)(k) &= A_{\alpha(k)}x(k) + B_{\alpha(k)}u(k)\\ y(k) &= C_{\alpha(k)}x(k) + D_{\alpha(k)}u(k)\end{aligned} \qquad \textbf{(3.1)}$$

where $u(k)$,$y(k)$ and $x(k)$ are respectively the input, output and state vectors. The trajectories that satisfy these equations comprises the set $\mathfrak{B}$ (Recall the state space representation from Section 2.2.3). The sequence α is the switching signal which takes values from a set $Q = \{1,2,...,q\}$. For a fixed value $\alpha(k) = i$ of the switching signal, the linear system represented by the state-space parameters (A_i,B_i,C_i,D_i) is called a mode of the switched system. Thus the system represented by (**3.1**) comprises q subsystems (modes). The active mode which operates at a time instant is determined by the value of the switching signal at that instant. The time between two consecutive changes of the switching sequence is called the *dwell time*.

By eliminating $x(k)$ from (**3.1**) a mode i of the switched system can alternatively be represented by the kernel representation in input/output form (Recall Ch. 2.2.1 and 2.2.2) by

$$D_i(\sigma)y - N_i(\sigma)u = 0. \qquad \textbf{(3.2)}$$

For $u = 0$ the autonomous system described by the state-space parameters (A_i,C_i) or equivalently by the kernel of $D_i(\sigma)$ is from now on called the zero-input dynamics of mode i. The problem studied in this chapter is defined as follows:

Problem: Partial realization of DTLSSs

Given the impulse response sequence $h(k) = h_k$ for $k = 0,1,...,N$ of the switched system (**3.1**) corresponding to the initial state $x(0) = 0$, input $u(k) = \delta(k)$ and an unknown switching signal α; find the state equation representation or kernel representation of the starting mode and zero-input dynamics of all subsequent modes assuming that the orders n of the modes are known.

Remark 3.0.1. Note that in the partial realization problem only a finite length of the impulse response sequence is given and the solution is highly nonunique. In the case of linear systems a parameterization of all solutions and their possible orders are studied in [4, 10]. In our formulation it is assumed that the orders of the modes are known in order to find a unique solution.

Remark 3.0.2. Notice that for switched systems, after a switch at time instant τ the kernel representation (**3.2**) found for the mode is not actually valid for time interval $[\tau - n, \tau]$. Because since we make use of the backward shift operator to write the kernel representations, in time interval $[\tau - n, \tau]$ data from the subsequent mode must be used and the kernel representation found for the previous mode would not hold. The method studied in this work gives kernel representation (**3.2**) (and therefore the transfer function of the controllable part of the behavior) of the individual local modes as if they are distinct systems. To interpret a DTLSS as a concatenation of different local systems (this is the approach adopted for the purposes of the thesis) is not always very advantagous. For instance, the minimality of a DTLSS realization does not depend on the minimality of all local modes. In addition, the state equation representation (A_i, B_i, C_i, D_i) of the modes can only be determined up to a change of basis (within a similarity transformation) in state space. The problem of writing state equations of the modes in a common coordinate system is not a trivial one and it is considered in [13]. These remarks and problem about state space representations with formal definitions and concepts in partial realization theory for DTLSSs will also be discussed in Chapter 5.

3.1 Identifiability Conditions

A system is identifiable from the observed trajectory $w = (w_0, w_1, ..., w_{N-1})$ if there exists no other system in the given model class which generates the same trajectory. In this case, the trajectory w is called sufficiently rich for the system. In other words, a sufficiently rich trajectory reflects all characteristic features of the system to distinguish it from other systems in the same model class. In [18] it is shown that a controllable system of order n and lag l is identifiable from the trajectory $w = \text{col}(y, u)$ if the input component u is persistently exciting of order $n + l + 1$. This result, however, cannot be directly applied to this problem since some of the modes of the switched system are not driven by inputs, but only the responses due to initial conditions are known. In the following lemma an identifiability condition similar to the one in [18] is given which is suitable in the present situation and which is also necessary for single output systems. In order to state this result let $H_r(w)$ denote the block Hankel matrix of r rows associated with the trajectory $w = (w_0, w_1, ..., w_{N-1})$ which is explicitly defined as

$$H_r(w) = \begin{bmatrix} w_0 & w_1 & \dots & w_{N-r} \\ w_1 & w_2 & \dots & w_{N-r+1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{r-1} & w_r & \dots & w_{N-1} \end{bmatrix} \qquad \mathbf{(3.3)}$$

The term identification in this thesis is used in the meaning "to find the kernel representation of unique, minimal order, controllable model for local modes of a DTLSS". Also for the next lemma the definition of left kernel of a matrice M can be reminded as the nullspace of M^T i.e., the subspace consists of all nonzero row vectors r_i that makes $r_iM = 0$.

Lemma 3.1.1. *Assume that the trajectory $w = (w_0, w_1, ..., w_{N-1})$ is generated by a linear, time-invariant single output system of order n. The system can be identified from w if and only if dim (left ker $H_{n+1}(w)$)= 1.*

Proof. Let the system be described by the kernel representation $R_0w(k) + R_1w(k+1) + ... + R_nw(k+n) = 0$. Clearly $\begin{bmatrix} R_0 & R_1 & ... & R_N \end{bmatrix} \in$ left ker $H_{n+1}(w)$ and the system can be uniquely identified if and only if left ker $H_{n+1}(w)$ is the one dimensional subspace span $\left(\begin{bmatrix} R_0 & R_1 & ... & R_N \end{bmatrix}\right)$. □

In the next lemma the above result is specialized to the identification of the system from impulse response or zero-input response. This is stated in terms of a state equation representation of the system and also the minimum number of samples required to identify the system is given.

Lemma 3.1.2. *Consider a single input-single output, nth order system defined by $(\sigma x)(k) = Ax(k) + Bu(k), y(k) = Cx(k) + Du(k)$. Let $N_{min} = 2n$.*

(i) *The system can be identified from the impulse response sequence $h(k) = h_k$, $k = 0, ..., N-1$; $h(k) = 0$ for $k < 0$ if and only if the system is controllable, observable and*

$N \geq N_{min}+1$ *(impulse response is defined as the response of a system due to an input as a unit impulse at time* $k=0$*).*

(ii) *The zero-input dynamics of the system can be identified from the response* $y(k)=y_k$ *for* $k=0,...,N-1$ *due to the initial state* $x(0)=x_0$ *if and only if the pair* (A,x_0) *is controllable, the system is observable and* $N \geq N_{min}$.

Proof. (i)

Take $z(k) = \text{col}(h(k),\delta(k))$ and note that $z(k)=0$ for $k<0$. We will apply Lemma 3.1.1 to the shifted trajectory defined by $w(k)=0$ for $k=0,...,n-1$ and $w(k)=z(k-n)$ for $k=n,...,N+n-1$ (Notice that the trajectory $w(k)$ is the n times forward shifted version of $z(k)$, the partial impulse response). If we write the Hankel matrix for $w(k)$ we get

$$H_{n+1}(w) = \begin{bmatrix} w(0) & w(1) & ... & w(n+1) & ... & w(N-1) \\ w(1) & w(2) & ... & w(n+2) & ... & w(N) \\ \vdots & \vdots & ... & \vdots & ... & \vdots \\ w(n) & w(n+1) & ... & w(2n+1) & ... & w(N+n-1) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & ... & h_1 & ... & h_{N-n-1} \\ 0 & 0 & ... & 0 & ... & 0 \\ 0 & 0 & ... & h_2 & ... & h_{N-n} \\ 0 & 0 & ... & 0 & ... & 0 \\ 0 & \vdots & ... & 0 & ... & 0 \\ 0 & \vdots & ... & 0 & ... & 0 \\ \vdots & h_0 & ... & h_n & ... & h_{N-2} \\ \vdots & 1 & ... & 0 & ... & 0 \\ h_0 & h_1 & ... & h_{n+1} & ... & h_{N-1} \\ 1 & 0 & ... & 0 & ... & 0 \end{bmatrix}. \quad \textbf{(3.4)}$$

By reordering the rows of the $(2n+2)\times N$ matrix in (**3.4**) we get the form in (**3.5**). Define the four block columns in (**3.5**) in the usual sense as H_{11},H_{12},H_{21} and H_{22}. H_{11} is $(n+1)\times(n+1)$, H_{12} is $(n+1)\times(N-n-1)$, H_{21} is $(n+1)\times(n+1)$ and H_{22} is $(n+1)\times(N-n-1)$. Notice by using the last $n+1$ rows of (**3.5**) we can eliminate H_{11} to get a zero matrix, without changing H_{12}. Again by reordering the rows we get the block matrix seen in (**3.6**).

$$H_{n+1}(w) = \left[\begin{array}{ccccc|cccc} 0 & 0 & \dots & \dots & h_0 & h_1 & \dots & \dots & h_{N-n-1} \\ 0 & 0 & \dots & h_0 & h_1 & h_2 & \dots & \dots & h_{N-n} \\ \vdots & \vdots & & & \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots & \vdots & & & \vdots \\ h_0 & h_1 & \dots & \dots & h_n & h_{n+1} & \dots & \dots & h_{N-1} \\ \hline 0 & 0 & \dots & \dots & 1 & 0 & \dots & \dots & 0 \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & \dots & 0 \\ \vdots & \vdots & & & \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots & \vdots & & & \vdots \\ 1 & 0 & \dots & \dots & 0 & 0 & \dots & \dots & 0 \end{array}\right]. \quad \mathbf{(3.5)}$$

$$H_{n+1}(w) = \left[\begin{array}{c|c} I_{n+1} & 0 \\ \hline 0 & H_{n+1}(\sigma h) \end{array}\right] \quad \mathbf{(3.6)}$$

In (**3.6**), Hankel matrix of the backward shifted impulse response $\sigma h = (h_1, ..., h_{N-1})$ is defined as

$$H_{n+1}(\sigma h) = \begin{bmatrix} h_1 & h_2 & \dots & h_{N-n-1} \\ h_2 & h_3 & \dots & h_{N-n} \\ \vdots & \vdots & & \vdots \\ h_{n+1} & h_{n+2} & \dots & h_{N-1} \end{bmatrix}. \quad \mathbf{(3.7)}$$

From (**3.6**) it can be seen that rank $H_{n+1}(w) = n+1+$ rank $H_{n+1}(\sigma h)$. By Lemma 3.1.1 the system can be identified from w if and only if dim (left ker $H_{n+1}(w)$)$= 1$. This implies the system can be identified from w if and only if rank$H_{n+1}(w) = (2n+2)-1 = 2n+1$. This, in turn, implies rank$H_{n+1}(\sigma h)$ must be n. By defining the controllability matrix as $K = \begin{bmatrix} B & AB & \dots & A^{N-n-2}B \end{bmatrix}$ and the observability matrix as $O = \begin{bmatrix} C & CA & \dots & CA^n \end{bmatrix}^T$, $H_{n+1}(\sigma h)$ can be written as $H_{n+1}(\sigma h) = O_{(n\times n)}K_{(n\times(N-n-1))}$. For $H_{n+1}(\sigma h)$ to be of rank n both O and K must be of rank n (It can be seen from the rank inequality rank$O$+rank$K - n \leq$ rank$(OK) \leq$ min$\{$rank$O$, rank$K\}$). This implies the number of columns of K must be at least n (This, in turn, implies the condition $N \geq 2n+1$) and the system must be controllable and observable.

(ii)

Let us construct the Hankel matrix associated with the zero-input output due to an initial state x_0 as

$$H_{n+1}(y) = \begin{bmatrix} y_0 & y_1 & \cdots & y_{N-n-1} \\ y_1 & y_2 & \cdots & y_{N-n} \\ \vdots & \vdots & \cdots & \vdots \\ y_n & y_{n+1} & \cdots & y_{N-1} \end{bmatrix}_{(n+1)\times(N-n-1)} \tag{3.8}$$

We will again apply Lemma 3.1.1 to $H_{n+1}(y)$. For left ker $H_{n+1}(y)$ to be one dimensional rank of $H_{n+1}(y)$ must be $n+1-1=n$. Note that $H_{n+1}(y)$ can be written as $H_{n+1}(y) = O_{n\times n}X_{n\times(N-n-1)}$ where O is the observability matrix defined in part *(i)* and $X = \begin{bmatrix} x_0 & Ax_0 & \dots & A^{N-n-1}x_0 \end{bmatrix}$. By the rank inequality in part *(i)* for $H_{n+1}(y)$ to be of rank n both O and X must be of rank n. This implies the number of columns of X must be at least n (This, in turn, implies the condition $N \geq 2n$) and the system must be observable and the pair (A, x_0) must be controllable. □

3.2 Recursive Modeling Procedure

The recursive solution of the partial realization problem for switched systems is presented under the following assumptions.

Assumptions

A1. The switched system (**3.1**) has single input, single output. The modes of the system are controllable, observable and the orders n of the modes are known.

A2. The dwell time of the starting mode driven by the impulse is greater than $N_{min} = 2n$ i.e., if the switching instants for the DTLSS are defined as $(\tau_1, \tau_2, \ldots)$, $\tau_1 > N_{min}$.

A3. Dwell time of the subsequent modes whose zero-input responses are observed are at least $2N_{min} - 1 = 4n - 1$ i.e., $\tau_i - \tau_{i-1} \geq 2N_{min} - 1$ for $i = 2, 3, \ldots$

A4. For each mode, there is at least one period in the impulse response sequence which is sufficiently rich for the mode active in the same period i.e., for each mode j of the DTLSS there is at least one switching instant τ_i to that mode such that $(A_j, x(\tau_i))$ is controllable.

It should be noted that the dwell time assumed for the modes is about twice the time required to identify a single nth order system since the zero-input response produced by a mode may coincide with the response of the previous mode and it may take $N_{min}-1$ samples to detect a mode change (since in Lemma 3.1.2 *(ii)*, it is shown that the zero input dynamics of a system is identifiable from its zero input response by using at least N_{min} samples, zero-input response produced by a mode may coincide with the response of the previous mode for at most $N_{min}-1$ samples) and N_{min} samples more to identify after the event is detected. Thus controllability, observability of the modes and the assumption on dwell times are necessary conditions for identifiability. Assumption *A4* which is also necessary is tested in *Step 5* of the recursive procedure, no rank tests are required.

In the partial realization problem, the input/output trajectory to be modeled is $w(k) = \mathrm{col}(h(k), \delta(k))$ where $h(k) = h_k, k = 0,...,N; h(k) = 0, k < 0$. The sequence $w(k)$ is defined for $k \leq N$. In order to work with sequences defined on the nonnegative time axis $\mathbb{N}$ take

$$z(k) = w(N-k),\ k = 0,1,... \tag{3.9}$$

and define the backward shifts of the sequence z by

$$z^i = \sigma^{N-i}z,\ i = 0,1,...,N \tag{3.10}$$

Note that $z^N = z$ and $z^{i-1} = \sigma z^i$ for $i = 1,2,...,N$. It is easily seen that the MPUM of z^i in $\mathfrak{L}^2$ (notice $q = 2$ since we have a SISO system) is $\mathfrak{B}_i^* = \mathrm{span}\{z^0, z^1, ..., z^i\}$. The procedure described below recursively finds a shortest lag kernel representation of $\mathfrak{B}_i^*$ until a mode change is detected. Then the sequences z^i are redefined and the procedure is repeated.

Recursive Procedure

The flowchart of the recursive procedure is depicted in Figure 3.1. The steps of the procedure are then explained in detail.

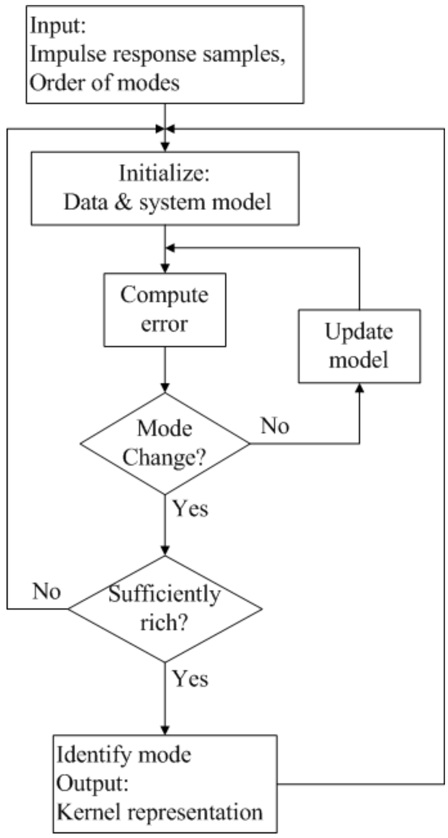

Figure 3.1: Flowchart of the recursive partial realization procedure of DTLSSs

1. Initialization: The time reversed trajectory z and its backward shifts z^i are initially defined by (**3.9**) and (**3.10**). If a mode change is detected when the data sample $w(r)$ is received, let $N_r = N - r + 1$ and redefine the sequences z, z^i as

$$\begin{aligned} z(k) = w(N-k)\ k = 0,1,...,N-r, z(N_r) = \mathrm{col}(0,1), \\ z(k) = 0\ k > N_r,\ \text{and}\ z^i = \sigma^{N_r - i} z\ i = 0,1,...,N_r. \end{aligned} \qquad (\mathbf{3.11})$$

In this way, the remaining data samples are considered as being generated by an impulse applied at time $r-1$ to set up initial conditions (state). Furthermore we initially take $R_{-1} = I$. The following steps are repeated for $k = 0,1,...$

2. Error Computation: Let $R_{k-1}(\sigma)$ be the kernel representation of the MPUM for z^{k-1}. The error at stage k is defined by

$$e^k = R_{k-1}(\sigma) z^k \qquad (\mathbf{3.12})$$

Since $\sigma e^k = R_{k-1}(\sigma)\sigma z^k = R_{k-1}(\sigma) z^{k-1} = 0$ the sequence e^k is a pulse of the form $e^k = (e_0, 0, 0, ...)$ and $e_0 = \mathrm{col}(\Delta_k, \widetilde{\Delta}_k)$.

3. Event Detection: In general, a mode change is detected when the model represented by the first row of $R_k(\sigma)$ is falsified by the recently received data. Because of the assumptions *(A2,A3)* on dwell time the test for event detection need not be performed for $k \leq N_{min} = 2n$. Thus, a mode change is detected if the conditions

$$(i)k > N_{min} \quad (ii)\Delta_k \neq 0 \tag{3.13}$$

are simultaneously satisfied. Under the conditions (**3.13**) $(i-ii)$ the controllable model derived for the sequence z^{k-1} cannot be updated without increasing the order which is an indication of the mode change. Note that an event may be detected before the currently active mode is identified. This happens when the zero-input response is not sufficiently rich for the mode and is indicated by the conditions $k > N_{min}$ and $L_{k-1} < n$ where L_{k-1} denotes the degree of the first row of $R_{k-1}(s)$. After the detection of mode change *Step 4* for the model update is skipped and the procedure proceeds with *Step 5* to identify the mode.

4. Model Update: When the error sequence e^k defined by (**3.12**) is nonzero, the kernel representation $R_{k-1}(\sigma)$ has to be updated to obtain a kernel representation of the MPUM for z^k. Let $V_k(\sigma)$ be a kernel representation of the MPUM for e^k. Then it is easily seen that

$$R_k(\sigma) = V_k(\sigma)R_{k-1}(\sigma) \tag{3.14}$$

is a kernel representation of the MPUM for z^k. As explained in [10] the update matrix $V_k(s)$ is chosen in such a way that $R_k(s)$ is row reduced at each step and the row which does not lose rank at $s = 0$ (the row which gives a controllable model for z^k) is the first row. Let L_{k-1} and $\widetilde{L}_{k-1}$ denote the degrees of the first and second rows of $R_{k-1}(s)$ respectively then the choice

$$V_k(s) = \begin{bmatrix} \widetilde{\Delta}_k & -\Delta_k \\ 0 & s \end{bmatrix} \text{ if } \Delta_k = 0 \text{ or } L_{k-1} > \widetilde{L}_{k-1} \text{ and}$$

$$V_k(s) = \begin{bmatrix} \widetilde{\Delta}_k & -\Delta_k \\ s/\Delta_k & 0 \end{bmatrix} \text{ otherwise}$$

meets the above requirements as shown in [10]. After the model update, the counter k is incremented and the procedure returns to *Step 2* for error computation.

5. Identifiability Test and Identification of the Mode: For $k \geq N_{min} = 2n$ the first row of $R_k(s)$ which does not lose rank at $s = 0$ is the kernel representation of the unique, minimal order, controllable model of the sequence z^k which is entirely generated by a single mode of the switched system. Let $r(s) = \begin{bmatrix} r_1(s) & r_2(s) \end{bmatrix}$ denote the first row of $R_k(s)$ for $k \geq N_{min}$. If the degree of $r(s)$ is equal to the order n of the modes, then the mode is identified. Taking the time reversal operation (**3.9**) into account, the kernel representation of the mode is given by $\tilde{r}(\sigma)w = 0$ and the kernel representation of the zero-input dynamics is $\tilde{r}_1(\sigma)w = 0$ where for a polynomial $p(s)$ of degree n, $\tilde{p}(s)$ is defined as $\tilde{p}(s) = s^n p(s^{-1})$. Under the assumptions *(A1 A4)* the starting mode driven by the impulse is identified as soon as $N_{min} + 1$ data samples are received. For the subsequent modes whose zero-input responses are known, the degree of $r(s)$ may be lower than n. This indicates that the trajectory z^k is not sufficiently rich hence the mode cannot be identified on this visit. The identification of the mode is postponed until a portion of the impulse response sequence which satisfies the condition of Lemma 3.1.2 *(ii)* is received. Assumption *A4* ensures that such a period exists. After the identifiability test and identification of the mode, the flow of the procedure is returned to *Step 1*.

Remark 3.2.1. It can be proven that the condition in Lemma 3.1.2 *(ii)* can be used to guarantee the detection of switching in *Step 3*. The related theorem and its proof are below.

Theorem 3.2.2. *Assume R_1 and R_2 are minimal representations of the zero input dynamics of any two modes (A_1, C_1), (A_2, C_2) of the DTLSS with $R_1 \neq R_2$ and $degR_1 = degR_2 = n$. Also suppose the assumptions* (A1-A3) *are fulfilled for the DTLSS. The switching can always be detected by the recursive algorithm (the conditions in* Step 3 (i) *and* (ii) *are always satisfied) if the there is a switching instant $x(\tau_{i_j})$ ($x(\tau_{i_j})$ shows the ith switching instant to mode j) for each mode j such that $(A_j, x(\tau_{i_j}))$ is controllable and the pair (A_j, C_j) is observable for $j = 2, 3, ...$ and $i \in \{1, 2, ...\}$.*

Proof. Without loss of generality consider any two adjacent modes of the DTLSS. Call the former mode operating as mode 1 and the latter as mode 2. Call n as the order of these modes. Suppose zero-input dynamics of these modes are represented with the distinct polynomials $R_1(s)$ and $R_2(s)$ respectively.

Now suppose $(A_2, x_0^{(2)})$ (with $x_0^{(2)}$ showing the state of the DTLSS at the switching instant to mode 2) is controllable, (A_2, C_2) is observable and after the dwell time assumptions *(A2,A3)* are fulfilled, there is a switch to mode 2 from mode 1. Consider the Hankel matrix of mode 1 $H^{(1)}_{n+1,n+1}$ constructed from the response due to the initial state $x_0^{(1)}$ ($x_0^{(1)}$ shows the state of the DTLSS at the switching instant to mode 1) as

$$H^{(1)}_{n+1,k} = \begin{bmatrix} C_1 x_0^{(1)} & \dots & C_1 A_1^{k-1} x_0^{(1)} \\ \vdots & & \vdots \\ C_1 A_1^{k-1} x_0^{(1)} & \dots & C_1 A_1^{2k-1} x_0^{(1)} \end{bmatrix}. \tag{3.15}$$

If $R_1(s) = s^n + a_{n-1}s^{n-1} + \dots + a_1 s + a_0$, from $R_1(\sigma)y = 0$, it is clear that

$$\begin{bmatrix} a_0 & a_1 & \dots & a_{n-1} & 1 \end{bmatrix} \begin{bmatrix} y(t) \\ \vdots \\ y(t+n) \end{bmatrix} = 0. \tag{3.16}$$

Hence, we can see that

$$\begin{bmatrix} a_0 & a_1 & \dots & a_{n-1} & 1 \end{bmatrix} H_{n+1,k} = 0 \tag{3.17}$$

for $k \geq 1$. Define also $R_2(s) = s^n + b_{n-1}s^{n-1} + \dots + b_1 s + b_0$. Suppose after $2n$ steps from the switch, there is still no error found with the algorithm (i.e. the switching could not be detected). This means both kernel representations are valid for mode 2 i.e.,

$$\begin{bmatrix} a_0 & a_1 & \dots & a_{n-1} & 1 \end{bmatrix} H^{(2)}_{n+1,n+1} = 0 \tag{3.18}$$

$$\begin{bmatrix} b_0 & b_1 & \dots & b_{n-1} & 1 \end{bmatrix} H^{(2)}_{n+1,n+1} = 0 \tag{3.19}$$

where $H^{(2)}_{n+1,n+1}$ is the Hankel matrix constructed with the first $2n+1$ output data after the switching due to the initial state $x_0^{(2)}$ i.e.,

$$H^{(2)}_{n+1,n+1} = \begin{bmatrix} C_2 x_0^{(2)} & \dots & C_2 A_2^{n} x_0^{(2)} \\ \vdots & & \vdots \\ C_2 A_2^{n} x_0^{(2)} & \dots & C_2 A_2^{2n} x_0^{(2)} \end{bmatrix}. \tag{3.20}$$

Substituting (**3.19**) from (**3.18**) yields

$$\begin{bmatrix} a_0 - b_0 & \dots & a_{n-1} - b_{n-1} & 0 \end{bmatrix} H^{(2)}_{n+1,n} = 0 \tag{3.21}$$

which implies

$$\begin{bmatrix} a_0 - b_0 & \dots & a_{n-1} - b_{n-1} \end{bmatrix} H_{n,n}^{(2)} = 0. \quad \textbf{(3.22)}$$

Notice $H_{n,n}^{(2)}$ can be written as

$$H_{n,n}^{(2)} = O^{(2)} X^{(2)} = \begin{bmatrix} C_2 \\ \vdots \\ C_2 A_2^{n-1} \end{bmatrix} \begin{bmatrix} x_0^{(2)} & \dots & A_2^{n-1} x_0^{(2)} \end{bmatrix}. \quad \textbf{(3.23)}$$

We know from the observability of (A_2, C_2) and controllability of $(A_2, x_0^{(2)})$ that $O^{(2)} \in \mathbb{R}^{n \times n}$ is of full column rank and $X^{(2)} \in \mathbb{R}^{n \times n}$ is of full row rank. That means $\text{rank} H_{n,n}^{(2)} = n$. By linear independence of the rows of $H_{n,n}^{(2)}$ we can say $\begin{bmatrix} a_0 - b_0 & \dots & a_{n-1} - b_{n-1} \end{bmatrix} = 0$. This implies

$$a_0 = b_0, \dots, a_{n-1} = b_{n-1} \quad \textbf{(3.24)}$$

Since kernel representation of two distinct modes of the DTLSS must be different, this is a contradiction. Therefore, the proof is complete. □

Remark 3.2.3. Note that after the switch detection, an impulse is applied in the initialization step (*Step 1*) to build the initial state at the instant of switching. Thus, after a switch, the parameters of the system to be identified is (A', B', C') where (A', C') are the original (A, C) matrices of the active mode and B' is the initial state vector x_0 (x_0 is the state of the DTLSS at the switching instant). Since the matrices defining the zero-input dynamics of the original system (A, C) are equal to (A', C'), zero-input dynamics can be correctly identified with the recursive procedure from $(1, 1)$ element of $R(s)$ after N_{min} new data. Note that first row of $R(s)$, $r(s)$, is a kernel representation for this newly "created" system (A', B', C'), thus, it is not the kernel representation of the original system. One cannot derive original B matrix of the system, since only a partial response due to an initial state is known.

3.3 Example

Consider the bimodal switched linear system seen in Figure 3.2. The first mode is defined by the state space parameters

$$A_1 = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}, D_1 = \begin{bmatrix} 1 \end{bmatrix} \quad \textbf{(3.25)}$$

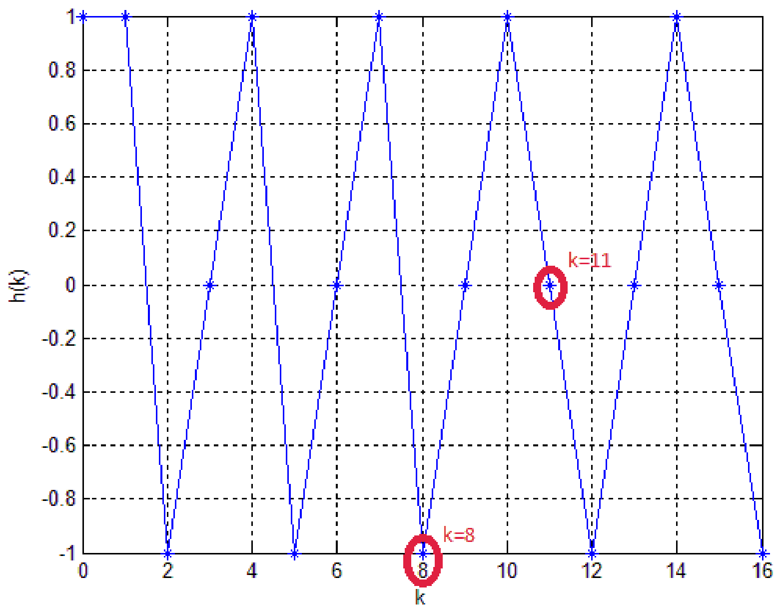

Figure 3.2: Impulse response of the DTLSS for $k=[0,16]$

and the second mode is given by

$$A_2=\begin{bmatrix} -1 & -1 \\ 2 & 1 \end{bmatrix}, B_2=\begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_2=\begin{bmatrix} -1 & 0 \end{bmatrix}, D_2=\begin{bmatrix} 0 \end{bmatrix}. \tag{3.26}$$

The impulse response sequence of total length 17 obtained by starting with the first mode, running mode 1 for $k \in [0,7]$ and then switching to second mode for $k \in [8,16]$ is given as $h=\begin{bmatrix} h_1 & h_2 \end{bmatrix}$ where $h_1=(1,1,-1,0,1,-1,0,1)$ is the sequence produced by mode 1 and $h_2=(-1,0,1,0,-1,0,1,0,-1)$ is the zero-input response due to initial state $x(8)=\mathrm{col}(1,-1)$ generated by mode 2. Figure 3.2 shows the response of the system for $k=[0,16]$ where the instant of switching ($k=8$) and detection of switching ($k=11$) are shown in circles. When the first 5 samples of h are received, the procedure described above gives

$$R_4(s)=\begin{bmatrix} s^2+s+1 & -s^2-2s-1 \\ s^3+s^2+2s & -3s^2-2s \end{bmatrix} \tag{3.27}$$

as the kernel representation of the MPUM for the sequence z^4 defined in (**3.10**). The kernel representation of the first mode is correctly identified from the first row of $R(s)$ as

$$(\sigma^2+\sigma+1)y=(\sigma^2+2\sigma+1)u. \tag{3.28}$$

As the procedure proceeds it is seen that the first component Δ_k of the error sequence (**3.12**) is zero hence the first row of $R_k(s)$ remains the same for $k \in [5,10]$. The mode change is detected at $k = 11$ when $h(11) = 0$ is received. The switching to mode 2 takes place at $k = 8$ but cannot be detected before $k = 11$ since the second mode produces the same response with mode 1 for $k \in [8, 10]$. After the event detection, the sequence is initialized as explained above and

$$R_4(s) = \begin{bmatrix} s^2+1 & s^2 \\ s & s^3 \end{bmatrix} \tag{3.29}$$

is found as the kernel representation of the MPUM for the time-reversed trajectory consisting of the samples $h(11)...h(14)$. From the $(1,1)$ element of $R_4(s)$ the kernel representation of the zero-input dynamics of mode 2 is correctly read as $(\sigma^2+1)y = 0$. It should be noted that if the dwell time of mode 2 were not sufficiently long there would not be enough data samples to uniquely identify the mode. This justifies the assumption *A3* on minimum dwell times.

To illustrate the identifiability condition given in Lemma 3.1.2 *(ii)* and the equivalent condition in *Step 5* of the recursive procedure, suppose that the $(1,1)$ element of the matrix A_2 is changed to -2 everything else remaining the same. Then the zero-input response corresponding to initial state $x(8) = \text{col}(1,-1)$ produced by the second mode is $h_2(k) = (-1)^{k+1}$ for $k \geq 8$. The mode change is detected at $k = 9$ but mode 2 cannot be identified from the available sequence h_2 since the pair $(A_2, x(8))$ is uncontrollable. The condition given in Lemma 3.1.2 *(ii)* is not satisfied. In the recursive procedure this is revealed by the matrix

$$R_4(s) = \begin{bmatrix} s+1 & -s \\ -s^3 & s^4 \end{bmatrix} \tag{3.30}$$

which is obtained after the samples $h_2(k), k \in [9, 12]$ are received. The degree of the first row of $R_4(s)$ is 1 which means that the given trajectory is not sufficiently rich for mode 2 and could also be produced by a first order system.

3.4 A Different Assumption For the Procedure

From the previous sections, it can be seen that there is a possibility that even though the assumptions on dwell times *(A2,A3)* are satisfied, detecting a mode change may not be possible by the procedure. This fact is characterized by Lemma 3.1.2 *(ii)* and the problem solved by assumption *A4* (Note that apart from the condition on dwell times assumptions *(A1,A4)* are mild conditions and are generically satisfied).

However, a different assumption can be made to guarantee the detection of switching. This assumption should guarantee that there is no element in the intersection of each individual mode's behavior set. In this section, such an assumption will be derived and explained. It should be noted that even though the assumption which will be derived in this section guarantees the detection of mode change, but not identifiability. So, it cannot replace Lemma 3.1.2 *(ii)*. First, the problem will be illustrated on an example.

Example 3.4.1. Let $\Sigma = (p, m, n, Q, \{(A_q, B_q, C_q) | q \in Q\}, x_0)$ with $Q = \{1, 2\}$, $n = 2$, $m = 1$, $p = 1$, $x_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$,

$$\begin{aligned} A_1 &= \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \\ A_2 &= \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}. \end{aligned} \tag{3.31}$$

Suppose a unit impulse at time $t = 0$ is applied to this DTLSS. Also suppose the assumptions on the dwell time *(A2,A3)* hold. It can be seen that starting from time $t = 2$ the output of the first mode will always be $y(t) = 1$ and the state will be $x(t) = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$. If a switch to the second mode occurs after the dwell time assumption is fulfilled, it can also be seen that the output and state will still be the same. Thus it will be impossible to detect the switching with the recursive algorithm. Notice also that the pair (A_2, x_0) will always be uncontrollable when $x_0 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$($x_0$ is the initial state of second mode generated by the previous mode). In Chapter 3.3, this problem is solved by assuming that a sufficiently rich period in the impulse response data always exists. In an example like this such a period can never exist, therefore we conclude the second mode of this DTLSS is not identifiable by this algorithm (Note that in this case the second system is not minimal, but the example still illustrates the problem).

Instead of the assumption *A4*, another assumption (which is more restrictive but can be still regarded as generic) can be made to guarantee the switch detection and DTLSSs as in Example 6 are formally excluded from this recursive algorithm's application area. One such solution would be assuming the intersection of the behavior sets of each modes of the DTLSS is the empty set. In the following, a condition on the behavioral equation representations of each mode will be given. This condition guarantees any two modes of the switched system to have no common trajectories in their behavior sets. For this, some known results of polynomial matrix algebra in behavioral context will be used.

Lemma 3.4.2. ***(Smith form, square case)*** *Let $R(s) \in \mathbb{R}^{q\times q}[s]$. There exist unimodular matrices $U(s), V(s) \in \mathbb{R}^{q\times q}[s]$ such that*

1. $U(s)R(s)V(s) = diag(d_1(s), ..., d_q(s))$.

2. There exist (scalar) polynomials $g_i(s)$ such that $d_{i+1}(s) = g_i(s)d_i(s)$, $i = 1, ..., q-1$

Proof. For the proof, see [8]. □

Remark 3.4.3. *(Lemma 3.4.2)* If $R(s)$ is not square, then the Smith form can also be defined. If $R(s)$ is wide $(g < q)$ or if $R(s)$ is tall $(g > q)$, the Smith forms are given by

$$\begin{bmatrix} d_1(s) & & & 0 & \dots & 0 \\ & \ddots & & \vdots & & \vdots \\ & & d_g(s) & 0 & \dots & 0 \end{bmatrix}, \begin{bmatrix} d_1(s) & & \\ & \ddots & \\ & & d_g(s) \\ 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{bmatrix} \tag{3.32}$$

respectively (Blank spaces are all zeros).

Lemma 3.4.4. *Let $\mathfrak{B}(R) := ker R(\sigma)$. $\mathfrak{B}(R) = \{0\}$ if and only if $R(s)$ is unimodular.*

Proof. "If"

We know that for any matrix $\Gamma(s) \in \mathbb{R}^{g'\times g}[s]$, $R_1(s) \in \mathbb{R}^{g\times q}[s]$ and $R_2(s) \in \mathbb{R}^{g'\times q}[s]$ with $g' \leq g$; if $R_2(s) = \Gamma(s)R_1(s)$, $\mathfrak{B}(R_1) \subseteq \mathfrak{B}(R_2)$ (See [4]). If $R(s)$ is unimodular, then $R(s)R^{-1}(s) = I$ implies $\mathfrak{B}(R) \subset \mathfrak{B}(I) = \{0\}$.

"Only if"

Suppose $R(s) \in \mathbb{R}^{g\times q}[s]$, $g \leq q$ is not unimodular. By Lemma 3.4.2, we know that there exist unimodular matrices $U(s) \in \mathbb{R}^{g\times g}[s]$, $V(s) \in \mathbb{R}^{q\times q}[s]$ such that

$$U(s)R(s)V(s) = D(s) \tag{3.33}$$

where $D(s) \in \mathbb{R}^{g\times q}[s]$ is the wide rectangular matrix with one $g \times g$ diagonal submatrix composed of invariant factors of $R(s)$ and one $g \times (q-g)$ submatrix composed of zeros i.e.,

$$\begin{bmatrix} d_1(s) & & & 0 & \dots & 0 \\ & \ddots & & \vdots & & \vdots \\ & & d_g(s) & 0 & \dots & 0 \end{bmatrix}. \tag{3.34}$$

Multiplying (**3.33**) from the left by $U^{-1}(s)$ and from the right by $V^{-1}(s)$ we get $R(s) = U^{-1}(s)D(s)V^{-1}(s)$. Since $U(s)$ and $V(s)$ are unimodular it is clear that their inverses exist and they are again unimodular. We want to prove there exists a nonzero w such that

$$R(\sigma)w = U^{-1}(\sigma)D(\sigma)V^{-1}(\sigma)w = 0 \tag{3.35}$$

Define new variables h_j, $j = 1,...,q$ as $h := V^{-1}(\sigma)w$ where $h = \mathrm{col}(h_1,...,h_q)$. Since left unimodular transformations do not change the behavior (see Lemma 2.2.1.4), the equations $U^{-1}(\sigma)D(\sigma)h = 0$ and $D(\sigma)h = 0$ represent the same behavior. Also, since $R(s)$ is not unimodular, there exist a non-unity polynomial invariant factor $d_i(s)$ for an $i \in \{1,...,g\}$ which has at least one nonzero root. Take this row of the equation and consider

$$d_i(\sigma)h_i = 0. \tag{3.36}$$

Now the proof is reduced to scalar case. It is clear that $d_i(s) = s^n + d_{i_{n-1}}s^{n-1} + ... + d_{i_0}$ is the characteristic equation of the scalar recurrence relation (**3.36**) (Note that (**3.36**) can be written as $h_i(t+n) + d_{i_{n-1}}h_i(t+n-1) + ... + d_{i_0}h_i(t) = 0$). Let $\lambda_k \in \mathbb{C}$, be one root of $d_i(s)$. We know that there exist solutions to this recurrence relation which include the term $c\lambda_k^t$ with c being an arbitrary constant (Notice even when λ_k has complex part i.e., $\lambda_k = \alpha + j\beta$, the term $c\lambda_k^t$ has still a real part which is $\Re\left\{c\sum_{l=0}^{t}\binom{t}{l}\alpha^{t-l}(j\beta)^l\right\}$). Since we know that there is at least one nonzero λ_k, (**3.36**) has a solution with nonzero real part i.e., $h_i(t) \neq 0$. Thus $h(t) \neq 0$, $w(t) \neq 0$ which means $\mathfrak{B}(R) \neq \{0\}$. □

Definition 3.4.5. (g.c.r.d) [16] If three polynomial matrices satisfy the relation: $P(s) = H(s)G(s)$, then $G(s)$ is called a right divisor of $P(s)$, and $P(s)$ is called a left multiple of $G(s)$. A greatest common right divisor (g.c.r.d.) of two polynomial matrices $P(s)$ and $R(s)$ is a common right divisor which is a left multiple of every common right divisor of $P(s)$ and $R(s)$.

Lemma 3.4.6. *[19]Consider $P(s) \in \mathbb{R}^{q\times q}[s]$ and $R(s) \in \mathbb{R}^{g\times q}[s]$. If a unimodular matrix $U(s) \in \mathbb{R}^{(g+q)\times(g+q)}[s]$ and a matrix $T(s) \in \mathbb{R}^{q\times q}[s]$ are such that*

$$U(s)\begin{bmatrix} P(s) \\ R(s) \end{bmatrix} = \begin{bmatrix} T(s) \\ 0 \end{bmatrix} \tag{3.37}$$

then $T(s)$ is a g.c.r.d. of $\{P(s), R(s)\}$.

Proof. Consider the equation given in the statement of the lemma

$$U(s)\begin{bmatrix} P(s) \\ R(s) \end{bmatrix} = \begin{bmatrix} T(s) \\ 0 \end{bmatrix}. \tag{3.38}$$

We can rewrite (**3.38**) as

$$\begin{bmatrix} P(s) \\ R(s) \end{bmatrix} = U^{-1}(s)\begin{bmatrix} T(s) \\ 0 \end{bmatrix} = \begin{bmatrix} L_1(s) & L_2(s) \\ L_3(s) & L_4(s) \end{bmatrix}\begin{bmatrix} T(s) \\ 0 \end{bmatrix} = \begin{bmatrix} L_1(s)T(s) \\ L_3(s)T(s) \end{bmatrix} \tag{3.39}$$

if $U(s)$ is partitioned accordingly. Thus, $P(s) = L_1(s)T(s)$ and $R(s) = L_3(s)T(s)$ implies $T(s)$ is a common right divisor of $\{P(s), R(s)\}$. To show that it is the greatest common right divisor; consider the partition $U(s) = \begin{bmatrix} L_5(s) & L_6(s) \\ L_7(s) & L_8(s) \end{bmatrix}$ so that from (**3.38**) we can write

$$L_5(s)P(s) + L_6(s)R(s) = T(s) \tag{3.40}$$

Consider a common right divisor $X(s)$ of $\{P(s), R(s)\}$ i.e.;

$$\begin{aligned} P(s) &= A(s)X(s) \\ R(s) &= B(s)X(s) \end{aligned} \tag{3.41}$$

where $A(s)$ and $B(s)$ are arbitrary polynomial matrices with suitable dimensions. Substituting (**3.41**) into (**3.40**) yields:

$$L_5(s)A(s)X(s) + L_6(s)B(s)X(s) = T(s). \tag{3.42}$$

We can see that for any common right divisor $X(s)$ of $\{P(s), R(s)\}$, (**3.42**) holds. Since $L_5(s)$ and $L_6(s)$ have the same number of rows, by defining the matrix $L(s) = L_5(s)A(s) + L_6(s)B(s)$, (**3.42**) can be rewritten as $L(s)X(s) = T(s)$. So $T(s)$ is a left multiple of *any* $X(s)$, and that means $T(s)$ is the g.c.r.d. of $\{P(s), R(s)\}$.

□

Below is the main result of this section. This result is derived from the work [4].

Theorem 3.4.7. *Let $R_1(s) \in \mathbb{R}^{g\times q}[s]$ and $R_2(s) \in \mathbb{R}^{g'\times q}[s]$ represent two behavior sets. $\mathfrak{B}(R_1)\cap\mathfrak{B}(R_2) = \{0\}$ if and only if greatest common right divisor(g.c.r.d.) of R_1 and R_2, $\widetilde{R}(s) \in \mathbb{R}^{k\times q}[s]$ is unimodular (If $R_1(s)$ and $R_2(s)$ are scalar, $\mathfrak{B}(R_1)\cap\mathfrak{B}(R_2) = \{0\}$ if and only if any common divisor of the polynomials R_1 and R_2 is constant).*

Proof. Since by Lemma 3.4.6 R_1,R_2 and their g.c.r.d. $\widetilde{R}$ satisfy $U\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} \widetilde{R} \\ 0 \end{pmatrix}$ for some unimodular $U(s)$;

$$\mathfrak{B}(R_1)\cap\mathfrak{B}(R_2) = \mathfrak{B}\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \mathfrak{B}\begin{pmatrix} \widetilde{R} \\ 0 \end{pmatrix} = \mathfrak{B}(\widetilde{R}). \quad \textbf{(3.43)}$$

By Lemma 3.4.4, we know that $\mathfrak{B}(\widetilde{R}) = \{0\}$ if and only if $\widetilde{R}(s)$ is unimodular. So unimodularity of $\widetilde{R}(s)$ implies $\mathfrak{B}(R_1)\cap\mathfrak{B}(R_2) = \{0\}$. □

Corollary 3.4.8. *Suppose the assumptions in Chapter 3.2* (A1-A3) *are fulfilled for the DTLSS. The switching can always be detected by the recursive algorithm given in Chapter 3.2 if kernel representations of the DTLSS's each mode's zero-input dynamics (denominators of each mode's transfer function) are co-prime.*

Proof. The proof is done in a similar way to the proof of Theorem 3.2.2. Again, without loss of generality consider any two adjacent modes of the DTLSS. Call the former mode operating as mode 1 and the latter as mode 2. Call n as the order of these modes. Suppose zero-input dynamics of these modes are represented with the polynomials $R_1(s)$ and $R_2(s)$ respectively.

Now suppose $\mathfrak{B}(R_1)\cap\mathfrak{B}(R_2) = \{0\}$ and after the dwell time assumptions *(A2,A3)* are fulfilled, there is a switch to mode 2 from mode 1. Suppose after $2n$ steps from the switch, there is still no error found with the algorithm (i.e. the switching could not be detected). Consider the Hankel matrix $H_{n+1,n+1}$ constructed by the impulse response data as

$$H_{n+1,n+1} = \begin{bmatrix} y(0) & \dots & y(n) \\ \vdots & & \vdots \\ y(n) & \dots & y(2n) \end{bmatrix}. \quad \textbf{(3.44)}$$

If $R_1(s) = s^n + a_{n-1}s^{n-1} + \dots + a_1 s + a_0$, from $R_1(\sigma)y = 0$, it is clear that

$$\begin{bmatrix} a_0 & a_1 & \dots & a_{n-1} & 1 \end{bmatrix} \begin{bmatrix} y(t) \\ \vdots \\ y(t+n) \end{bmatrix} = 0. \quad \textbf{(3.45)}$$

Hence, we can see that

$$\left[\begin{array}{ccccc} a_0 & a_1 & \dots & a_{n-1} & 1 \end{array}\right] H_{n+1,k} = 0 \quad \textbf{(3.46)}$$

for all $k \geq n+1$. Since $H_{n+1,n+1}$ is constructed from the impulse response samples of a minimal realization, a new column added to $H_{n+1,n}$ would be linearly dependent on the first n columns. This implies (**3.46**) holds for all $k \geq 1$ (Notice since first n columns comprise the basis for the column space of H, $\mathrm{Im}H_{n+1,k} \subseteq \mathrm{Im}H_{n+1,n+1}$).

Suppose also $R_2(s) = s^n + b_{n-1}s^{n-1} + \dots + b_1 s + b_0$. By our assumption, if there is no error after $2n$ steps (after the switch), that means the kernel representation $R_1(\sigma)y = 0$ is valid for new data which implies

$$\left[\begin{array}{ccccc} b_0 & b_1 & \dots & b_{n-1} & 1 \end{array}\right] H_{n+1,n+2} = 0 \quad \textbf{(3.47)}$$

also holds. Since any new column added to $H_{n+1,n+2}$ would be linearly dependent on first $n+1$ columns, $\left[\begin{array}{ccccc} b_0 & b_1 & \dots & b_{n-1} & 1 \end{array}\right] H_{n+1,k} = 0$ actually holds for all $k \geq 1$. This implies kernel representations of both modes are valid for $k \geq 1$ i.e., $\mathfrak{B}(R_1) \cap \mathfrak{B}(R_2) \neq \{0\}$. This contradicts our initial assumption. Therefore the proof is complete.

□

Theorem 3.4.7 and Corollary 3.4.8 provides a formal assumption on the domain of recursive algorithm's application area. Therefore, assuming "the intersection of zero-input behaviors of each mode of the DTLSS is the set $\{0\}$" will guarantee not to encounter problems as in Example 3.4.1. However, even though we can always find a relation that represents the data, note that for identifiability, condition in Lemma 3.1.2 *(ii)* must be satisfied.

4. RECURSIVE IDENTIFICATION OF DTLSSs FROM ARBITRARY INPUT/OUTPUT SEQUENCES

The aim of first part of this chapter is to present the necessary conditions for a general discrete-time linear system to be identifiable from a measured input/output sequence of it. Then the problem of recursively identifying the local modes of a DTLSS will be stated. A recursive method for the identification of a discrete time linear system from measured input/output data is given in [4] . The method is mainly presented in [4] as a solution to the continuous time polynomial-exponential time series modeling problem. Then, the connection to the problem of identification of a discrete time linear system from measured input/output data is given. In the problem statement section, we discuss how to reformulate the problem of identification of a DTLSS to use this recursive procedure. In the third part, the modified recursive procedure for identification of the local modes of a DTLSS will be stated. Finally, the chapter ends with an example to illustrate the procedure.

4.1 Identifiability Conditions

The material presented in this section can be found in [17] and [18]. In this section, sufficient conditions for identifiability of a general discrete time linear system from arbitrary input/output measurements, will be given. It turns out, by reformulating the problem (how to do it will be explained in Section 4.2) these sufficient conditions can be used for identifiability of the local modes of a DTLSS from arbitrary input/output measurements.

First, for the purposes of this section let us define the Hankel matrix of L block rows as in (**3.3**) associated with the trajectory $w = (w_0, w_1, ..., w_{N-1})$ as

$$H_L(w) = \begin{bmatrix} w_0 & w_1 & \cdots & w_{N-L} \\ w_1 & w_2 & \cdots & w_{N-L+1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{L-1} & w_L & \cdots & w_{N-1} \end{bmatrix} \qquad \textbf{(4.1)}$$

The following definition is important for stating the identifiability conditions.

Definition 4.1.1. (Persistency of excitation) The time series $u = (u(0), u(1), ..., u(N-1))$ is persistently exciting of order L if the Hankel matrix $H_L(u)$ is of full row rank.

Lemma 4.1.2. *[18]Let*

1. $w = col(u,y)$ be an N samples long trajectory of the LTI system $\mathfrak{B}$, i.e.,

$$w = \begin{bmatrix} u \\ y \end{bmatrix} = \left(\begin{bmatrix} u(0) \\ y(0) \end{bmatrix}, ..., \begin{bmatrix} u(N-1) \\ y(N-1) \end{bmatrix} \right) \in \mathfrak{B}|_{[0,N-1]}; \tag{4.2}$$

2. the system $\mathfrak{B}$ be controllable and

3. the input sequence u be persistently exciting of order $L+n$

Then any L samples long trajectory $w = col(u,y)$ of $\mathfrak{B}$ can be written as a linear combination of the columns of $H_L(w)$, and any linear combination $H_L(w)g$, $g \in \mathbb{R}^{N-L+1}$ is a trajectory of $\mathfrak{B}$, i.e.,

$$col\ span\ (H_L(w)) = \mathfrak{B}|_{[0,N-1]}. \tag{4.3}$$

Proof. For a proof, see [18]. □

In [17] it is stated that for sufficiently large L, namely $L \geq l+1$, Lemma 4.1.2 answers the identifiability question.

Theorem 4.1.3. *The system $\mathfrak{B} \in \mathfrak{L}^q$ is identifiable from the exact data $w = col(u,y) \in \mathfrak{B}$ if $\mathfrak{B}$ is controllable and u is persistently exciting of order $l+n+1$.*

Note that applying Theorem 4.1.3 to a single input single output LTI system of order n, it can be said that minimum number of sufficiently rich samples required for guaranteeing the identification of the system is $n+l+n+l+1 = 2n+2l+1$. For systems whose order n are equal to their lag l, this number is from now on defined as $N'_{min} = 4n+1$. Note that the condition in Theorem 4.1.3 is just a sufficient one, and the identification of the system may occur before N'_{min} number of samples.

4.2 Problem Statement

In this section, how to reformulate the problem of recursive identificiation of local modes of DTLSS for making it possible to use the procedure given in [4] will be explained. For this purpose, without loss of generality consider any local mode of a SISO DTLSS represented as

$$\begin{aligned}(\sigma x)(k) &= Ax(k) + Bu(k)\\ y(k) &= Cx(k) + Du(k)\end{aligned} \tag{4.4}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$, $C \in \mathbb{R}^{1\times n}$, $D \in \mathbb{R}^{1\times 1}$, $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}$, $y(k) \in \mathbb{R}$ and the initial state of the mode is defined as $x(0) = x_0$. Recall from Chapter 3 that the kernel representation of one mode can be written as,

$$\begin{bmatrix} D(\sigma) & -N(\sigma) \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} = 0. \tag{4.5}$$

Also, recall from Definition 2.2.2.4 that the transfer function of the zero state behavior of this mode can be acquired from $H(s) = D^{-1}(s)N(s)$. It can also be found by $H(s) = C(sI - A)^{-1}B + D$. Now consider the system with zero initial condition defined as

$$\begin{aligned}(\sigma x)(k) &= Ax(k) + \widetilde{B}u'(k)\\ \widetilde{y}(k) &= Cx(k) + \widetilde{D}u'(k)\end{aligned} \tag{4.6}$$

where the new matrix $\widetilde{B}$ is created with adding a new column to B which is equal to x_0, $\widetilde{D}$ is created with adding a zero column to D and $\widetilde{u}(k)$ consists of two inputs with the new input $v(k)$ being an impulse i.e., $\widetilde{B} = \begin{bmatrix} B & x_0 \end{bmatrix}$, $\widetilde{D} = \begin{bmatrix} D & 0 \end{bmatrix}$ and $u'(k) = \mathrm{col}(\widetilde{u}, v)$ where $\widetilde{u} = (0, u(0), u(1), \ldots)$ and $v = \delta = (v(0), v(1), v(2), \ldots) = (1, 0, 0, \ldots)$. Note that if the response of the system (**4.4**) to a specific sequence of arbitrary inputs $u = (u(0), u(1), \ldots)$ is $y = (y(0), y(1), \ldots)$; the response of (**4.6**) to the sequence of inputs $u' = (u'(0), u'(1), u'(2), \ldots) = \mathrm{col}(\widetilde{u}, v) = \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} u(0) \\ 0 \end{bmatrix}, \begin{bmatrix} u(1) \\ 0 \end{bmatrix}, \ldots \right)$ is $\widetilde{y} = (\widetilde{y}(0), \widetilde{y}(1), \widetilde{y}(2), \ldots) = (0, y(0), y(1), \ldots)$. That means the system (**4.6**) produces the same response with (**4.4**) preceded by a zero. This is the idea similar to the one stated in Remark 3.2.3 to construct the initial state of the DTLSS at the switching instant. By this reformulation, and by constructing an initialization step similar to the one in Chapter 3.2, the problem is reduced to recursively identifying a discrete time linear system from the observed sequence $w \in (\mathbb{R}^3)^{\mathbb{T}}$ of length N. Observe that if we define the kernel representation of this newly "created" system (**4.6**) as

$$\begin{bmatrix} D(\sigma) & -N_1(\sigma) & -N_2(\sigma) \end{bmatrix} \begin{bmatrix} \widetilde{y} \\ \widetilde{u} \\ v \end{bmatrix} = 0, \tag{4.7}$$

$(1,1)$ element of the 1×2 transfer matrix $D^{-1}(s)\begin{bmatrix} N_1(s) & N_2(s)\end{bmatrix}$ is the transfer function of the zero state behavior of the original mode $D^{-1}(s)N(s)$.

Now, to show that this reformulation does not change the identifiability conditions given in Theorem 4.1.3, following fact is stated as a theorem.

Theorem 4.2.1. *The system $\mathfrak{B}$ represented by* (**4.4**) *is identifiable from the data sequence $w = col(y,u)$ if and only if the system $\widetilde{\mathfrak{B}}$ represented by* (**4.6**) *is also identifiable from the data sequence $\widetilde{w} = col(\widetilde{y},\widetilde{u},v)$.*

Proof. Recall from Chapter 3 Lemma 3.1.1 and how the block Hankel matrix of L block rows associated with a partial data sequence w, $H_L(w)$ is defined in (**4.1**). By making use of Lemma 3.1.1 we need to prove that if dim left ker $H_{n+1}(w)$ is 1, dim left ker $H_{n+1}(\widetilde{w})$ is also 1. Since $H_{n+1}(w)$ and $H_{n+1}(\widetilde{w})$ has $2n+2$ and $3n+3$ rows respectively, this is equivalent to stating

$$\text{rank } H_{n+1}(w) = 2n+1 \Rightarrow \text{rank } H_{n+1}(\widetilde{w}) = 3n+2. \tag{4.8}$$

Consider

$$H_{n+1}(w) = \begin{bmatrix} w_0 & w_1 & \dots & w_{N-n-1} \\ \vdots & \vdots & \dots & \vdots \\ w_n & w_{n+1} & \dots & w_{N-1} \end{bmatrix} = \begin{bmatrix} y_0 & y_1 & \dots & y_{N-n-1} \\ u_0 & u_1 & \dots & u_{N-n-1} \\ \vdots & \vdots & \dots & \vdots \\ y_n & y_{n+1} & \dots & y_{N-1} \\ u_n & u_{n+1} & \dots & u_{N-1} \end{bmatrix} \tag{4.9}$$

and suppose rank $H_{n+1}(w) = 2n+1$. Now consider the Hankel matrix associated with the n times forward shifted trajectory $\widetilde{w}$ as in the proof of Lemma 3.1.2 as

$$H_{n+1}(\widetilde{w}) = \begin{bmatrix} 0 & 0 & \dots & \dots & \dots & y_0 & y_1 & \dots & y_{N-n-1} \\ 0 & 0 & \dots & \dots & \dots & u_0 & u_1 & \dots & u_{N-n-1} \\ 0 & 0 & \dots & \dots & \dots & 1 & 0 & \dots & 0 \\ \vdots & \vdots & & & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & & & \vdots & \vdots & & \vdots \\ & y_0 & \dots & \dots & \dots & & & & \\ & u_0 & \dots & \dots & \dots & & & & \\ & 1 & \dots & \dots & \dots & & & & \\ y_0 & y_1 & \dots & \dots & \dots & y_n & y_{n+1} & \dots & y_{N-1} \\ u_0 & u_1 & \dots & \dots & \dots & u_n & u_{n+1} & \dots & u_{N-1} \\ 1 & 0 & \dots & \dots & \dots & 0 & 0 & \dots & 0 \end{bmatrix} \tag{4.10}$$

Notice that by elementary row operations matrix $H_{n+1}(\widetilde{w})$ can be rewritten as

$$H_{n+1}(\widetilde{w}) = \left[\begin{array}{c|c} I_{n+1} & 0 \\ \hline 0 & H_{n+1}(w) \end{array}\right]. \tag{4.11}$$

Since rank $I_{n+1} = n+1$ and rank $H_{n+1}(w) = 2n+1$, rank $H_{n+1}(\widetilde{w}) = (n+1)+(2n+1) = 3n+2$. Therefore the proof is complete. □

4.3 Recursive Modeling Procedure

The recursive solution of the identification problem for DTLSSs from arbitrary input/output sequences is presented under the following assumptions.

Assumptions

A1'. The switched system (**3.1**) has single input, single output. The modes of the system are controllable, observable and the orders n of the modes are known.

A2'. The dwell time of the starting mode driven by an initial state and arbitrary inputs is at least $N'_{min} = 4n+1$ i.e., if the switching instants for the DTLSS are defined as $(\tau_1, \tau_2, ...)$, $\tau_1 > N'_{min}$.

A3'. Dwell times of the subsequent modes whose responses due to arbitrary inputs are observed are at least $2N'_{min} - 1 = 8n+1$ i.e., $\tau_i - \tau_{i-1} \geq 2N_{min} - 1$ for $i = 2, 3, ...$

A4'. Every window of length N'_{min} in the input sequence $u(t)$ of the DTLSS is persistently exciting of order $2n+1$, i.e., the Hankel matrix $H_{2n+1}(u)$ constructed by $u|_{[t_0,t']}$ where $t_0, t' \in \mathbb{N}$ and $t' - t_0 \geq N'_{min}$ is always of full row rank.

It should be noted that the dwell time assumed for the modes is again about twice the time required to identify a single nth order system since the response produced by a mode may coincide with the response of the previous mode and it may take $N'_{min} - 1$ samples to detect a mode change (since in Chapter 4.1, it is stated that the a system is guaranteed to be identifiable from its response due to arbitrary inputs by using at least N'_{min} samples, the response of a mode may coincide with the response of the previous mode for at most $N'_{min} - 1$ samples) and N'_{min} samples more to identify after the event is detected. Thus controllability of the modes, the assumption on dwell times and the assumption *A4'* are sufficient conditions for identifiability. Note that even though the assumption *A4'* seems restrictive, it is necessary for identification and a random input signal would generically satisfy this assumption. By adopting these assumptions it is not needed to check for identifiability in the recursive procedure. Nevertheless, in an example at the end of the chapter, the implication of an input signal not satisfying *A4'* on the recursive procedure will be illustrated.

Also note that the partial realization problem explained in Chapter 3 can be considered as a special case of problem at hand. In the thesis, it is explained first since making the proofs of identifiability conditions for the partial realization problem is relatively simpler because the DTLSS has zero inputs for $t < 0$ (the initial state is zero).

In the recursive procedure for the problem at hand, the input/output trajectory originally aimed to be modeled is $w'(k) = \mathrm{col}(y(k), u(k))$ where $y(k) = y_k, k = 0, \ldots, N; y(k) = 0, k < 0$ and $u(k) = u_k, k = 0, \ldots, N; u(k) = 0, k < 0$. In this problem, we assume that the initial state of the active mode can always be nonzero i.e., past inputs for $k < 0$ are represented with an initial state $x(0) = x_0$. However, as it is explained in the previous section, this initial state will be constructed with an additional impulse input. Therefore, the data to be modeled is converted into a prepended and augmented version of $w'(k)$. Also, by taking the time reversal operation into account the modified data is defined as

$$
\begin{aligned}
w^k &= \left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \ldots \right) \text{ for } k = 0 \\
w^k &= \left(\begin{bmatrix} y_{k-1} \\ u_{k-1} \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} y_0 \\ u_0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \ldots \right) \text{ for } k = 1, \ldots, N+1.
\end{aligned}
\qquad (\mathbf{4.12})
$$

We also need to define an integer $p = 0$ initially for further use in the procedure.

Recursive Procedure

The flowchart of the recursive procedure is depicted in Figure 4.1. The steps of the procedure are then explained in detail.

1. Initialization: The time reversed trajectory is initially defined by (**4.12**). If a mode change is detected in the time instant $k = p'$, redefine $k = 0$, $p = p + p'$ and define the new trajectory at each step as

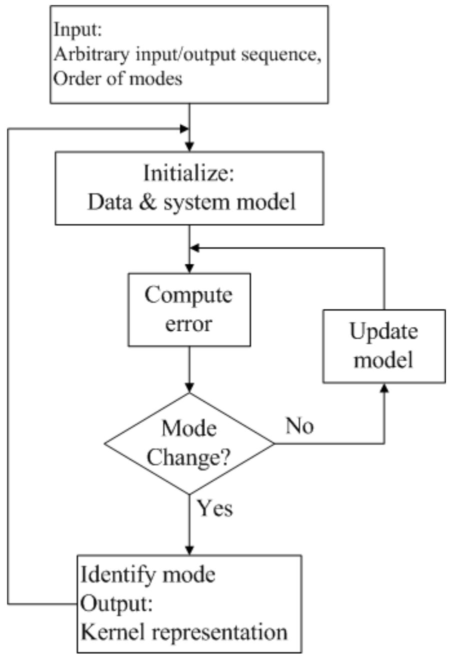

Figure 4.1: Flowchart of the recursive identification procedure of DTLSSs from arbitrary input/output sequences

$$\begin{aligned} w^k &= \left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \ldots \right) \text{ for } k = 0 \\ w^k &= \left(\begin{bmatrix} y_{p+k-1} \\ u_{p+k-1} \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} y_p \\ u_p \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \ldots \right) \text{ for } k = 1, \ldots, N-p. \end{aligned} \tag{4.13}$$

Furthermore, initially take $R_{-1}(s) = I$. The following steps are repeated for $k = 0, 1, \ldots$

2. *Error Computation:* Let $R_{k-1}(\sigma)$ be the kernel representation of the MPUM for w^{k-1}. The error at stage k is defined by

$$e^k = R_{k-1}(\sigma) w^k. \tag{4.14}$$

As in partial realization problem, the sequence e^k is again in the simple form $e^k = \left(\begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_q \end{bmatrix}, \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, \ldots \right)$. To give an explicit formula for error computation similar to the one given in [4], consider the sequence w^k is defined as $w^k = \left(d_{k-1}, d_{k-2}, \ldots, d_0, d_{-1}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \ldots \right)$ for $k = 0, 1, \ldots$ where $d_{-1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$. The first element ε_k of the error sequence e^k at each step is given by

$$\begin{aligned} \varepsilon_k &= R_{k-1}(0)d_{k-1} \text{ for } k=0 \\ \varepsilon_k &= R_{k-1}(0)d_{k-1} + \sum_{j=1}^{k} \frac{R_{k-1}^{(j)}(0)}{j!} d_{k-1-j} \text{ for } k=1,2,... \end{aligned} \quad \textbf{(4.15)}$$

where $R_{k-1}^{(j)}(0)$ denotes the jth derivative of $R_{k-1}(s)$ at $s=0$.

Next, preprocess $R_{k-1}(s)$ such that only last element of the error corresponding to the rows of same degree in $R_{k-1}(s)$ is nonzero i.e., find a $q \times q$ matrix P representing elementary row operations such that:

$$R_{k-1}(s) = PR_{k-1}(s) \quad \textbf{(4.16)}$$

and only last element of the error corresponding to the rows of same degree in $R_{k-1}(s)$ is nonzero. Then normalize the error such that its first nonzero element is 1, i.e. it has the shape:

$$\hat{\varepsilon}_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \hat{\varepsilon}_k \end{bmatrix} \quad \textbf{(4.17)}$$

3. Event Detection: This step is very similar to the corresponding one in the partial realization problem. In general, a mode change is detected when the model represented by the first row of $R_k(\sigma)$ is falsified by the recently received data. Because of the assumptions *(A2,A3)* on dwell time the test for event detection need not be performed for $k \leq N'_{min} = 4n+1$. Thus, a mode change is detected if the conditions

$$(i) k > N_{min} \quad (ii) \varepsilon_1 \neq 0 \quad \textbf{(4.18)}$$

are simultaneously satisfied. Under the conditions (**4.18**) $(i-ii)$ the controllable model derived for the sequence w^{k-1} cannot be updated without increasing the order which is an indication of the mode change. Note that if an event is detected before the currently active mode is identified, it means that the assumption *A4'* does not hold. This happens when the input is not persistently exciting for the mode and is indicated by the conditions $k > N'_{min}$ and $L'_{k-1} < n$ where L'_{k-1} denotes the degree of the first row of $R_{k-1}(s)$. After the detection of mode change *Step 4* for the model update is skipped and the procedure proceeds with *Step 5* to identify the mode.

4. Model Update: Find a kernel representation for the MPUM of the error. Let r be the number showing the index of the first element of the error equal to 1. Define the kernel representation of error as the update matrix $V_k(s)$ where

$$V_k(s) = \begin{bmatrix} I_{r-1} & 0 & 0 \\ 0 & s & 0 \\ 0 & -\hat{\varepsilon}_k & I_{q-r} \end{bmatrix}. \tag{4.19}$$

Then, update the kernel representation of the MPUM, i.e., the matrix $R_k(\sigma)$ in accordance with the equation:

$$R_k(\sigma) = V_k(\sigma)R_{k-1}(\sigma). \tag{4.20}$$

Finally reorder the rows of $R_k(\sigma)$ such that the row degrees are ascending from top to the bottom. Note that by the preprocessing operation defined in *Step 2* and by reordering the rows such that the row degrees are ascending from top to the bottom, the update matrix defined by (**4.19**) does not change the row reducedness of $R_k(\sigma)$, i.e., resulting $R_k(\sigma)$ is always row reduced. This guarantees that the representation for the MPUM acquired at each step is always the shortest lag representation.

Step 5. Identification of the mode: For $k \geq N'_{min} = 4n+1$ the first row of $R_k(s)$ which does not lose rank at $s = 0$ and which has the least order among all rows is the kernel representation of the unique, minimal order, controllable model of the sequence w^k which is generated by the modified single mode of the switched system. Let $\begin{bmatrix} D(s) & -N_1(s) & -N_2(s) \end{bmatrix}$ denote the first row of $R_k(s)$ for $k \geq N'_{min}$. If the degree of $r(s)$ is equal to the order n of the modes and the degrees of other two rows are strictly greater than n, then the mode is identified. Taking the time reversal operation (**4.12**) into account, the kernel representation of the mode is given by $\tilde{r}(\sigma)w = 0$ where for a polynomial $p(s)$ in a row of $R_k(s)$ of degree n, $\tilde{p}(s)$ is defined as

$$\tilde{p}(s) = s^n p(s^{-1}). \tag{4.21}$$

In addition, if we take the reciprocal of this row as defined in (**4.21**) and define the new row as $\begin{bmatrix} \tilde{D}(s) & -\tilde{N}_1(s) & -\tilde{N}_2(s) \end{bmatrix}$, as explained in the previous section $(1,1)$ element of the 1×2 transfer matrix $D^{-1}(s)\begin{bmatrix} N_1(s) & N_2(s) \end{bmatrix}$ is the transfer function of the zero state behavior of the original mode i.e.,

$$H(s) = D^{-1}(s)N_1(s). \tag{4.22}$$

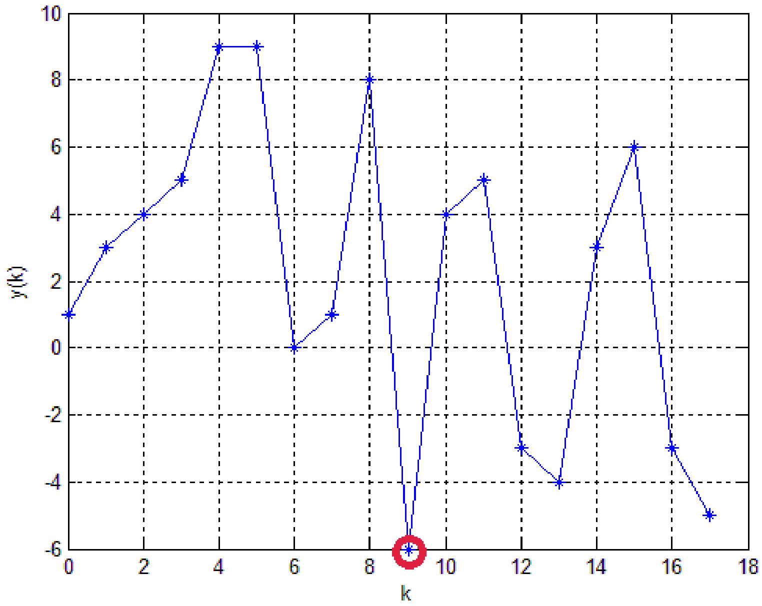

Figure 4.2: Response of the DTLSS for a sequence of arbitrary inputs for $k = [0, 17]$

Under the assumptions *(A1'-A4')* the modes driven by persistently exciting inputs are identified after at most N'_{min} data samples are received. After the identification of the mode, the flow of the procedure is returned to *Step 1*.

4.4 Example

Consider again the bimodal DTLSS example used in Chapter 3.4 with the data as seen in Figure 4.2. The two modes of the DTLSS are represented by (**3.25**) and (**3.26**) respectively.

The output sequence of total length 18 obtained in response to the arbitrary input sequence u. Starting with the first mode, running mode 1 for $k \in [0, 8]$ and then switching to the second mode for $k \in [9, 17]$ corresponding input sequence for the DTLSS is $u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}$ where $u_1 = (1, 2, 3, 4, 7, 5, 1, 3, 2)$ and $u_2 = (-1, 1, 1, 0, 2, 0, 1, -1, 5)$. Corresponding output sequence is given by $y = \begin{bmatrix} y_1 & y_2 \end{bmatrix}$ where $y_1 = (1, 3, 4, 5, 9, 9, 0, 1, 8)$ and $y_2 = (-6, 4, 5, -3, -4, 3, 6, -3, -5)$. In u and y parts of the sequences with same indices are showing the inputs and outputs of the corresponding modes (See Figure 4.2 for the response of this DTLSS). Suppose that the initial condition for the first mode was zero (note that the algorithm could also be used for nonzero initial state with no difference).

After 8 steps of the recursive algorithm, at $k=7$ (recall that we always start with one additional impulse input), i.e., after receiving

$$w(7)=\left(\begin{bmatrix}0\\1\\0\end{bmatrix},\begin{bmatrix}9\\5\\0\end{bmatrix},\begin{bmatrix}9\\7\\0\end{bmatrix},\begin{bmatrix}5\\4\\0\end{bmatrix},\begin{bmatrix}4\\3\\0\end{bmatrix},\begin{bmatrix}3\\2\\0\end{bmatrix},\begin{bmatrix}1\\1\\0\end{bmatrix},\begin{bmatrix}0\\0\\1\end{bmatrix},\begin{bmatrix}0\\0\\0\end{bmatrix},\dots\right) \tag{4.23}$$

the procedure gives

$$R_7(s)=\begin{bmatrix}-s^2-s-1 & s^2+2s+1 & 0\\ -0.125s^3+0.75s^2+0.125s & 1.125s^3-1.5s^2-0.25s & 0.875s^3+0.125s^2\\ s^3+1.8261s^2+0.0435s+0.1739 & -s^3-2.4348s^2-0.087s-0.3478 & s^3+0.2174s^2+0.1739s\end{bmatrix} \tag{4.24}$$

as the kernel representation of the MPUM for the sequence $w(7)$. By taking the reciprocal of the first row as defined in the procedure we get the "augmented" kernel representation of mode 1 as

$$\begin{bmatrix}\widetilde{D}(\sigma) & -\widetilde{N}_1(\sigma) & -\widetilde{N}_2(\sigma)\end{bmatrix}\begin{bmatrix}y\\u\\v\end{bmatrix}=\begin{bmatrix}-\sigma^2-\sigma-1 & \sigma^2+2\sigma+1 & 0\end{bmatrix}\begin{bmatrix}y\\u\\v\end{bmatrix}=0. \tag{4.25}$$

As explained in the previous two sections, the transfer function of the original mode 1 is correctly obtained from the $(1,1)$ element of the acquired transfer matrix from (**4.25**) as

$$H(z)=\widetilde{D}^{-1}(z)\widetilde{N}_1(z)=\frac{z^2+2z+1}{z^2+z+1}. \tag{4.26}$$

For the remaining data from mode 1, first element of the error vector is found to be zero and no change is made in the first row of $R(s)$. After receiving the first data from the second mode at $k=9$, the conditions in *Step 3* of the recursive procedure are satisfied and the switch is detected as soon as it occurs. Then, the procedure is initialized as described in *Step 1* and again after using 7 data points from mode 2, i.e., for

$$\text{w(7)=}\left(\begin{bmatrix}6\\1\\0\end{bmatrix},\begin{bmatrix}3\\0\\0\end{bmatrix},\begin{bmatrix}-4\\2\\0\end{bmatrix},\begin{bmatrix}-3\\0\\0\end{bmatrix},\begin{bmatrix}5\\1\\0\end{bmatrix},\begin{bmatrix}4\\1\\0\end{bmatrix},\begin{bmatrix}-6\\-1\\0\end{bmatrix},\begin{bmatrix}0\\0\\1\end{bmatrix},\begin{bmatrix}0\\0\\0\end{bmatrix},\dots\right) \tag{4.27}$$

the recursive procedure gives

$$R_7(s)=\begin{bmatrix}s^2+1 & -s^2 & -4s^2+6s\\ 0.0173s^2+0.2491s & s^3-0.3391s^2+0.0242s & -1.2561s^3+1.519s^2\\ -2.8571s^2-0.0572s-2.8804 & 3.1527s^2-0.0795s-0.28 & s^3+11.3792s^2-17.5624s\end{bmatrix} \tag{4.28}$$

Again, by applying the same procedure we get the correct transfer function of zero state behavior of mode 2 as

$$H(z) = \widetilde{D}^{-1}(z)\widetilde{N}_1(z) = \frac{1}{z^2+1}. \tag{4.29}$$

Notice, different from the partial realization problem, since we could use the responses due to persistently exciting inputs, we could identify both modes' complete input/outpur behavior.

Now we will illustrate the case when the input trajectory is not persistently exciting of order $2n+1=5$. Consider the first mode only and take the input sequence $u = (1,1,1,1,1,1,1,1,1)$. Notice the length of the input sequence is $9 \geq N'_{min}$ but clearly sequence u is not persistently exciting for the mode. After 5th step of the procedure, it gives the first row of $R(s)$ as

$$r(s) = \begin{bmatrix} 0 & s-1 & s \end{bmatrix}. \tag{4.30}$$

For subsequent data, first element of error is always zero and no update is done in $r(s)$ so the mode cannot be identified. This is seen in the recursive procedure when $k \geq N'_{min}$ and the order of $r(s)$ is still equal to 1 which is lesser than the order $n=2$ of the mode assumed to be known a priori.

5. PROBLEM OF CONSTRUCTING A CONSISTENT STATE SPACE REPRESENTATION

Constructing state space representations for the modes from the acquired kernel representations with the procedures described in Chapters 3 and 4 is not entirely trivial. More precisely, the state space realization of each different mode of the switched system may not generate the observed input-output data. This is due to the fact that the state of the active mode in the exact moment of switching, acts as the initial state of the subsequent mode of the switched system. Since one can use an arbitrary realization algorithm for one mode, the state trajectory may not be the same for each realization. For linear case, input-output behavior does not change under similarity transformations. However, for switched systems, a similarity transformation on one arbitrary mode also changes the state trajectory of that mode. This changes the initial state of the subsequent mode. Thus it changes the input-output behavior of the switched system. This problem can be fixed by slightly modifying the statements of results. In future research, it may also be possible to characterize all possible state space realizations which are consistent with the kernel representations and which would generate the observed data.

In the following, this point is stated clearly using an example.

Example 5.0.1. Consider again the bimodal switched linear system used as an example in both Chapter 3 and 4 given by state space parameters

$$A_1 = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}, D_1 = \begin{bmatrix} 1 \end{bmatrix} \tag{5.1}$$

$$A_2 = \begin{bmatrix} -1 & -1 \\ 2 & 1 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_2 = \begin{bmatrix} -1 & 0 \end{bmatrix}, D_2 = \begin{bmatrix} 0 \end{bmatrix}. \tag{5.2}$$

Suppose a similarity transformation is applied to the first mode with the transformation matrix

$$P = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}. \quad (5.3)$$

So the new state space parameters of the first mode is given by

$$A_1 = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 1/2 \end{bmatrix}, C_1 = \begin{bmatrix} 0 & 2 \end{bmatrix}, D_1 = \begin{bmatrix} 1 \end{bmatrix}. \quad (5.4)$$

Suppose as in the partial realization problem, the first mode operates for $k \in [0,7]$ and the switching occures in $k = 8$. It is clear that the first mode will have the same impulse response data as in Chapter 3.3 for $k \in [0,7]$. However, in this case the state at the switching instant will be changed into $x(8) = \begin{bmatrix} 1/2 \\ -1/2 \end{bmatrix}$ from $x(8) = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$. So the output for $k = 8$ will be $y(8) = -1/2$. From that, it can be easily seen that input output behavior of the switched system has changed.

It should be noted that if the same transformation matrix P in Example 5.0.1 was used for similarity transformation of both modes, the input-output behavior would have remained the same. However, since an arbitrary realization algorithm could be used for finding a state space representation from the acquired kernel representations of each mode, one cannot be sure whether the found state space realization would generate the observed data. Therefore, the global viewpoint presented in [20] should be used for realizability of switched linear systems. This viewpoint gives the identifiability conditions for the whole switched system instead of dealing with local modes of the system. It turns out minimality of the global system does not necessarily imply minimality of local systems(modes). In [20], there is also a procedure given for constructing the minimal realization for DTLSSs from Markov parameters. This procedure however, is different from the viewpoint adopted in this thesis. Still, some formal aspects of this new viewpoint is studied through the progress of the thesis, and they will be presented in this chapter briefly. In the next section definitions of span-reachability, observability and minimality of a DTLSS will be given. An example will be provided to show that all modes of a switched system should not necessarily be minimal for the corresponding switched system to be minimal. Then, realizability conditions for DTLSSs in terms of Hankel matrices will be given. Finally, at the end of the chapter, ways of solutions to the problem defined above will be discussed by making use of the existing literature.

5.1 Minimality of DTLSS Realizations

Notation 5.1.1. Denote by $\mathbb{N}$ the set of natural numbers including 0. Consider a set Q which will be called the *alphabet*. Denote by Q^* the set of finite sequences of elements of Q. Finite sequences of elements of Q are called *strings* or *words* over Q. Each non-empty word w is of the form $w = a_1a_2...a_k$ for some $a_1, a_2, ..., a_k \in Q$. The element a_i is called the *ith letter of* w, for $i = 1, 2, ..., k$ and k is called the *length* w. The *empty sequence(word)* is denoted by ε. $|w|$ denotes the length of word w; note that $|\varepsilon| = 0$. The set of non-empty words is denoted by Q^+, i.e. $Q^+ = Q^* \backslash \{\varepsilon\}$. wv is called the *concatenation* of word $w \in Q^*$ with $v \in Q^*$. For each $j = 1, ..., m$, $e_j \in \mathbb{R}^m$ is the jth unit vector which has 1 in its jth element and zeros elsewhere.

Definition 5.1.2. (Lexicographic Ordering) Suppose that $Q = \{1, ..., D\}$. A *lexicographic ordering* $\prec$ can be defined as follows: For any $v, s \in Q^*$, if $|v| < |s|$, then $v \prec s$. If $0 < |v| = |s|$, $v \neq s$ and for some $l \in \{1, ..., |s|\}$, $v_l < s_l$ with the usual ordering of integers and $v_i = s_i$ for $i = 1, ..., l-1$ then $v \prec s$. Here v_i and s_i denote the ith letter of v and s respectively. Note that $\prec$ is a complete ordering and $Q^* = \{v_1, v_2, ...\}$ with $v_1 \prec v_2 \prec ....$ Note that $v_1 = \varepsilon$ and for all $i \in \mathbb{N}$, $q \in Q$, $v_i \prec v_iq$.

Example 5.1.3. (Notation 5.1.1, Definition 5.1.2) This is an example to illustrate Notation 5.1.1 and Definition 5.1.2. Consider a bimodal switched linear system. Since the system has two modes, the *alphabet* set is: $Q = \{1, 2\}$. The set Q^* consists of *all possible switching sequences* of the system which is: $Q^* = \{\varepsilon, 1, 2, 11, 12, 21, 22, 111, 112, 121, 122, 211, 212, 221, 222, ...\}$. Note that the elements of the set Q^* is ordered in accordance with the *lexicographic ordering* defined above. Each element(switching sequence) of the set Q^* except ε is called a *word*. Define two elements of set Q^* as $w = 112$ and $v = 212$. Note that both $|w|, |v| = 3$. The concatenation of these two words is: $wv = 112212$.

Definition 5.1.4. (DTLSS) A definition of a DTLSS similar to the one used earlier in this work is given to make the connection with given notation and context for the purposes of this chapter. A discrete-time linear switched system (DTLSS) is a discrete-time control system of the form

$$\Sigma \begin{cases} x_{t+1} = A_{q_t} x_t + B_{q_t} u_t \text{ and } x_0 \text{ is fixed} \\ y_t = C_{q_t} x_t. \end{cases} \tag{5.5}$$

Here $Q = \{1, ..., D\}$ is the finite set of discrete modes, D is a positive integer. For each $t \in \mathbb{N}$, $q_t \in Q$ is the discrete mode, $u_t \in \mathbb{R}$ is the continuous input, $y_t \in \mathbb{R}^p$ is the output at time t. Moreover, $A_q \in \mathbb{R}^{n\times n}$, $B_q \in \mathbb{R}^{n\times m}$, $C_q \in \mathbb{R}^{p\times n}$ are the matrices of the linear system in mode $q \in Q$, and x_0 is the initial continuous state. The notation

$$(p, m, n, Q, \{(A_q, B_q, C_q) | q \in Q\}, x_0) \tag{5.6}$$

is used as a short-hand representation for DTLSSs of the form (**5.5**).

Throughout the chapter, Σ denotes a DTLSS of the form (**5.5**). The inputs of Σ are the continuous inputs $\{u_t\}_{t=0}^{\infty}$ and the switching signal $\{q_t\}_{t=0}^{\infty}$. The state of the system at time t is x_t. Note that any switching signal is admissible.

Notation 5.1.5. Let Q be a finite set, $\mathscr{X}$ be a linear space, $A_\sigma : \mathscr{X} \to \mathscr{X}$, $\sigma \in Q$ be linear maps and let $w \in Q^*$. The linear map A_w in $\mathscr{X}$ is defined as follows. If $w = \varepsilon$, then A_ε is the identity map, i.e $A_\varepsilon x = x$ for all $x \in \mathscr{X}$. If $w = \sigma_1\sigma_2...\sigma_k \in Q^*$, $\sigma_1, ..., \sigma_k \in Q$, $k > 0$, then

$$A_w = A_{\sigma_k} A_{\sigma_{k-1}} ... A_{\sigma_1}. \tag{5.7}$$

If $\mathscr{X} = \mathbb{R}^n$ for some $n > 0$, then A_w and each A_σ, $\sigma \in Q$ can be identified with an $n\times n$ matrix. In this case A_w defines a product of matrices.

The notation $Q^{<n}$ is used to represent the set $\{w \in Q^* || w| < n\}$ of all words $w \in Q^*$ of length at most $n-1$. M_n is the cardinality of $Q^{<n}$ and an enumeration is fixed such that

$$Q^{<n} = \{v_1, ..., v_{M_n}\}. \tag{5.8}$$

Notation defined above will be used to define observability and reachability matrices for DTLSSs.

Theorem 5.1.6. ***(Span-Reachability)*** *Define the span-reachability matrix* $\mathfrak{R}(\Sigma)$ *of* Σ

$$\mathfrak{R}(\Sigma) = \left[\begin{array}{cccc} A_{v_1}\widetilde{B}, & A_{v_2}\widetilde{B}, & \ldots, & A_{v_{M_n}}\widetilde{B} \end{array} \right] \in \mathbb{R}^{n\times(|Q|m+1)M_n} \text{ where}$$
$$\widetilde{B} = \left[\begin{array}{cccc} x_0, & B_1, & \ldots, & B_D \end{array} \right]. \tag{5.9}$$

Then, Σ *is span-reachable if and only if rank* $\mathfrak{R}(\Sigma) = n$.

Theorem 5.1.7. ***(Observability)*** *Define the observability matrix* $\mathfrak{O}(\Sigma) \in \mathbb{R}^{p|Q|M_n\times n}$ *of* Σ *as follows:*

$$\mathfrak{O}(\Sigma) = \left[\begin{array}{c} \widetilde{C}A_{v_1} \\ \widetilde{C}A_{v_2} \\ \vdots \\ \widetilde{C}A_{v_{M_n}} \end{array} \right] \quad \text{where} \quad \widetilde{C} = \left[\begin{array}{c} C_1 \\ C_2 \\ \vdots \\ C_D \end{array} \right] \tag{5.10}$$

Then Σ *is observable if and only if rank* $\mathfrak{O}(\Sigma) = n$.

Example 5.1.8. Consider a bimodal single input single output DTLSS of order $n = 3$ with the initial condition x_0(Let $\Sigma = (p,m,n,Q,\{(A_q,B_q,C_q)|q \in Q\},x_0)$ with $Q = \{1,2\}$, $n = 3$, $m = 1$, $p = 1$). The span-reachability matrix of Σ is defined as:

$$\mathfrak{R}(\Sigma) = \left[\begin{array}{ccccccc} \widetilde{B} & A_1\widetilde{B} & A_2\widetilde{B} & A_1A_1\widetilde{B} & A_1A_2\widetilde{B} & A_2A_1\widetilde{B} & A_2A_2\widetilde{B} \end{array} \right] \in \mathbb{R}^{3\times 21} \text{ where}$$
$$\widetilde{B} = \left[\begin{array}{ccc} x_0 & B_1 & B_2 \end{array} \right]. \tag{5.11}$$

The observability matrix $\mathfrak{O}(\Sigma)$ of Σ is defined similarly.

Procedure 5.1.9. (Reachability Reduction) Assume dim $\mathfrak{R}(\Sigma) = n^r$ and choose a basis $b_1,\ldots,b_n$ of $\mathbb{R}^n$ such that $b_1,\ldots,b_{n^r}$ span Im $\mathfrak{R}(\Sigma)$. In the new basis, $A_q,B_q,C_q,q \in Q$ and x_0 become as follows

$$A_q = \left[\begin{array}{cc} A_q^r & A_q' \\ 0 & A_q'' \end{array} \right], C_q = \left[\begin{array}{cc} C_q^r & C_q^{nr} \end{array} \right], B_q = \left[\begin{array}{c} B_q^r \\ 0 \end{array} \right], x_0 = \left[\begin{array}{c} x_0^r \\ 0 \end{array} \right] \tag{5.12}$$

where $A_q^r \in \mathbb{R}^{n^r\times n^r}$, $B_q^r \in \mathbb{R}^{n^r\times m}$, $x_0^r \in \mathbb{R}^{n^r}$. Then $\Sigma_r = (p,m,n^r,Q,\{(A_q^r,B_q^r,C_q^r)|q \in Q\},x_0^r)$ is span-reachable, and has the same input-output map as Σ.

Intuitively, Σ_r is obtained from Σ by restricting the dynamics and the output map of Σ to the space Im $\mathfrak{R}(\Sigma)$.

Procedure 5.1.10. (Observability Reduction) Assume that dim ker $\mathfrak{O}(\Sigma) = n - n^o$ and let $b_1,\ldots,b_n$ be a basis in $\mathbb{R}^n$ such that $b_{n^0+1},\ldots,b_n$ span Ker $\mathfrak{O}(\Sigma)$. In the new basis, A_q,B_q,C_q and x_0 can be rewritten as

$$A_q = \left[\begin{array}{cc} A_q^o & 0 \\ A_q' & A_q'' \end{array} \right], C_q = \left[\begin{array}{cc} C_q^o & 0 \end{array} \right], B_q = \left[\begin{array}{c} B_q^o \\ B_q' \end{array} \right], x_0 = \left[\begin{array}{c} x_0^o \\ x_0' \end{array} \right] \tag{5.13}$$

where $A_q^o \in \mathbb{R}^{n^o \times n^o}$, $B_q^o \in \mathbb{R}^{n^o \times m}$, $C_q^o \in \mathbb{R}^{p \times n^o}$ and $x_0^o \in \mathbb{R}^{n^o}$. Then the DTLSS $\Sigma_o = (p, m, n^o, Q, \{(A_q^o, B_q^o, C_q^o) | q \in Q\}, x_0^o)$ is observable and its input-output map is the same as that of Σ. If Σ is span-reachable, then so is Σ_o.

Intuitively, Σ_o is obtained from Σ by merging any two states x_1, x_2 of Σ, for which $\mathfrak{O}(\Sigma)x_1 = \mathfrak{O}(\Sigma)x_2$.

Procedure 5.1.11. (Minimization) First transform Σ to a span-reachable DTLSS Σ_r and then transform Σ_r to an observable DTLSS $\Sigma_m = (\Sigma_r)_o$. Then Σ_m is a minimal realization of the input-output map of Σ.

Using the definitions and procedures above, Example 5.1.12 is given to highlight the fact that minimality of the DTLSS does not imply minimality of its modes.

Example 5.1.12. Let $\Sigma = (p, m, n, Q, \{(A_q, B_q, C_q) | q \in Q\}, x_0)$ with $Q = \{1, 2\}$, $n = 3$, $m = 1$, $p = 1$, $x_0 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$,

$$\begin{aligned} A_1 &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, C_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\ A_2 &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, C_2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{5.14}$$

The system is observable, but it is not span-reachable. In order to see observability, notice that the sub-matrix $\begin{bmatrix} C_1^T (C_1 A_1)^T C_2^T \end{bmatrix}^T$ of $\mathfrak{O}(\Sigma)$ is of rank 3. In order to see that Σ is not span-reachable, notice that the last row of $\mathfrak{R}(\Sigma)$ is a zero row. Hence dim $\mathfrak{R}(\Sigma) \leq 2$. Using Procedure 5.1.11, we can transform Σ to the minimal realization

$$\Sigma_m = (p, m, n^m, Q, \{(A_q^m, B_q^m, C_q^m) | q \in Q\}, x_0^m) \tag{5.15}$$

of y_Σ (input-output map of Σ): $Q = \{1, 2\}, n^m = 2, x_0^m = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ and

$$\begin{aligned} A_1^m &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, B_1^m = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, C_1^m = \begin{bmatrix} 0 & 1 \end{bmatrix} \\ A_2^m &= \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, B_2^m = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C_2^m = \begin{bmatrix} 0 & 0 \end{bmatrix} \end{aligned} \tag{5.16}$$

Since a minimal realization of a linear system must be reachable and observable, it is easy to see that neither $(A_1^m, B_1^m, C_1^m, x_0^m)$ nor $(A_2^m, B_2^m, C_2^m, x_0^m)$ are minimal.

5.2 Existence of a Realization

In this section, necessary and sufficient conditions for the existence of a DTLSS realization for a known Markov-parameter sequence similar to the conditions in the linear case will be given. Following [20], first, the definitions of Markov parameters, combined Markov-parameters and Hankel matrices for DTLSSs will be briefly given. Then the result on existence of a realization in [20] will be stated.

Definition 5.2.1. (Input/Output Map) Denote $\mathscr{U} = Q \times \mathbb{R}^m$ and $\mathscr{U}^+$ as the set of all non-empty and finite sequences of elements of $\mathscr{U}$. A sequence

$$w = (q_0, u_0)...(q_t, u_t) \in \mathscr{U}^+,\ t \geq 0 \tag{5.17}$$

describes the case, when the discrete mode q_i and the input u_i are fed to Σ at time i, for $i = 0, ..., t$. Also, consider a state $x_0 \in \mathbb{R}^n$. For any $w \in \mathscr{U}^+$ of the form (**5.17**), denote by $y_\Sigma(x_0, w)$ the output of Σ at time t, if Σ is started from x_0 and the inputs $\{u_i\}_{i=0}^t$ and the discrete modes $\{q_i\}_{i=0}^t$ are fed to the system. The map $y_\Sigma : \mathscr{U}^+ \to \mathbb{R}^p$, defined by $\forall w \in \mathscr{U}^+ : y_\Sigma(w) = y(x_0, w)$, is called the input/output map of Σ.

Definition 5.2.1 implies that the input/output behavior of a DTLSS can be formalized as a map

$$f : \mathscr{U}^+ \to \mathbb{R}^p. \tag{5.18}$$

The value $f(w)$ for w of the form (**5.17**) represents the output of the DTLSS (considered as a black-box system) at time t, if the inputs $\{u_i\}_{i=0}^t$ and the switching sequence $\{q_i\}_{i=0}^t$ are fed to the system.

In the sequel, we identify any element $w = (q_0, u_0)...(q_t, u_t) \in \mathscr{U}^+$ with the pair of sequences (v, u), $v \in Q^+$, $u \in (\mathbb{R}^m)^+$, $v = q_0...q_t$ and $u = u_0...u_t$. Also, the following notation is needed to define the Markov Parameters of an input/output map:

Consider the input/output map f. For each word $v \in Q^+$ of length $|v| = t > 0$ we define $f_v : (\mathbb{R}^m)^t \to \mathbb{R}^p$ as $f_v(u) = f((v, u))$.

Definition 5.2.2. (Markov Parameters) Denote $Q^{k,*} = \{w \in Q^* \mid |w| \geq k\}$. Define the maps $S_0^f : Q^{1,*} \to \mathbb{R}^p$ and $S_j^f : Q^{2,*} \to \mathbb{R}^p$, $j = 1, ..., m$ as follows; for any $v \in Q^*$, $q, q_0 \in Q$,

$$\begin{aligned} S_0^f(vq) &= f_{vq}(0,...,0) \text{ and} \\ S_j^f(q_0vq) &= f_{q_0vq}(e_j,0,...,0) - f_{q_0vq}(0,...,0) \end{aligned} \tag{5.19}$$

where $e_j \in \mathbb{R}^m$ is the vector with 1 as its jth entry and zero everywhere else. The collection of maps $\{S_j^f\}_{j=0}^m$ is called the *Markov Paremeters* of f.

In addition, Markov parameters of a state space realization can be defined as follows: Define v as any possible switching sequence(word) for DTLSS, i.e define v as any element of the set Q^*, that is $v \in Q^*$. Define q, q_0 as any discrete mode $q, q_0 \in Q$(Define q, q_0 as any letter of the alphabet Q). Note that q, q_0 are not necessarily distinct. The *Markov Parameters* of a DTLSS are defined by,

$$\begin{aligned} S_0^f(vq) &= C_qA_vx_0 \text{ and} \\ S_j^f(q_0vq) &= C_qA_vB_{q_0}e_j, \; j = 1, ..., m. \end{aligned} \tag{5.20}$$

Markov parameters of a DTLSS can be interpreted as Markov parameters of an ordinary linear system except they are defined for all possible switching sequences in the switched case.

Definition 5.2.3. (Combined Markov Parameters) A combined Markov-parameter $M^f(v)$ of f indexed by the word $v \in Q^*$ is the following $pD \times (Dm+1)$ matrix

$$M^f(v) = \begin{bmatrix} S_0^f(v1) & S^f(1v1) & ... & S^f(Dv1) \\ S_0^f(v2) & S^f(1v2) & ... & S^f(Dv2) \\ \vdots & \vdots & ... & \vdots \\ S_0^f(vD) & S^f(1vD) & ... & S^f(DvD) \end{bmatrix} \tag{5.21}$$

where for any $w \in Q^+$, $|w| > 2$, $S^f(w) = \begin{bmatrix} S_1^f(w) & S_2^f(w) & ... & S_m^f(w) \end{bmatrix}$.

Definition 5.2.4. (Hankel Matrix) Let the elements of the set Q^* are ordered with the lexigoraphic ordering $\prec$ (as in Example 2), i.e $Q^* = \{v_1, v_2, ...\}$ with $v_1 \prec v_2 \prec ....$ The

Hankel Matrix H_f of f is defined as the following infinite matrix

$$H_f = \begin{bmatrix} M^f(v_1v_1) & M^f(v_2v_1) & ... & M^f(v_kv_1) & ... \\ M^f(v_1v_2) & M^f(v_2v_2) & ... & M^f(v_kv_2) & ... \\ M^f(v_1v_3) & M^f(v_2v_3) & ... & M^f(v_kv_3) & ... \\ \vdots & \vdots & ... & \vdots & \ddots \end{bmatrix} \tag{5.22}$$

Example 5.2.5. Consider again the bimodal DTLSS in Example 2. Remember $Q^* = \{v_1, v_2, ...\} = \{\varepsilon, 1, 2, 11, 12, ...\}$ with $v_1 \prec v_2 \prec ....$ The Markov parameters of this system are given by

$$\begin{array}{rr} S_0^f(1) = C_1x_0 & S_1^f(11) = C_1B_1 \\ S_0^f(2) = C_2x_0 & S_1^f(12) = C_2B_1 \\ S_0^f(11) = C_1A_1x_0 & S_1^f(21) = C_1B_2 \\ S_0^f(12) = C_2A_1x_0 & S_1^f(22) = C_2B_2 \\ S_0^f(21) = C_1A_2x_0 & S_1^f(111) = C_1A_1B_1 \\ S_0^f(22) = C_2A_2x_0 & S_1^f(112) = C_2A_1B_1 \\ S_0^f(111) = C_1A_1^2x_0 & S_1^f(121) = C_1A_2B_1 \\ S_0^f(112) = C_2A_1^2x_0 & S_1^f(122) = C_2A_2B_1 \\ S_0^f(121) = C_1A_2A_1x_0 & S_1^f(211) = C_1A_1B_2 \\ \vdots & \vdots \end{array} \tag{5.23}$$

Combined Markov-parameters of the system are given by

$$\begin{aligned} M^f(\varepsilon) &= \begin{bmatrix} S_0^f(1) & S_1^f(11) & S_1^f(21) \\ S_0^f(2) & S_1^f(12) & S_1^f(22) \end{bmatrix} \\ M^f(1) &= \begin{bmatrix} S_0^f(11) & S_1^f(111) & S_1^f(211) \\ S_0^f(12) & S_1^f(112) & S_1^f(212) \end{bmatrix} \\ M^f(2) &= \begin{bmatrix} S_0^f(21) & S_1^f(121) & S_1^f(221) \\ S_0^f(22) & S_1^f(122) & S_1^f(222) \end{bmatrix} \\ &\vdots \end{aligned} \tag{5.24}$$

And the Hankel matrix constructed from combined Markov-parameters is given by

$$H_f = \begin{bmatrix} M^f(\varepsilon) & M^f(1) & M^f(2) & ... \\ M^f(1) & M^f(11) & M^f(21) & ... \\ M^f(2) & M^f(12) & M^f(22) & ... \\ M^f(11) & M^f(111) & M^f(211) & ... \\ M^f(12) & M^f(112) & M^f(212) & ... \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{5.25}$$

Theorem 5.2.6. *A Markov parameter sequence is realizible by a DTLSS if and only if rank $H_f < +\infty$. A minimal realization of f can be constructed from H_f (see Procedure 5 in [2]) and any minimal DTLSS realization of the sequence has dimension rank H_f.*

Theorem 5.2.6 gives a compact realizability condition for DTLSSs similar to the linear case. However, checking this condition in the recursive identification algorithm given in Chapter 3 is not possible, since impulse response data for all possible switching sequences would be required. Nevertheless, after acquiring the kernel representation for the modes it can be stated that there *exists* a minimal state space realization that would generate the observed data. In particular, the method given in [13] can be used to solve the problem of writing the state equations in a common basis (which is defined in the beginning of this chapter) so that found state space representations for each local mode generate the same response to an impulse or arbitrary inputs that are used in the identification process. For this method to be used the switching instants must be known. As illustrated in the example in Chapter 3.4. detecting the exact instant of switching may not be possible with the recursive algorithms presented in this thesis. However, taking the instants of switch detection as the exact moment of switching and applying the method in [13] would be possible and solve the problem. It should be noted that even though the state space representations acquired with the method in [13] would be consistent with the data at hand, they can be acquired up to a similarity transformation and they may not be consistent with another data sequence from the same DTLSS. Characterizing all possible state space realizations that would generate the observed data may also be possible, so it is another problem.

6. CONCLUSIONS

The two main results of this thesis can be stated as follows: Firstly, a recursive procedure which gives kernel representations of the modes of a DTLSS from impulse response measurements is presented. Then this procedure is modified accordingly and also presented as a solution to the problem of recursive identification of a DTLSS from a measured arbitrary input/output sequence. The behavioral approach to linear system theory is adopted to state the results. In partial realization problem, it is shown that the zero-input dynamics of the modes can be uniquely identified provided the observed trajectory is sufficiently rich and the dwell time of the modes is greater than a given lower bound. For the latter problem, it is also shown that input/output behavior of all modes can be identified if the input sequence for the mode satisfies a specific persistency of excitation criterion and the dwell times of the modes are greater than a given lower bound. In both problems, the system model is recursively updated every time a new input-output sample is available. This makes the methods suitable for on-line implementation and to detect mode changes of the switched system. Finally, a way, existing in the literature, to find state space representations of the modes which are consistent with the observed data is suggested. In future work the recursive method can be extended to the identification of a DTLSS from multiple input-output trajectories corresponding to different switching sequences. The method can be further improved with respect to numerical efficiency and accuracy. A block recursive version of the method can also be developed in which only the model testing and event detection steps are performed recursively. Then, subspace methods can be used to identify the modes which satisfy the dwell time assumptions.

REFERENCES

[1] **Liberzon, D.**, 2003. Switching in Systems and Control, Birkhäuser, Boston, MA.

[2] **Sun, Z. and Ge, S.**, 2005. Switched Linear Systems: Control and Design, Springer-Verlag, London.

[3] **Ljung, L.**, 1987. System Identification: Theory for the User, P T R Prentice Hall, New Jersey, NJ.

[4] **Antoulas, A.C. and Willems, J.C.**, 1993. A behavioral approach to linear exact modeling, *IEEE Trans. Automatic Control, vol.38, no.12*, 1776–1802.

[5] **Willems, J.C.**, 1986. From time series to linear system - Part II: Exact modeling, *Automatica, vol. 22*, 675–694.

[6] **Willems, J.C.**, 2007. Recursive Computation of the MPUM, *Lecture Notes in Control and Information Sciences, Volume 364/2007*, 329–344.

[7] **Willems, J.C.**, 1991. Paradigms and puzzles in the theory of dynamical systems, *IEEE Trans. Automatic Control, vol. 36, No. 3*, 259–294.

[8] **Polderman, J.W. and Willems, J.C.**, 1998. Introduction to Mathematical Systems Theory: A Behavioral Approach, Springer-Verlag, New York, NY.

[9] **Kalman, R.E.**, 1971. On minimal partial realizations of a linear input output map, in R.E. Kalman, N. DeClaris (Eds.), Aspects of Network and System Theory, Holt, Rinehart and Winston, New York, NY.

[10] **Kuijper, M. and Willems, J.C.**, 1997. On constructing a shortest linear recurrence relation, *IEEE Trans. Automatic Control, vol. 42, no. 11*, 1554–1558.

[11] **Bako, L.**, **Mercere, G. and Lecoeuche, S.**, 2009. On-line structured subspace identification with application to switched linear system, *Int. J. of Control vol. 82, no. 8*, 1496–1515.

[12] **Paoletti, S.**, **Julosky, A.**, **Trecate, G.F. and Vidal, R.**, 2007. Identification of hybrid systems: A tutorial, *European Journal of Control, vol. 13*, 242–260.

[13] **Verdult, V. and Verhaegen, M.**, 2004. Subspace identification of piecewise linear systems, *43rd IEEE Conf. on Decision and Control*, 3838–3843.

[14] **Petreczky, M. and Van Schuppen, J.**, 2011. Partial realization theory and algorithms for linear switched systems: A formal power series approach, *Accepted for publication in Automatica*.

[15] **Willems, J.C.**, 1986. From time series to linear system-Part I: Finite dimensional linear time invariant systems, *Automatica, vol. 22*, 561–580.

[16] **Wolovich, W.A.**, 1974. Linear Multivariable Systems, Springer-Verlag, New York, NY.

[17] **Markovsky, I.**, **Willems, J.C.**, **Van Huffel, S. and De Moor, B.**, 2006. Exact and Approximate Modeling of Linear Systems: A Behavioral Approach, SIAM, Philadelphia, PA.

[18] **Willems, J.C.**, **Rapisarda, P.**, **Markovsky, I. and De Moor, B.L.M.**, 2005. A note on persistency of excitation, *System and Control Letters, vol. 54*, 325–329.

[19] **Rugh, W.J.**, 1996. Linear System Theory, Prentice Hall, New Jersey, NJ.

[20] **Petreczky, M.**, **Bako, L. and Van Schuppen, J.H.**, 2011. Realization theory of discrete-time linear switched systems, *Submitted to Automatica*.

APPENDICES

APPENDIX A.1 : MATLAB Code for Recursive Partial Realization of DTLSSs
APPENDIX A.2 : MATLAB Code for Recursive Identification of DTLSSs from Arbitrary Input/Output Sequences

APPENDIX A.1

```
clear
clc
syms x

R11=1;
R12=0;
R21=0;
R22=1;
R=[R11 R12;R21 R22];
ck=[1 0]*R*[1;0];
L=0;
deltak1=0;
deltak2=1;
V=[x x;x x];

flag=0;
%Program will be ended with this flag when there are
%no data remaining
counter=0;
flag2=0;
counter2=0;

deg=input('Order of modes:\n');

while flag==0
    flag2=0;
    counter2=0;
    counter=counter+1;
    y=input('Please enter the impulse response sequence\n');

    if counter==1
        y=y
    else
        y=[B y]
    end
        B=y;
    while flag2==0
        counter2=counter2+1;
        if (counter2>length(y))
            flag2=1;
            R
```

```
    R11=1;
    R12=0;
    R21=0;
    R22=1;
    R=[R11 R12;R21 R22];
    ck=[1 0]*R*[1;0];
    L=0;
    deltak1=0;
    deltak2=1;
    V=[x x;x x];
    continue
end
D=y(1:counter2);
D_ters=fliplr(D);%Time reversal operation
D_sym=poly2sym(D_ters);

hata_pol1=D_sym*ck;
hata_pol2=sym2poly(hata_pol1);
hata_pol=poly2sym(hata_pol2);
hata1=sym2poly(hata_pol1);
hata2=fliplr(hata1);
if (counter2>length(hata2))
    deltak1=0;
else
    deltak1=hata2(counter2);
end
%The degree of second row is defined as Lk2
Lk2=max((length(R21)-1),(length(R22)-1));


%%%%%%%%%%%%%%%EVENT DETECTION%%%%%%%%%
if (L==deg) && (L<Lk2) && (deltak1~=0)
    an=counter2;
    fprintf('Switch at k=%i\nEnter the data starting from
    instant n=%i again\n',an,an)
    fprintf('Ker. repr. of the MPUM for the previous mode:')
    R
    R=[1 0;0 1];
    V=[x x;x x];
    ck=[1 0]*R*[1;0];
    L=0;
    deltak1=0;
    deltak2=1;
    kontrol2=0;
    counter2=0;
    flag2=1;
    y=0;
```

```
            counter=0;
            continue
        end

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%UPDATE MODEL%%%%%%%%%%%%%%%%%%%%%%
        if ((deltak1==0) || (L>((counter2 -1)/2)))
            V(1,1)=1;
            V(1,2)=-deltak1;
            V(2,1)=0;
            V(2,2)=x;
            L=L;
        else
            V(1,1)=1;
            V(1,2)=-deltak1;
            V(2,1)=x/deltak1;
            V(2,2)=0;
            L=(counter2 -1)-L;
        end

        V;
        R=V*R;
        R11=sym2poly(R(1,1));
        R(1,1)=poly2sym(R11);
        R12=sym2poly(R(1,2));
        R(1,2)=poly2sym(R12);
        R21=sym2poly(R(2,1));
        R(2,1)=poly2sym(R21);
        R22=sym2poly(R(2,2));
        R(2,2)=poly2sym(R22);
        ck=[1 0]*R*[1;0];
    end
    kontrol=input('Are there any remaining data?(1/0)\n');
    if (kontrol==0)
        flag=1;
    end
end
```

APPENDIX A.2

```
clear
clc
syms x
%%%%%% 1. INITIALIZATION %%%%%%%%%%
R=[x 0 0;0 1 0;0 0 1];
R(1,1)=1;
V=[x x x;x x x;x x x];
sequence=[];
```

```
y=[];
u=[];
w=[];
flag=1;
n=input('Order of the modes:\n');
Nmin=4*n+1;
while flag==1
    R=[x 0 0;0 1 0;0 0 1];
    R(1,1)=1;
    V=[x x x;x x x;x x x];
    y=input('Enter the output sequence:\n');
    u=input('Enter the input sequence:\n');
    seq=[y;u];
    sequence=[sequence, seq];
    [q, N]=size(sequence);
    w=[0 sequence(1,:); 0 sequence(2,:); 1 zeros(1,(N))];
    N=N+1;
    k=1;
    for k=1:N
        data=fliplr(w(:,1:k));

        %%%%%%%%%%%% 2. ERROR COMPUTATION %%%%%%%%%%%%%%%%%%%%%%%

        R11=sym2poly(R(1,1));
        l11=length(R11);
        v11=zeros(1,k);
        for i=1:l11
            if i==l11
                v11=v11+R11(i).*data(1,:);
            else
                v11=v11+R11(i).*[data(1,l11-i+1:k) zeros(1,l11-i)];
            end
        end

        R12=sym2poly(R(1,2));
        l12=length(R12);
        v12=zeros(1,k);
        for i=1:l12
            if i==l12
                v12=v12+R12(i).*data(2,:);
            else
                v12=v12+R12(i).*[data(2,l12-i+1:k) zeros(1,l12-i)];
            end
        end

        R13=sym2poly(R(1,3));
        l13=length(R13);
```

```
v13=zeros(1,k);
for i=1:l13
    if i==l13
        v13=v13+R13(i).*data(3,:);
    else
        v13=v13+R13(i).*[data(3,l13-i+1:k) zeros(1,l13-i)];
    end
end

vsum=v11+v12+v13;
e1=vsum(1);

R21=sym2poly(R(2,1));
l21=length(R21);
u21=zeros(1,k);
for i=1:l21
    if i==l21
        u21=u21+R21(i).*data(1,:);
    else
        u21=u21+R21(i).*[data(1,l21-i+1:k) zeros(1,l21-i)];
    end
end

R22=sym2poly(R(2,2));
l22=length(R22);
u22=zeros(1,k);
for i=1:l22
    if i==l22
        u22=u22+R22(i).*data(2,:);
    else
        u22=u22+R22(i).*[data(2,l22-i+1:k) zeros(1,l22-i)];
    end
end

R23=sym2poly(R(2,3));
l23=length(R23);
u23=zeros(1,k);
for i=1:l23
    if i==l23
        u23=u23+R23(i).*data(3,:);
    else
        u23=u23+R23(i).*[data(3,l23-i+1:k) zeros(1,l23-i)];
    end
end
usum=u21+u22+u23;
e2=usum(1);
```

```
R31=sym2poly(R(3,1));
l31=length(R31);
y31=zeros(1,k);
for i=1:l31
    if i==l31
        y31=y31+R31(i).*data(1,:);
    else
        y31=y31+R31(i).*[data(1,l31-i+1:k) zeros(1,l31-i)];
    end
end

R32=sym2poly(R(3,2));
l32=length(R32);
y32=zeros(1,k);
for i=1:l32
    if i==l32
        y32=y32+R32(i).*data(2,:);
    else
        y32=y32+R32(i).*[data(2,l32-i+1:k) zeros(1,l32-i)];
    end
end

R33=sym2poly(R(3,3));
l33=length(R33);
y33=zeros(1,k);
for i=1:l33
    if i==l33
        y33=y33+R33(i).*data(3,:);
    else
        y33=y33+R33(i).*[data(3,l33-i+1:k) zeros(1,l33-i)];
    end
end
ysum=y31+y32+y33;
e3=ysum(1);

er=[e1;e2;e3];

%%%%%%%%%%%%%%%%%%%%%%% 3. EVENT DETECTION %%%%%%%%%%%%%%%%%%%%%%%%%%
if (k>Nmin) && (e1~=0)
    fprintf('Switch detected at k=%d \n',k-1)
    R
    R11=sym2poly(R(1,1));
    R12=sym2poly(R(1,2));
    R13=sym2poly(R(1,3));
    degr1=max([length(R11),length(R12),length(R13)])-1;
    row1=x^degr1.*subs(R(1,:),x^-1);
    num=-sym2poly(row1(2));
```

```
        den=sym2poly(row1(1));
        tf(num,den)
        pause
        data(1:2,1)
        fprintf('When asked, enter again the data received
        after the displayed point including it.\n')
        pause
        %%%%%%%%%%%%%%%%%%%%%%%% 1. INITIALIZATION %%%%%%%%%%%%%%
        R=[x 0 0;0 1 0;0 0 1];
        R(1,1)=1;
        V=[x x x;x x x;x x x];
        sequence=[];
        y=[];
        u=[];
        w=[];
        flag=1;
        break
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Preprocess R such that only last element of error
%corresponding to the same degree rows of R is nonzero
degr1=max([length(R11),length(R12),length(R13)])-1;
degr2=max([length(R21),length(R22),length(R23)])-1;
degr3=max([length(R31),length(R32),length(R33)])-1;
r=find(er);

if (degr1==degr2) && (degr1~=degr3)
    if (any(r==1)==1) && (er(2)==0)
        R([1 2],:)=R([2 1],:);
        er([1 2])=er([2 1]);
    elseif (any(r==1)==1) && (er(2)~=0)
        R(1,:)=R(1,:)-(er(1)/er(2)).*R(2,:);
        er(1)=0;
    end
elseif (degr2==degr3) && (degr1~=degr2)
    if (any(r==2)==1) && (er(3)==0)
        R([2 3],:)=R([3 2],:);
        er([2 3])=er([3 2]);
    elseif (any(r==2)==1) && (er(3)~=0)
        R(2,:)=R(2,:)-(er(2)/er(3)).*R(3,:);
        er(2)=0;
    end
elseif (degr1==degr2) && (degr2==degr3)
    if (length(r)==1) && (er(1)~=0)
        R([1 3],:)=R([3 1],:);
        er([1 3])=er([3 1]);
```

```
    elseif (length(r)==1) && (er(2)~=0)
        R([2 3],:)=R([3 2],:);
        er([2 3])=er([3 2]);
    elseif (length(r)==2) && (er(1)==0)
        R(2,:)=R(2,:)-(er(2)/er(3)).*R(3,:);
        er(2)=0;
    elseif (length(r)==2) && (er(2)==0)
        R(1,:)=R(1,:)-(er(1)/er(3)).*R(3,:);
        er(1)=0;
    elseif (length(r)==2) && (er(3)==0)
        R([2 3],:)=R([3 2],:);
        er([2 3])=er([3 2]);
        R(1,:)=R(1,:)-(er(1)/er(3)).*R(3,:);
        er(1)=0;
    elseif (length(r)==3)
        R(1,:)=R(1,:)-(er(1)/er(3)).*R(3,:);
        R(2,:)=R(2,:)-(er(2)/er(3)).*R(3,:);
        er(1)=0;
        er(2)=0;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%Normalize the error such that first nonzero element
%%would be 1.
r=find(er);
er=(1/er(r(1))).*er;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

er;

%%%%%%%%%%%%%%%%%% 4. MODEL UPDATE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Create the update matrix V
if (length(r)==3)
    V=[x 0 0;-er(2) 1 0;-er(3) 0 1];
elseif (length(r)==2)
    if (er(1)==1)
        V=[x 0 0;-er(2) 1 0;-er(3) 0 1];
    elseif (er(1)==0)
        V=[1 0 0;0 x 0;0 -er(3) 1];
    end
elseif (length(r)==1)
    if (er(1)==1)
        V=[x 0 0;0 1 0;0 0 1];
    elseif (er(2)==1)
```

```
        V=[1 0 0;0 x 0;0 0 1];
    else
        V=[1 0 0;0 1 0;0 0 x];
    end
else
    V=eye(3);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

V;

%%Update the kernel representation of the MPUM: R
R=V*R;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%Define row degrees
R11=sym2poly(R(1,1));
R12=sym2poly(R(1,2));
R13=sym2poly(R(1,3));
R21=sym2poly(R(2,1));
R22=sym2poly(R(2,2));
R23=sym2poly(R(2,3));
R31=sym2poly(R(3,1));
R32=sym2poly(R(3,2));
R33=sym2poly(R(3,3));
degr1=max([length(R11),length(R12),length(R13)])-1;
degr2=max([length(R21),length(R22),length(R23)])-1;
degr3=max([length(R31),length(R32),length(R33)])-1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%Reorder the rows of R in ascending row degrees
if (degr1>degr2) && (degr2>degr3)
    R([1 3],:)=R([3 1],:);
elseif (degr1>degr3) && (degr3>degr2)
    R([1 3],:)=R([3 1],:);
    R([1 2],:)=R([2 1],:);
elseif (degr2>degr3) && (degr3>degr1)
    R([2 3],:)=R([3 2],:);
elseif (degr2>degr1) && (degr1>degr3)
    R([1 3],:)=R([3 1],:);
    R([2 3],:)=R([3 2],:);
elseif (degr3>degr1) && (degr1>degr2)
    R([1 2],:)=R([2 1],:);
elseif (degr1>degr2) && (degr2==degr3)
    R([1 3],:)=R([3 1],:);
elseif (degr2>degr1) && (degr1==degr3)
    R([2 3],:)=R([3 2],:);
```

```
            elseif (degr2>degr3) && (degr1==degr2)
                R([1 3],:)=R([3 1],:);
            elseif (degr1>degr2) && (degr1==degr3)
                R([1 2],:)=R([2 1],:);
            end
            %R
            %pause
        end
        a=input('Any more data? (1/0)');
        if a==1
            continue
        else
            flag=0;
        end
        R
        R11=sym2poly(R(1,1));
        R12=sym2poly(R(1,2));
        R13=sym2poly(R(1,3));
        degr1=max([length(R11),length(R12),length(R13)])-1;
        row1=x^degr1.*subs(R(1,:),x^-1);
        num=-sym2poly(row1(2));
        den=sym2poly(row1(1));
        tf(num,den)
        pause
    end
```

CURRICULUM VITAE

Name Surname: Mert Baştuğ

Place and Date of Birth: Adana, 22/07/1988.

Adress: Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Office: 1113.

E-Mail: bastug@itu.edu.tr

B.Sc.: Istanbul Technical University, Electronics Engineering, 2010.

Professional Experience and Rewards: Research Assistant in ITU Department of Electronics and Communication Engineering since November 2010.

List of Publications and Patents:

PUBLICATIONS/PRESENTATIONS ON THE THESIS

- **Baştuğ M.** and Çevik M. K. K., 2011: Recursive partial realization of switched linear systems, in *Proc. ELECO'11 7th Int. Conf. on Electrical and Electronics Eng.*, pp. 95-99, December 1-4, 2011 Bursa, Turkey.