Use Authorization

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Idaho State University, I agree that the library shall make it freely available for inspection. I further state that permission for extensive copying of my thesis for scholarly purposes may be granted by the Dean of Graduate School, Dean of my academic division, or by the University Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature ___________________________

Date___________________________

System Identification of the Portneuf River using Irregularly Spaced Data

by

Prasis Timilsena

A thesis

submitted in partial fulfillment of

the requirements for the degree of

Master of Science in the Department of Mechanical Engineering

Idaho State University

Summer 2022

Committee Approval

To the Graduate Faculty:

The members of the committee appointed to examine the thesis of Prasis Timilsena find it satisfactory and recommend that it be accepted.

Major Advisor

Dr. Ken Bosworth

Committee Member

Dr. Anish Sebastian

Graduate Faculty Representative

Dr. James Mahar

Acknowledgments

I would like to show my profound gratitude to my family, friends, and advisors who helped throughout this project. I would also like to heartily express my appreciation, especially, to my primary advisor Dr. Ken Bosworth who guided and supported me throughout the entire duration of this project. His suggestions and guidance always kept me inspired to work on this thesis and finally complete the research project. I would also like to thank the committee members Dr. Anish Sebastian and the Graduate Faculty Representative Dr. James Mahar.

I am also genuinely thankful to Michel Crump, Safal Lama, and Kiran Pandey for their generous help and support throughout the research. It would have been difficult to complete the project without their continuous assistance. I would also like to acknowledge Greg Mladenka of the Idaho Department of Environmental Quality for providing the data used in this project.

Finally, I am extremely grateful to my parents Pitamber Timilsena and Nirmala Timilsena, younger brother Prabin Timilsena, Sharita Bhandari, and my entire family for their kind love and support.

Table of Contents

List of Figures:

System Identification of Portneuf River Using Irregularly Spaced Data

Thesis Abstract- Idaho State University (2022)

System Identification or Data-based Modeling is an important tool in the field of data science which deals with modeling a dynamic system using available data. For some natural systems, different modeling techniques have been proposed and applied but the most successful techniques are process-based modeling and data-based modeling. Process-based models provide a detailed description about the process involved in any mechanical system whereas data-based models mainly focus on the behavior of the system itself. In this thesis, data-based modeling is proposed for the Portneuf River in Idaho, United states. The aim of this thesis is to find a suitable model for the river ecosystem which depends on various factors like the ambient temperature of the surroundings, the flow of the river, and the number of organisms present in the river itself. The System Identification models used in this thesis mainly deal with linear models, but non-linear grey box models were also proposed. Starting with a choice of three different black-box linear MIMO models, experiments were carried out to find a model which can best describe the Portneuf River ecosystem. Also, a major contribution of this research was to develop a means of making use of irregularly spaced temporal data, to produce regularly sampled temporal data. Our best linear time-invariant models were found to be those produced using transfer function estimates on weekly sampled data. Upon successful completion, this research might be very beneficial in protecting the river ecosystem and the animals living in and around it.

Keywords: System Identification; Linear Model, Non-Linear Model; Process-Based Model; Data-Based Model; River Ecosystem; Black-Box Model; Grey-Box Model

CHAPTER 1- INTRODUCTION

1.1 Background and Motivation

"I do not know what I may appear to the world, but to myself, I seem to have been only like a boy playing on the sea-shore and diverting myself in now and then finding a smoother pebble or a prettier shell then ordinary, while the great ocean of truth lay all undiscovered before me."

Sir Isaac Newton

(January 4, 1643 - March 31, 1727)

Human civilization has evolved from antiquity to the present day. Ever since the dawn of the time there has been cause and effect, action and reaction, in order to understand the world that we are living on. Throughout this journey of evolution, numerous tools of machine and machinery have been utilized. In today's world machines have been a key part of the human life. Although our society has been using machines on large scale for almost two centuries, up until now most of them have been completely designed or programmed in advance to perform a specific task in specific environment [1].

The systematic effort of building and organizing knowledge for the better understanding of and prediction of the universe and producing more accurate natural explanation of different phenomena has been known as science. The main emphasis of science has been understanding the deep knowledge about every system that comprises nature. There might be some explanation of some systems but there is also some hidden knowledge that is constantly coming out through different research. For instance, most physical and natural systems are bonded with some hidden set of rules and there is a dynamical system representation of these rules [1]. When the underlying dynamics of the system are not known, it becomes necessary to use some techniques that can discover these underlying or hidden sets of connecting structures. This field of study is called System Identification and Modeling [2].

In Control Theory, much knowledge and a deeper understanding is essential to describe a system which is usually done using a set of mathematical models employing the information obtained from scientific research. But sometimes when a system lacks the information for constructing a suitable mathematical model, the system can be a subgoal itself [1]. The main purpose of System Identification is to find a suitable mathematical model of a mechanical system and possibly use a control theory to control the system. In this thesis, System Identification has been used to find a suitable mathematical model of the Portneuf River using all the data that have been provided by the Idaho Department of Environment Quality.

The early work in System Identification was developed by the statistics and time series communities. It has its roots in the work of Gauss (1809) and Fisher (1912) and the theory of stochastic processes [2]. We can find this in an excellent survey conducted by Deistler in 2002 [2]. The Kalman's key paper started the model-based control era (Kalman: 1960a,b) [1]. For describing and analyzing systems which are presumed to be linear and time invariant, Kalman advocated the use of a linear state space description of the dynamics of the system.

We assume the system is observed at times t ∈ {0,1, 2, ….} (Which can denote multiples of a sampling periods Ts), and we assume the internal state of the system at time (t) is an (n x 1) vector (x_t). For multiple input, multiple output systems (MIMO), having (k) inputs and (p) outputs, we let u_t be the (k x 1) vector of inputs at time (t), and (y_t) be (p x 1) vector of outputs at time (t). The Kalman showed that the linear time invariance assumption demands that the dynamics of the system must be of the form:

$$\begin{cases} x_{t+1} & = & Ax_t + Bu_t \\ y_t & = & Cx_t \end{cases}$$

For some (non-unique) coefficient matrices A_{nxn}, B_{nxk}, and C_{pxn}. In some systems, we know these matrices from physical modeling, but for most system, these matrices must be identified by input/output data.

The development of this model-based theory for prediction, filtering and control using Kalman's approach replaced the Wiener filter, i.e., pole placement and LQG control. Ho and Kalman in 1965 effectively constructed a linear state variable model from input and output data which gave birth to realization theory, also known as subspace identification [2]. In the same year Astrom and Bohlin gave a numerical identification of dynamic systems from normal records which gave birth to prediction error identification. Since 1965 till 2000, two different System Identification approaches were developed, which are the parametric prediction error approach, and the non-parametric state space approach [2]. Also, a frequency domain approach has been developed, showing the advantage of using periodic excitations for exploring the behavior of a system. By this time, model predictive control has become the standard for most control applications. In the late 2000's, there was progress in the field of nonlinear System Identification as well [2].

Still today, modeling and identification is still the most difficult and costly part of implementing any advanced control system. The major problem till today still has been finding the best model structure. System Identification for multiple input and multiple output systems is still a very difficult task. Non-linear System Identification is still a very difficult task to complete. The major goals of present research in the field of System Identification involves in reducing the human intervention needed and making the System Identification user-friendly [1]. Some other goals are reducing the experiment time and performance degradation during data collection, cost complexity and application-oriented experiment design [3]. The research areas that are active today in the field of System Identification are improving the quality of estimates, nonlinear System Identification, and large distributed and network-controlled system, etc [3].

1.2 Problem Statement and Scope

Time series are said to be regularly spaced if the data are taken continuously in exact interval of time or if the observation of the data are uniform [3]. When the data are sampled at different times and the time elapsed between two consecutive observations changes then the sampled data are called irregularly sampled data [4]. Usually when the data are recorded using mechanical devices or sensors they are regularly spaced. When they are taken manually then they are mostly irregular. However, sensor failures can also give rise to irregularly sampled data.

In this project, the data considered to do research on was taken manually at different monitoring stations on the Portneuf River. Sometimes due to human error or sometime due to many other natural reasons, the data taken were irregularly sampled. Due to this a major part of this research was to try to use irregularly sampled data to produce regularly sampled data estimates and then resort to standard estimation techniques. Unfortunately, concepts like nonlinearity, noise and high dimensionality of the river system's internal state description make our task challenging.

The first part of the problem is to describe the problem mathematically. This might raise a concern of how we really describe mother nature and how she is affecting her entities? But like every other natural phenomenon, there should be some chemical and mechanical explanation that describes how changes are happing in the river ecosystem. This part of the problem and the steps taken to do it will be briefly explained in chapter 3 of this thesis.

The second part of the problem is to find those functions and variables that correctly describes the system. In this thesis the relevant input and output variables are chosen with the knowledge of chemical and physical processes when the environment is acting in a certain way. This part of the problem will also be explained more detailly in chapter 3 of this thesis.

The third part of the problem is to find a suitable model to explain the whole scenario of the system. This part should include a suitable mathematical theory to explain the system. Using state space models or transfer functions models, is it possible to explain how the observed system is behaving? This part of the problem and the steps taken to solve it will briefly be explained in chapters 5, 6, and 7 of this thesis.

The use of different control theories used in this thesis is different than in most system-theoretical literature. Controlling the system is not the goal here, rather control theory is an aid for the System Identification process. This thesis will be able to explain how an irregularly sampled dataset could be handled, and how we can deal with a complex model of a System Identification. In one part, this thesis will explain the process involved in selection of the data. In the second part, it will explain the organization and the estimation procedure. Furthermore, it will explain the model selection process and the best estimation of the model.

The final part of the thesis will explain if the implementation of a chosen model will be applicable in real life or not. If the thesis is a success, then it could be used for other natural phenomena. This might hugely benefit humans and other life present in the earth. This will also be beneficial in predicting what might happen if mother nature start acting in a certain way. This thesis can be a milestone for some nature researcher to explain how global warming is happening and if it is not controlled, then what might happen in another ten or fifteen years.

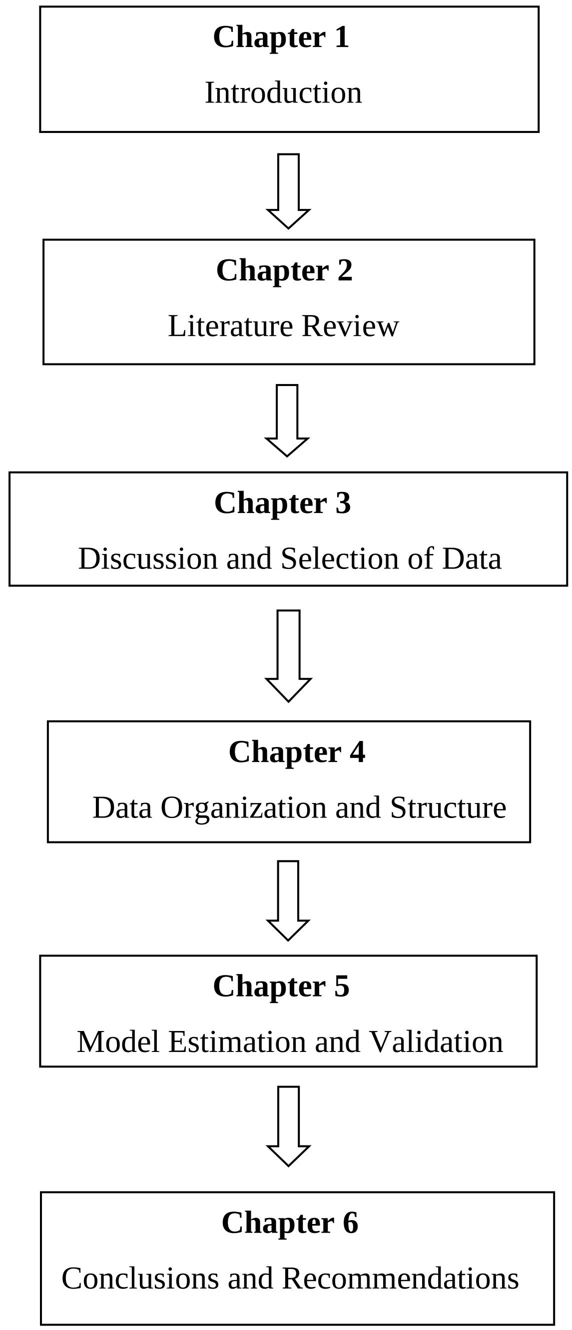
1.3 Objective

The objectives of this research are:

1. Literature review of past research and theoretical review of System Identification.
2. Description of one data selection process involved in System Identification. Explanation of data organization and the structure of the data.
3. Construction and determination of models that could match data estimation and evaluation.
4. Model evaluation and recommendations for future research.

1.4 Thesis Structure

The structure of the thesis is presented as a flowchart in figure below.



- Chapter 1- This chapter gives a brief introduction of the research. This chapter focuses on the background and motivation, scope, and objective of the thesis.
- Chapter 2- This chapter includes a literature review of past research on the use of System Identification using different models.
- Chapter 3- This chapter includes the discussion about data selection.
- Chapter 4- This chapter includes the steps that were taken to organize data and includes the discussion about the model structure.
- Chapter 5- This chapter includes detailed steps that were taken for estimating and validation the model.
- Chapter 6- This chapter includes results and conclusion from the research. This chapter also provides recommendations for future research on System Identification using irregularly spaced data.

CHAPTER 2. LITERATURE REVIEW

The literature review will investigate the research done earlier on the topics related to this thesis project. Constructing models from observed data is a fundamental element in science. Several methodologies and nomenclatures have been developed in different application areas. In the control area, the technique is known as System Identification. Several studies have been done on this topic to understand the nature and significance of System Identification.

2.1 Perspectives on System Identification

One of the most important elements of System Identification is to have a brief knowledge of a system whether it is mechanical or natural or chemical. System Identification is the art of defining a system from observed input and output data [6]. It can be seen as the interface between the real world of application and the mathematical world of control theory and model abstraction [6]. System Identification is a very large topic, with different techniques that depend on the character of the models to be estimated: linear, nonlinear, hybrid, nonparametric etc.

This part of the thesis will provide a subjective view of the state of art of System Identification. Due to the many subcultures in the general problem, it is very difficult to see a consistent and well-built structure. The review paper from Dr. Lennart Ljung contains a fundamental result of statistical nature around the concepts of information, estimation (learning) and validation (generalization) [6]. This paper starts with explaining the core of estimating models. This research has explained that a model can predict properties or behaviors of any system which basically is a mathematical expression but could also be a table or a graph. Realistically, it can be very difficult to achieve a true description of a system to be modeled, however it is sometimes convenient to assume a description as an abstraction. This contains some character of the model, but it is usually very complex.

After the true description of the system, model class comes into play according to the paper. A model is basically a set that can be parameterized by a finite-dimensional parameter which could be a linear state-space model or a linear transfer function model, but it may also have non-constant parameters that are piecewise continuous [6]. After model class, a measure of the size and flexibility of a model class is important. This will dimension the vector which will parameterize the set in a smooth way [6]. After model class and measurement of complexity, estimation will come into play [6]. According to Dr. Ljung, the process of selecting a model which is guided by the viable information is called estimation. The data selected for estimating the data is called estimating data. Also, the process of ensuring if the model is useful not only for estimating data but also for the dataset of interest is called validation of the data. Finally, the scalar measure of how well a particular model can explain or fit to a particular set is called model fit [6].

In [6], the author considers an unknown function g(x) for a sequence of x-values $\{x_1, x_2, ..., x_N\}$. This will give a corresponding value of function with a noise:

$$y(t) = g_0(x_t) + e(t)\,, \quad t \in \{1, …., N\}$$

The problem is to construct an estimate

$$\hat{g}_N(x)$$

from

$$Z^N = \{y(1), x_1, y(2), x_2, \dots, y(N), x_N\}$$

This is a well-known basic problem that many people already encountered [6]. In this problem x is a vector of dimension n. This means that g defines a surface R^{n+1} if y is scalar. Then this problem can be seen as a curve or surface fitting problem. Ljung points out that there are two ways we can approach this problem: parametrically and non-parametrically. In the parametric approach, if the model set is of n^{th} order then we can parametrize it by n+1 coefficients θ to minimize the least square fit between y(k) and $g(\theta,x_t)$ [6]. In a nonparametric approach for each x, a weighted average of neighborhood y(k)'s is taken, and the complexity could be the size of the neighborhoods. The smaller the neighborhoods, the more complex/flexible curve or surface can result.

All the datasets in every survey contains both useful and irrelevant information. Irrelevant information is typically denoted by noise in System Identification [6]. In order not to get fooled by irrelevant information, data should be passed through some sort of filter or prejudice. The conceptual process for estimation becomes:

$$\hat{\mathfrak{m}} = \arg \min_{\mathfrak{m} \in \mathcal{M}} \left[\mathcal{F}(\mathfrak{m}, Z_e^N) + h(\mathcal{C}(\mathfrak{m}), N) \right]$$

In this equation (F) is a measure of fit to the data, and (h) is a penalty term that penalizes over-fitting or over-parametrizing the model based on the complexity of the model (m) or the corresponding model set (M) and the amount of data.

Now the model should show good agreement with the estimation data and the model shouldn't be too complex [6]. Since the information or the data is typically described by a random variable, does model also become a random variable? The answer is yes, the model can also be treated as a random variable according to Dr. Ljung. The above equation has a flavor of a parametric fit to data. However according to Dr. Ljung, with a conceptual interpretation it can also be described as parametric modeling. It is almost like when a model is formed by kernel smoothing of the observed data [6]. Now the problem can also be treated as a curve fitting problem:

$$V_N(\theta, Z_e^N) = \sum (y(t) - g(\theta, x_t))^2$$
$$\hat{\theta}_N = \arg\min_{\theta} V_N(\theta, Z_e^N) + \delta\|\theta\|^2$$

Here, $g(\theta, x_t)$ is the output at time (t), state (x_t) and parameter (θ).

This is known as regularized least square according to Dr. Ljung. It is not very difficult to find a model that describes estimation data well. If we use any flexible model structure, it is always possible to find something which best describes the data. The real test is when the model is subjected to a new dataset [6]. For a conceptual form: let a model $\hat{m}$ be estimated from estimated dataset in a z_e^N model set M, then.

$$\bar{\mathcal{F}}(\hat{\mathrm{m}}, Z_v) = \mathcal{F}(\hat{\mathrm{m}}, Z_e^N) + f(\mathcal{C}(\mathcal{M}), N)$$

In this equation, the left-hand side denotes the expected fit to validation data, while the first term on the right is model's actual fit to estimated data. In almost every case the estimated fit is typically measured as the mean square error. The quantity f is a strictly positive function which increases with the complexity C and decreases with the number N of estimation data. Hence, quality of an estimation can be adjusted to the complexity of the model [6]. The more flexible the model set, the more adjustment is necessary. For the simple cure fitting problem with d being the number of degrees of freedom in the model, the above expression can be taken as a well-known form of:

$$\begin{aligned} E\bar{\mathcal{F}}(\hat{\mathrm{m}}, Z_v) &\approx \frac{1 + d/N}{1 - d/N}\mathcal{F}(\hat{\mathrm{m}}, Z_e^N) \\ &\approx (1 + 2d/N)\mathcal{F}(\hat{\mathrm{m}}, Z_e^N) \\ &\approx \frac{1}{(1 - d/N)^2}\mathcal{F}(\hat{\mathrm{m}}, Z_e^N) \end{aligned}$$

Where the left-hand side is the expected fit when applied to validation data. The first expression is Akaike's final predication error (FPE), the second one is Akaike's information criterion (AIC) when applied to the Gaussian case, (Akaike, 1974), and the third one is the generalized cross-validation (GCV) criterion, (Craven and Wahlba, 1979). Here the dimension d serves as the complexity measure of the model set. These expressions are derived with expectations on both sides from Dr. Ljung's paper [6]. But there is an exception that this is typically used to estimate the quantity on the LHS, and the expectation on the RHS is replaced with the observed fit which is also known as the empirical fit. When the regular criterion is used, d in the above expression is replaced with:

$$d^*(\delta) = \sum_{k=1}^{d} \frac{\sigma_k}{\sigma_k + 2\delta}$$
$$\sigma_k = \text{the singular values of } V_N''(\theta, Z_e^N)$$

Here the complicity measure C is the Vapnik-Chervonenkis (VC)- dimension of the model set, which basically measures how well the functions can separate random points from the initial dataset [6]. The main conclusion that we get from this is that one should not be so impressed by a good fit to estimation data, if the model has been quite flexible according to Dr. Ljung.

Now, if we assume that there is a true description S, we can conceptually write the model error

$$S - \hat{m}$$

In this equation we can interpret S and $\hat{m}$ to be any scalar property of an object, like the static gain of a dynamical system. The mean square error (MSE) is:

$$\begin{aligned} W &= E(\mathcal{S} - \hat{\mathrm{m}})^2 = (\mathcal{S} - \mathrm{m}^*)^2 + E(\hat{\mathrm{m}} - \mathrm{m}^*)^2 \\ &= B + V \\ \mathrm{m}^* &= E\hat{\mathrm{m}} \end{aligned}$$

where B denotes the bias contribution and variance error V. According to Dr. Ljung this is a very elementary and well-known relation but still worth some contemplation. There are elementary expressions for this, but it is sufficient to realize that the wider the model class used, the more susceptible the model will be to picking up random misinformation in the data. This means we shouldn't strive for the truth, but for a reasonable approximation according to Dr. Ljung. This means that it is beneficial to shrink the model set as much as possible using physical insights into the nature of the object. This is called grey box modeling in controls.

In structuring mechanical systems, the art and technique of building mathematical models is often considered crucial [6]. Many application areas rely on the skill and technology of creating mathematical models of (dynamic) systems. As a result, several scientific communities are working on theory and algorithms. Except for a few instances, this has occurred in a surprising number of different and isolated locations. With their own journals and conferences, they have created their own habitats. As a result, we witness distinct subcultures inside the larger problem. According to Dr. Ljung, mathematical statistics and time series analysis is the "mother" field of System Identification. He goes to say that statistics is clearly a broad field, and it is not meaningful to give terse summary of recent trends [6].

In this paper, Dr. Ljung has also done the analysis on econometrics and time series. The science of extracting information from economic data, considering both the special features of such data and a-priori information coming from economic theory is called econometrics [6]. According to Dr. Ljung, econometrics has a long tradition of giving inspiration to time series and difference equation modeling and its roots coincide with developments in statistics.

The work on time series dates to Jevons (1884), Yule (1927), and Wold (1938). The classic paper by Mann and Wald (1943) developed the asymptotic theory for the LS estimator for stochastic linear difference equations (AR systems). The results were extended to simultaneous (multivariate) systems, where LS is not consistent, in Koopmans et al. (1950), where also central identifiability issues were sorted out and Gaussian Maximum Likelihood estimates were proposed and analyzed. Important extensions to the ARMA(X) case have been proposed by Anderson (1971), and Hannan (1970) later. The problem of errors-in-variables modeling (when there are disturbances on both independent and dependent variable measurements) also has its origins in econometrics, (Frisch, 1934). More recently, important focus has been on describing volatility clustering, i.e., more careful modeling of conditional variances for modeling and forecasting of risk (GARCH models, (Engle, 1982)), as well as on describing non-stationary behavior of interesting variables in terms of a common stationary linear combination ("cointegration"), (Engle and Granger, 1987), which gives the long run equilibrium relation between these variables [6]. These two subjects were in focus for the Sveriges Riksbanks Prize in Economic Sciences in memory of Alfred Nobel in 2003.

Similarly, System Identification is the term that was coined by Zadeh (1965) for modeling estimation problems for dynamic systems in the control community. Two main avenues can be seen for the development of the theory and methodology (Gevers, 2006): One is the realization avenue, that starts from the theory how to realize linear state space models from impulse responses, Ho and Kalman (1966), followed by Akaike (1976), leading to so-called subspace methods, e.g., Larimore (1983) and Van Overschee and DeMoor (1996). The other avenue is the prediction-error approach, more in line with statistical time-series analysis and econometrics. This approach and all its basic themes were outlined in the pioneering paper by Åström and Bohlin (1965). It is also the main perspective in Ljung (1999). According to Dr. Ljung and the paper on his perspective of System Identification [6], the distinguished features on the efforts in System Identification are described as follows.

- To describe linear and non-linear dynamic systems, inventing parameterizations is the most importing thing. For underlying state-space realizations, realization theory has been an important source of inspiration. Having prior physical knowledge is the best way to start any System Identification [6].
- Translating core materials into estimated systems as well as the estimation procedure is also key [6].
- Choosing different effective ways to parameterize a model is also key for any System Identification. Some of the recently developed techniques such as SVD and QR factorization can also be used for the realization avenue. Also, the factorization of noise that can effectively reduce the model prediction error should be considered crucial [6].
- Experiment design now becomes the selection of the input signal. Can core material evaluation can be given concrete interpretations in terms of model quality for any mechanical control design, e.g., Gevers (1993) (6)? Specific features for control applications are the problems and opportunities of using inputs, partly formed from output feedback, e.g., Hjalmarsson (2005). An important problem is to quantify the model error, and its contribution from the variance error and the bias error, cf. (11), "model error models", e.g., Goodwin et al. (1992) [6].

In [6] Dr. Ljung has also discussed the use of System Identification in the industrial environment. Sometimes the gap between theories and practice can be different but use of sophisticated identification methods could be very beneficial [6]. The main problem is that the record of many un-useful data that has been stored in the system. Also, some missing time values has been the biggest problem in the industrial use. Dr. Ljung has also given a suggestion of using different data filtration methods to store only the useful data required for analyzing a system. For engineering industries, certain structural information can play a crucial role in System Identification [6]. Modern control theory also suggests us to take a multivariate view of the process and treat multiple inputs and outputs simultaneously into consideration. Sometimes, a simple process cannot incorporate all the things that are going into the system. Also, failure detection and predictive maintenance can be very difficult by using too simple of a model of a System Identification according to Dr. Ljung.

In [6], Dr. Ljung has tried to sketch a clear picture of where System Identification stands, and the main message is that much more interaction between communities around the core could be very beneficial. He has also pointed to some of the problems regarding theory and industrial practice where progress means a big step forward in the field of System Identification.

2.2 Summary

The literature review inspected many studies and offered relevant insight into the current study. Many researchers are conducting research on linear and non-linear System Identification and its application [10]. The prediction of the behavior of complex systems is essential in many fields such as weather forecasting, the motion of the planets, and modeling chaotic systems. Philosophers and scientists have tried to formulate observational models and infer future states of such systems [5]. Constructing an underlying mathematical model which can be applied as the predictor is the basis for many scientific predictions. For example, the existence of the planet Neptune was predicted through many mathematical modeling, not by observation. In 1821, Alexis Bouvard published astronomical tables of the orbit of Uranus, and following observations revealed deviations from the tables, which led Bouvard to a hypothesis that an unknown body was perturbing the orbit through gravitational interaction [12]. In 1846, Urbain Le published his estimate of the planet’s longitude, and in the same year, Neptune was discovered within 1 degree of where Le Verrier had predicted it to be.

Deep Learning (DL), Machine Learning (ML), and Artificial Intelligence (AI) are some of the very hot topics in today’s research world and often seems to be used interchangeably [10]. Researchers are trying to use deep learning for System Identification which itself is a subset of machine learning and machine learning is considered a subset of artificial intelligence. The application to which the identified model will be applied to can play a role in examining the quality and robustness of the model [14]. Similarly in this thesis, our main purpose is to find a model that provides a good enough prediction about how the Portneuf River ecosystem works regardless of whether this model is identical to the true system or not. The research also includes different mathematical models that were used to do System Identification of the river ecosystem.

CHAPTER 3. DISSCUSION AND SELECTION OF DATA

3.1 Introduction

This chapter presents the discussion and description of the data that was selected for the research carried out for System Identification of the Portneuf River. Section 3.2 describes the geographical location and importance of the river for east Idaho. Furthermore, we will present the historic importance of the river as well. In section 3 we will describe the different stations where the data were recorded and how were they recorded. Furthermore, we will present a brief discussion about the data structure.

3.2 The Portneuf River and its Geographical Location

The Portneuf River is 124 miles long and located at 4357 feet above sea level and is also a tributary of the Snake River in southeastern Idaho and a part of the Columbia River basin. It was named sometime before 1821 by French Canadian voyageurs working for the Montreal-based fur-trading Northwest Company [18]. According to historians, the Portneuf valley used to provide the route of the Oregon Trail and California Trail in the middle of 19th century [18]. After a series of heavy floods in the early 1960s, the Army Corps engineers designed and constructed a concrete channel on the portion of the river flowing through Pocatello to control floods in 1965 [18]. The channelization approximately followed the Rivers' route and cut through the west side of Pocatello, drastically altering the natural river processes. The river is subjected to use by Lava Hot Springs, McCammon, Inkom, and Pocatello [17]. Due to the heavy use of river for local purposes, it has a unique set of chemical characterizations associated with both biological processes and the interactions with local geology [17].

The Portneuf watershed drains almost about 850,290 acres in southeastern Idaho and is bounded by Malad Summit to the south, the Bannock Range to the west, the Portneuf Range to the southeast, and the Chesterfield Range to the northeast [19]. From its headwaters, it flows initially south, passing westward around the southern end of the 60 miles of the Bonneville/ Portneuf Range. It then turns north to flow between the Portneuf Range to the east and the Bannock Range to the west. It flows northwest through downtown Pocatello and enters the Snake River at the southeast corner of American Falls Reservoir, approximately 10 miles northwest of Pocatello [18]. The major tributary to the Portneuf River is still considered to be Marsh Creek, however watersheds including Mink, Rapid, Garden, Hawkins, etc. are also considered its other tributaries. The mean annual discharge as measured by USGS gauge is 418 cubic feet per second, with a maximum daily recorded flow of 1730 cubic feet per second [18]. Nitrates, phosphates, and calcium compounds are some of the main chemical contents found in the Portneuf River. The flow of the Portneuf River is given below in figure 1.

3.2.1 Importance of the Portneuf River

The cities of Pocatello, Chubbuck, and Inkom rely on ground water from the lower Portneuf River’s aquifer. The river water is important for all of their drinking, commercial, and industrial water needs [20]. As Idaho is also one of the main agricultural hubs in the arid intermountain region, river water is also used for various agricultural purposes. The Portneuf River is also home for a number of species of fish including both game and so-called rough fish which plays an important part in summer tourism [21]. Different species of aquatic life in the river have played an important role in surrounding ecosystem. The Portneuf River is a key element of the Portneuf watershed and provides flood control, wildlife habitat, and aesthetic amenities to the community [20]. The river is also key to the regional economy providing may recreational activities during summer. Many current residents, while appreciative of the flood protection afforded by the river channel, would also like to reconnect with their river, accessing it often and in many ways [19]. One of the main goals of this thesis was to find an appropriate model of the system which will aim at promoting ecosystem restoration in the long term.

Figure 1. Flow of Portneuf River [20]

3.3 Discussion of Data

Due to the growth of technology in the modern era, operational data from river systems are often now available on the internet or some specific software. However, the data used for this research was provided by the Idaho Department Environmental Quality (IDEQ) and the Idaho Department of Water Resources (IDWR) [17]. IDEQ and IDWR has played an important role in the planning and initiation of the Portneuf River canal. Officially named as the Idaho Department of Water Resources in 1974 after the merging of Department of Water Administration and Idaho Water Resource Board, both the IDWR and IDEQ were equally instrumental in contributing to the timely and successful monitoring of Portneuf River. Playing a huge factor in protecting wildlife in the Portneuf River, both have also played an important role in conserving all the watersheds that contribute to the river [19].

In this part of chapter 3, we will discuss the data that we received from Idaho Department of Environmental Quality. IDWR and IDEQ have been managing and allocating water resources as required by statute to optimize economic activity and protect public safety [20]. Both have also played an important role in promoting and financing projects that will advance the sustainability of water sources into the foreseeable future, and that will optimize the use of water of the State of Idaho. IDEQ provided us with environmental data to analyze the river system. If the research is successful, it could be a boon for the Portneuf River providing protection to the river water quality and all the animals that habitat in the Portneuf River.

The data provided to us by Idaho Department of Environmental Quality contains information ranging in time from 1997 to 2018. By setting up different stations along the Portneuf River, the data were collected nearly every month to analyze the health of the river. The “Jimmy drinks” station was setup near the mouth of the river, whereas the Batise Spring station is located below the Hatchery raceways. Similarly, the Edson Fichter station is in the Edson Fichter Nature Area and other stations are situated along the river. These stations attempt to collect the data from the river on a monthly basis. Some data were measured on the river and some of the data were measured using samples analyzed in the lab. The data that were collected from the river and laboratory measured data are given is the schematic figure below (from the dataset provided by IDEQ).

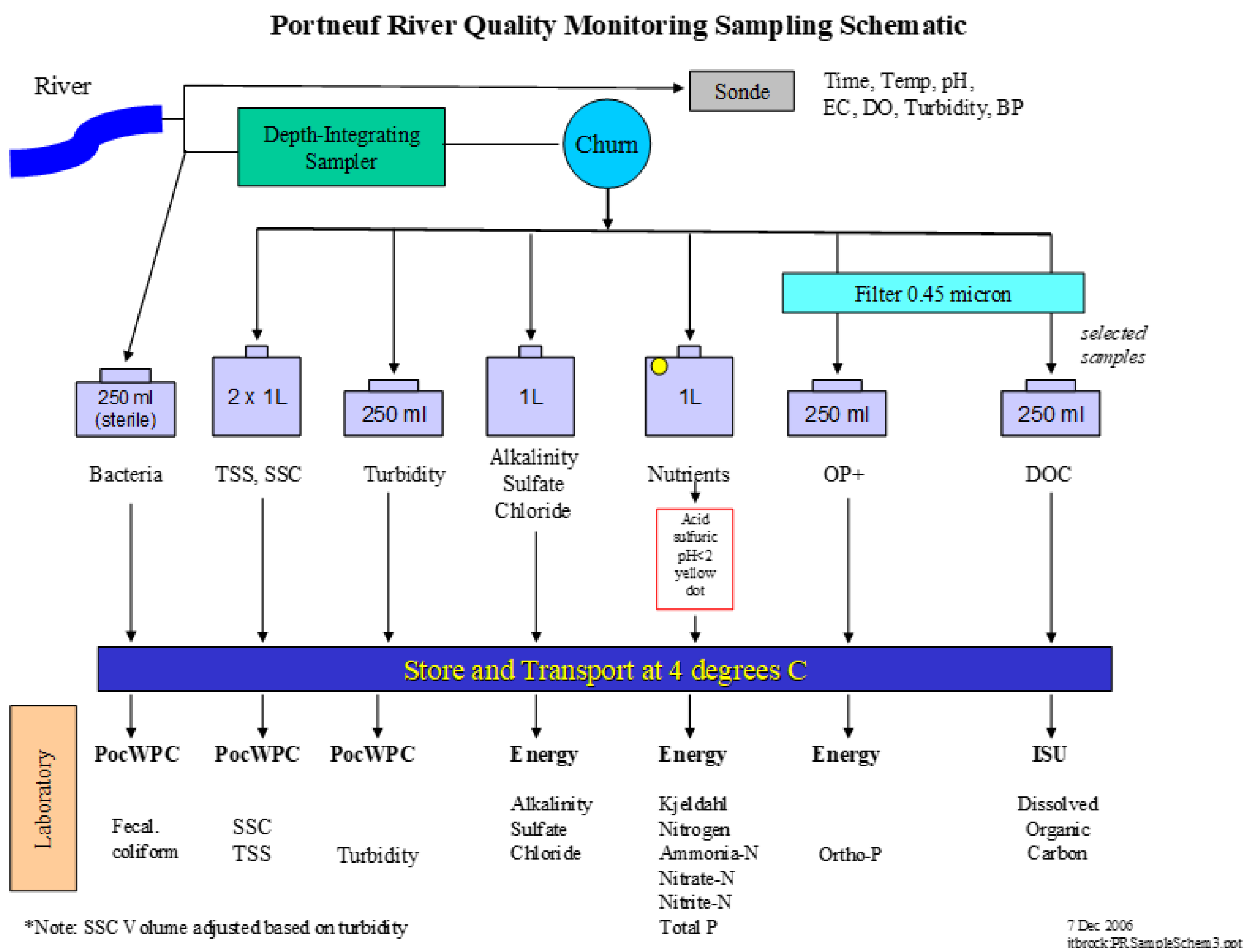


Figure 2. Portneuf River Quality Monitoring Sampling Schematic

All stations had a multiparameter water-quality sonde and a flow rate monitor. The sondes are usually equipped with temperature, conductivity, pH, dissolved oxygen (DO), and turbidity sensors. These sensors are evaluated for compliance with the U.S. Geological Survey (USGS) national field manual for the collection of water-quality data criteria for continuous water-quality monitors. To monitor other parameters of the river water like bacteria, lab turbidity, chloride, alkalinity, and nutrients, samples from the stations were analyzed in a lab.

3.4 Selection of Data

The data provided to us by IDEQ ranged in time from 1997 to 2018. Most of the data that were collected from the river were on a nearly monthly basis. Due to many circumstances, the data was not regularly taken and hence we have irregular sampling of the data. After receiving the data, the first and foremost thing to do was to analyze the data to see which station has the most complete datasets. Even though the Simon station has the most data compared to other stations, Fichter Station has most extensive record of the data. The recorded data received from the Fichter Station didn’t have many noticeable irregularities. The dataset from the Fichter Station ranges from 2000 to 2017 and also it has all the parameters that we need for the System Identification of the river. We decided to focus on with the Fichter Station dataset. The only motive to do this was to analyze the dataset from Fichter and then apply in future work analysis to other stations if it was successful.

CHAPTER 4. DATA ORGANISATION AND STRUCTURE

4.1 Introduction

The main goal of this thesis is to attempt to perform a MIMO input/output-based system identification of the temporal dynamics of the water quality data at any of the monitoring stations along the Portneuf River. As the physical, geological, riparian, etc. characteristics of the Portneuf vary along its course, we wouldn't expect that a model that describes what happens at one station will be appropriate at another, but we do suspect that the modeling approach will be.

As stated in the previous chapter, the most complete dataset provided to us was the Fichter Station's. So, we will hence go for the focus on modeling the dynamics at Fichter Station.

To perform input/output System Identification, we first need to specify the input variables of interest, and the output variables. Inputs are typically those variables which have an effect on the overall behavior of the system, especially on other variables of interest.

In controls, inputs can usually be manipulated (i.e., "controlled'). However, in our system this is only partially true. We chose as inputs to our model the flowrate of the river (which can be partially manipulated), and the ambient air temperature (which cannot be manipulated). The reason for this choice is that these two variables have known physical and biological effects on the remaining measured variables.

For outputs, we chose as a first pass on modeling, the river's water temperature and the dissolved oxygen concentration. These two variables are known to be important in the biological health of the aquatic system. If this work succeeds, the other relevant input/output variables can be considered.

The novel aspect of this work is that we are attempting a dynamic system description of our system, as opposed to finding a multivariate static correlation statistical model. Our approach could give water quality mangers a means of improving the health of the river system by manipulating the flows.

The ambient air temperature data was not part of the data provided to us by Idaho Department of Environmental Quality (IDEQ). We acquired daily ambient temperature from the National Oceanic and Atmospheric Administration (NOAA) for Pocatello, Idaho, for the data we needed.

As for the rationale for our choice of inputs and outputs, we clearly expect the flow rate of the river to have an effect on the remaining variables that were measured. E.g., slower flows allow the river water to heat up or cool down faster, higher flows can cause an increase in dissolved oxygen concentrations (indirectly), by raising or lowering the water temperature. So clearly, flow rate and ambient temperatures should be considered as inputs. As the water temperature and dissolved oxygen concentrations are influenced by the ambient air temperature and flow rate, they are reasonable choices for output variables. The other variables in our dataset could be classified in a similar fashion: are they inputs (somewhat controllable), or are they outputs (the result of the inputs)?

In the remaining portions of this chapter, we will address data extraction, and the data processing needed to utilize the standard system identification tools in MATLAB and other packages.

4.2 Data Organization

As we decided to proceed with ambient temperature and flow of the river as the inputs and water temperature and dissolved oxygen of the river as an outputs, the first problem was to extract those data into MATLAB, since MATLAB has all the tool needed for doing System Identification. First, we changed the date of the recorded data into Julian date. Julian date is the continuous count of days since January 1,1900. Since we had data, which was sampled roughly once in every month, and had the actual sampling dates, converting the date into Julian date helped in calculating the elapsed time between two samples. After converting all the dates into Julian dates, we used a simple piece of code to extract all the information needed for us into MATLAB. The code used to extract all the required information into MATLAB is given below with explanation.

```
>>fichter=xlsread('fichter_water_data.xlsx','sheet1','A2:AQ7');
>>Date=fichter(:,1);
>>cms=fichter(:,5);
>>cms_c=~isnan(cms);
>>cms_cens=cms(cms_c);
>>Date_cms=Date(cms_c);
```

The first line of code basically extracts the file of excel data into MATLAB. It extracts the data from the Fichter water dataset from the column A1 to column AQ1. The second line of code gives the name to the first column from the excel sheet which was extracted to the MATLAB workspace. Similarly, the third line of code gives name to the flow rate of the river which is cubic meters per second (cms). The fourth line of code will remove all the missing data and will only provide the data that were sampled. Similarly, the fifth and sixth line of code will give the specific Julian date to the specific sampled data. Similarly, sampled data for temperature and dissolved oxygen of the water was extracted for specific dates.

4.3 Data Structure

Nearly all System Identification software routines available are based on having available discrete time input/output datasets sampled at a uniform rate (constant sampling period Ts). However, all of our data obtained from Idaho Department of Environmental Quality (IDEQ) is irregular: the sampling was approximately done on a monthly basis, but variations up to several days are present. Also, some monthly sampling is missing, e.g., due to winter conditions, broken equipment, absent field personnel, etc. For the Fichter Station, we have nearly 200 measurements dates over 15 years, but some are incomplete (missing one or more of our input/output data values). Hence, to proceed with System Identification on this system, we needed to produce an approximate input/output dataset, gotten from the data we actually possess.

The route we chose was to build a callable function for each input/output variable, so that we could produce regularly spaced temporal (approximate) data for any desired sampling period Ts. In statistics, this is known as “imputation”.

4.3.1 Smoothing Splines

Smoothing splines estimators perform a regularized regression over the natural spline basis, placing knots at all the points. It circumvents the problem of knot selection and simultaneously controls for overfitting by shrinking the coefficients of the estimated function. Smoothing splines are similar to kernel regression and k-nearest neighbor regression and provide a flexible way of estimating the underlying regression function. However, smoothing splines often delivers similar fits to those from kernel regression, but they also are in a mathematical sense similar. Both kernel regression and the smoothing spline have a tuning parameter. A bandwidth value for kernel regression, and the smoothing parameter for smoothing splines. Since smoothing splines are generally much more computationally efficient, we decided to go with the smoothing spline data approximation method.

Smoothing splines, also known as thin-plate splines, or splines under tension, are C^2 cubic splines which can be used to approximate noisy data. In [16], it is shown that given possibly noisy data $\{t_{i,} y_i\}_{i=1}^{n}$ on the interval [0, T], the smoothing splines with smoothing parameter λ ∈[0,1] solves the following infinite dimensional optimization problem:

$$\min_{f \in H_2([0,T])} (1-\lambda) \sum_{i=1}^{n} \left(y_i - f(t_i)\right)^2 + \lambda \int_0^T (f"(t))^2 \, dt$$

Here, it is assumed $t_1 = 0$, $t_n = T$, and λ is the “tension’ or smoothing parameter. When λ=0, there is no tension on the curve, and the resulting spline hits (interpolates) every data points (and has the minimal curvature of any C^2 function to do so). When λ=1, there is “infinite tension” on the curve, and the resulting spline degenerates to the standard least squares linear fit to the data. The problem arises as to finding the correct value of the smoothing (tension) parameter λ.

As λ grows, the spline moves away from interpolating every data point, but also loses extraneous wiggles or overshoots in between the data points. Too small of values for λ, the closer we are to hitting the data, but pay for this with overshoots. Too large of values for λ, the “flatter” the curves become, but the more data we miss.

Statisticians have proposed various means of automatically selecting the proper choice of the smoothing parameter $\lambda \in [0, 1]$. All are based on various statistical assumptions about the noise structure on the measurements, and all authors warn that sometimes physical insight should dictate the choice. That is, when working with a dataset with unknown statistical properties, choose a λ which gives reasonable results.

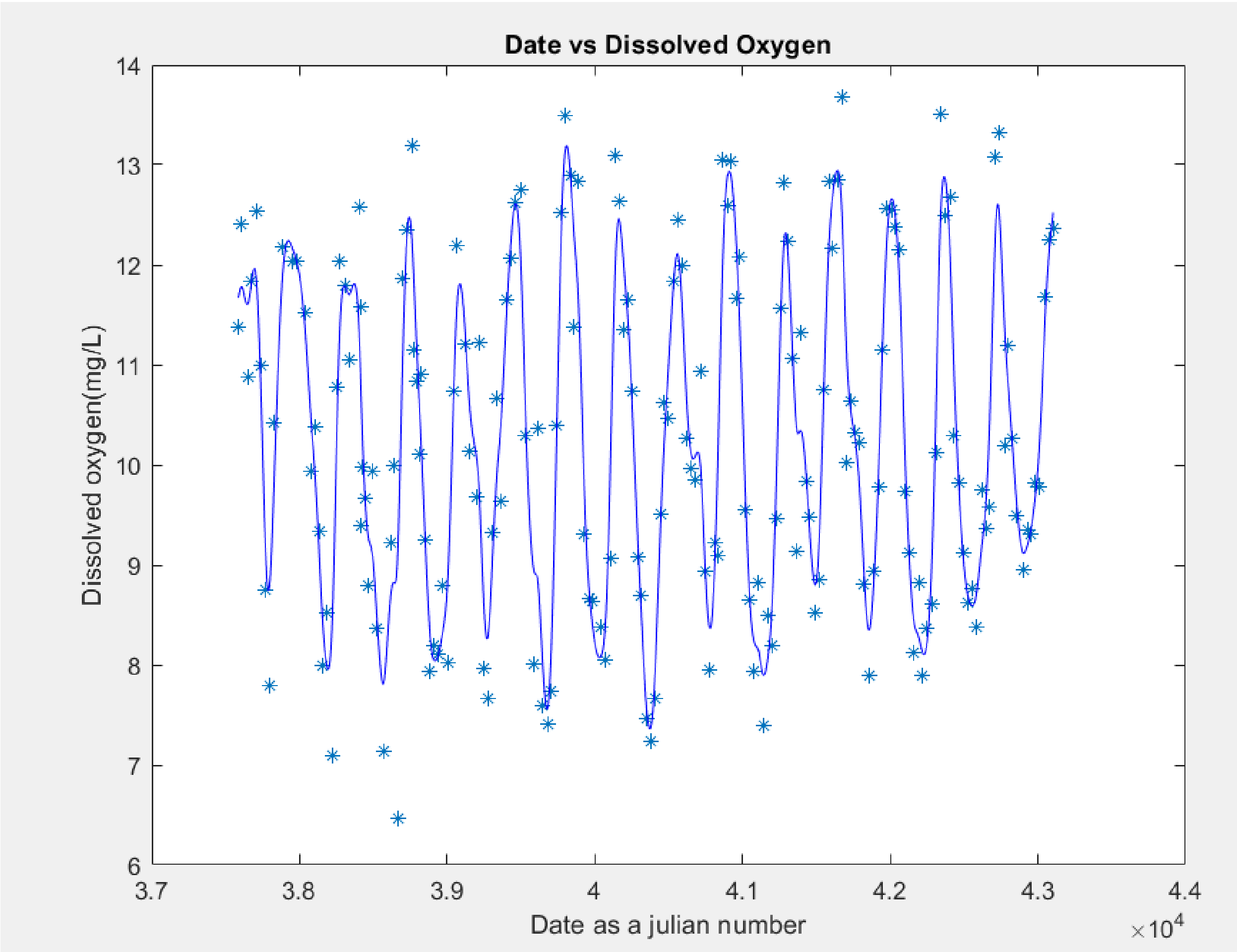


Figure 3. Using GCV as a Smoothing Spline

Generalized cross validation (GCV) is an approach that uses nonparametric regression. GCV is an asymptotic limit of Ordinary Cross Validation (OCV) [16]. When dealing with the choice of any parameter λ in an approximation problem, OCV states that one should choose the optimal values of λ as that which minimizes the overall effect on the fit due to leaving any one data point out. That is, one chooses the "optimal" λ so that no single datum has significant influence on the overall fit. GCV is gotten by taking $n \to \infty$ and assuming Gaussian noise structure on the data. See [16] for a derivation for this result. When we used GCV to select the smoothing parameter λ for our smoothing spline, it couldn't give us a proper curve as shown in the figure 3. The curve formed by using GCV is missing lots of data points especially maxima and minima, which is crucial for tracking dissolved oxygen.

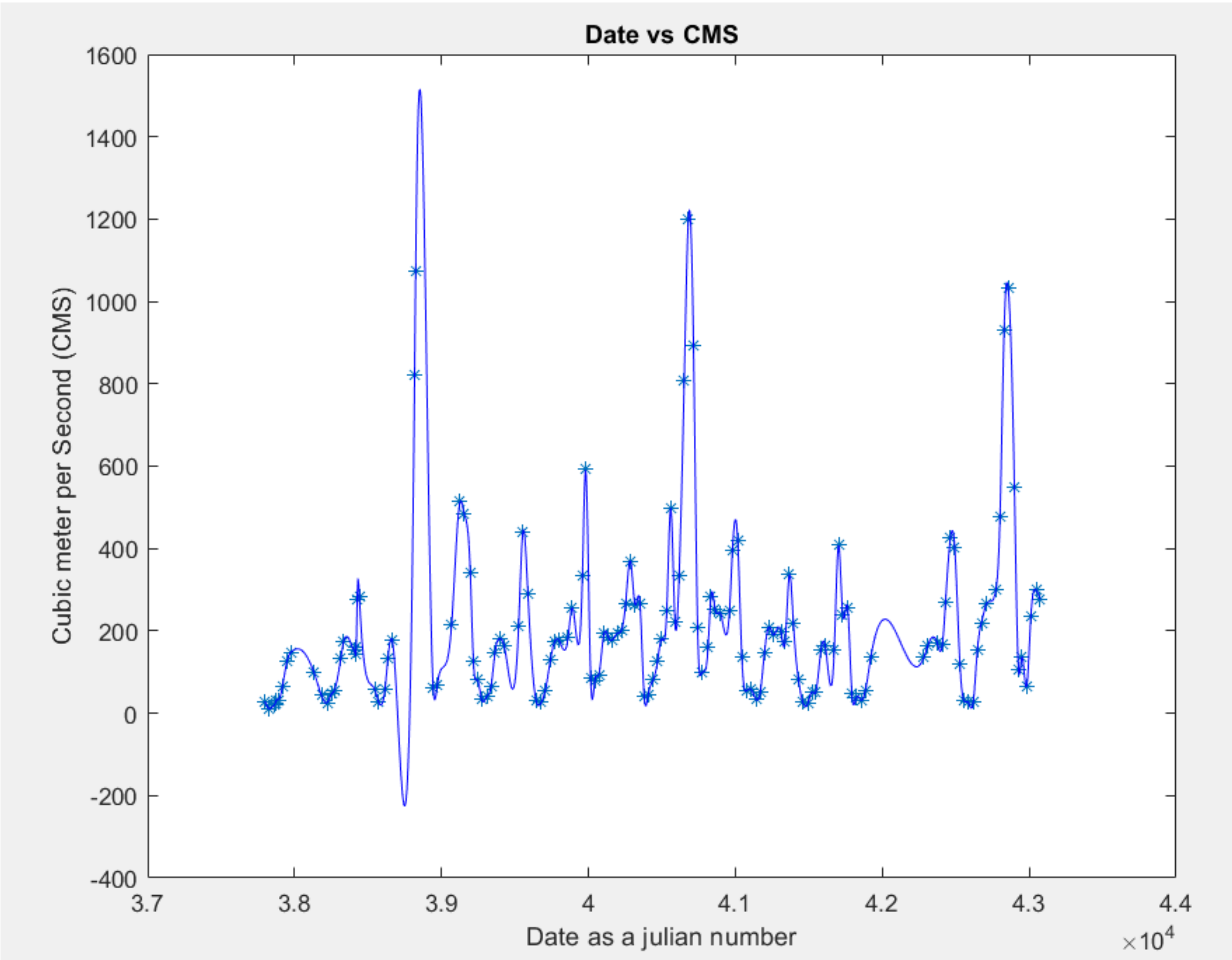


Figure 4. Using AIC as a Smoothing Spline

The Akaike Information Criterion (AIC) is based on information theoretic results in statistical modeling. Basically, AIC states that when dealing with the choice of any parameter λ in an approximation problem, one should choose the optimal value of λ as that which maximizes the information content of the data (interpolation), but balance this with the cost of over parametrizing the fit [6]. When we used AIC to select the smoothing parameter for our smoothing spline, we found lots of extraneous spikes were produced and the curve takes on negative values. Since the flow of the river nor the water temperature can never be negative, we didn't use Akaike information criterion (AIC) to select the smoothing parameter λ for our smoothing spline. See figure 4.

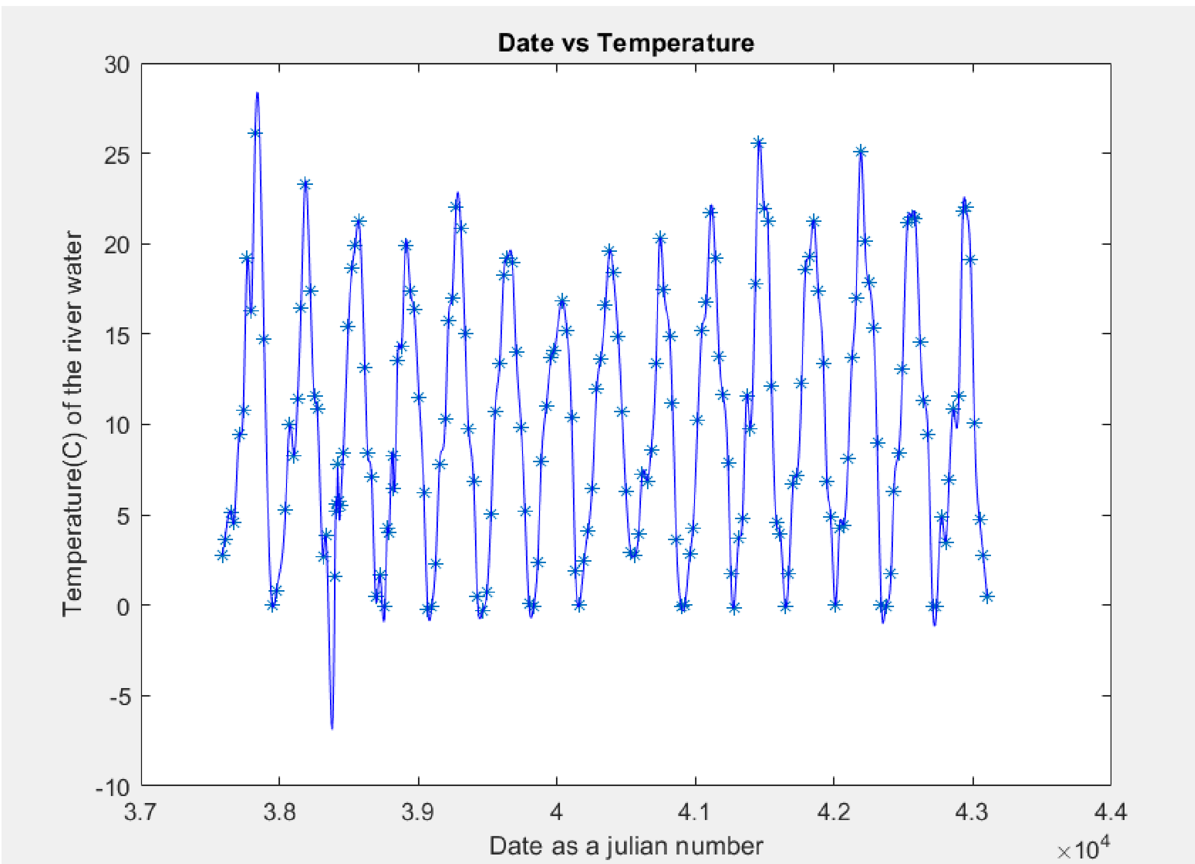


Figure 5. Using AICc as Smoothing Spline

The classical Akaike Information Criterion also known as AICc is a classical form of AIC smoothing spline. AICc modifies the standard AIC with a correction for sample sizes. If we look at figure 5, AICc form of smoothing spline does better than that of AIC. It also forms extraneous and non-physical spikes. Due to the spikes, we were afraid that it won't be able to give us reasonable approximation values. This is the main reason in not choosing AICc to select the smoothing parameter λ for our smoothing spline [5].

Now, we will try using random smoothing parameters to try and search for a suitable parameter. This method will be carried out using visual inspection. This is not a very common and trustable method, but we are left with no other option. The graphs that were obtained using various parameters are given below.

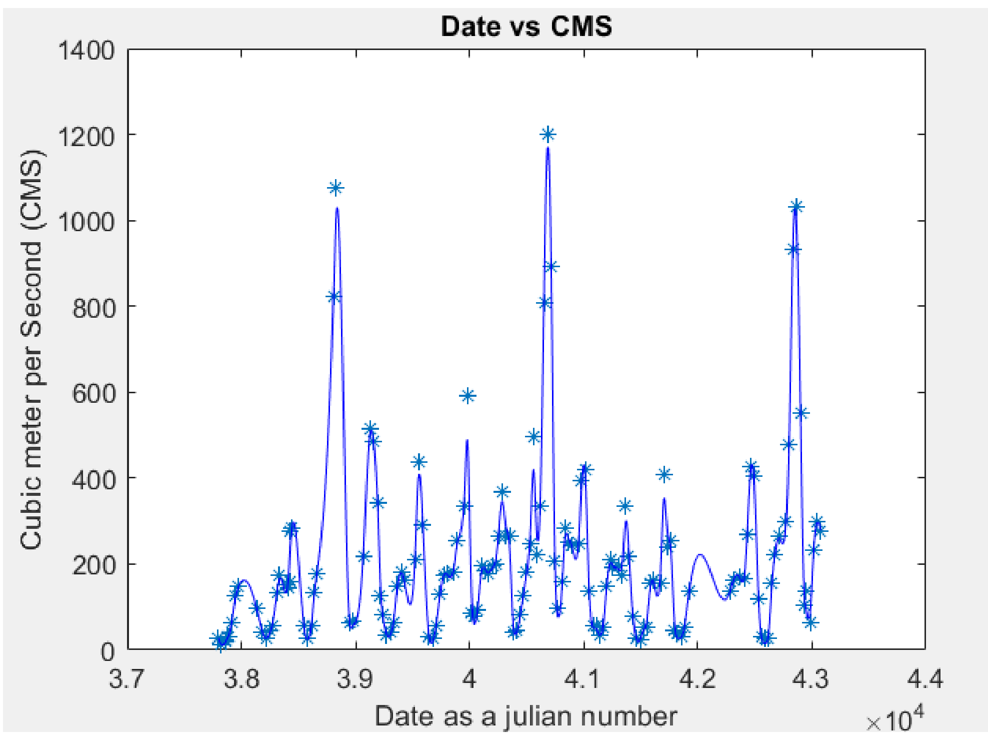


Figure 6. Curve Using Smoothing Parameter of 0.001

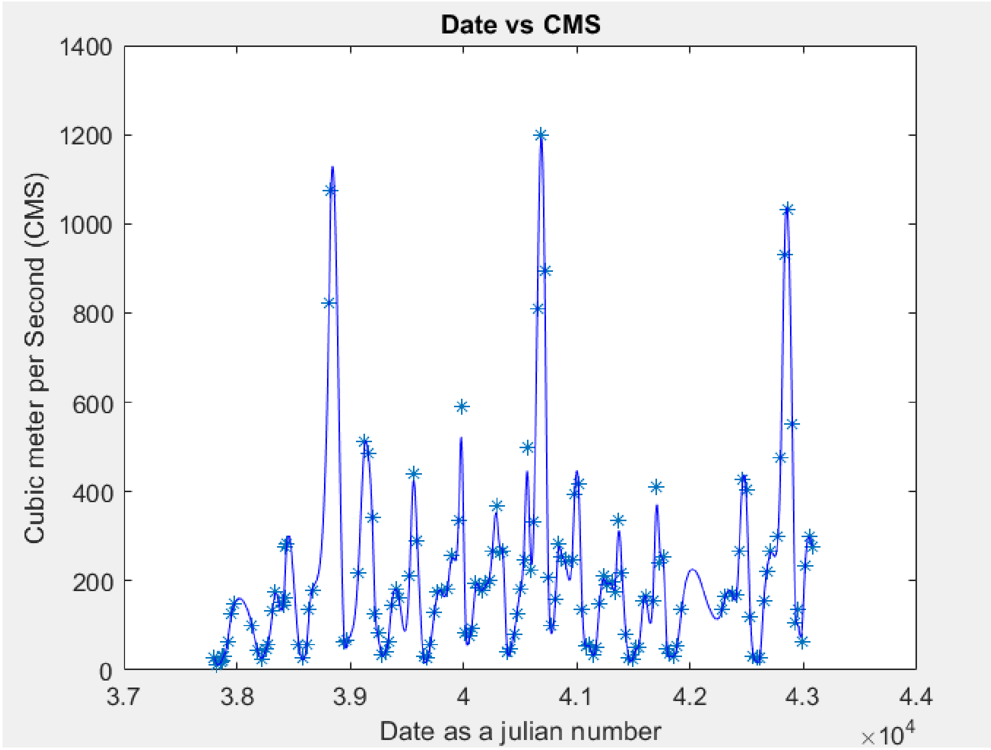


Figure 7. Curve Using Smoothing Parameter of 0.002

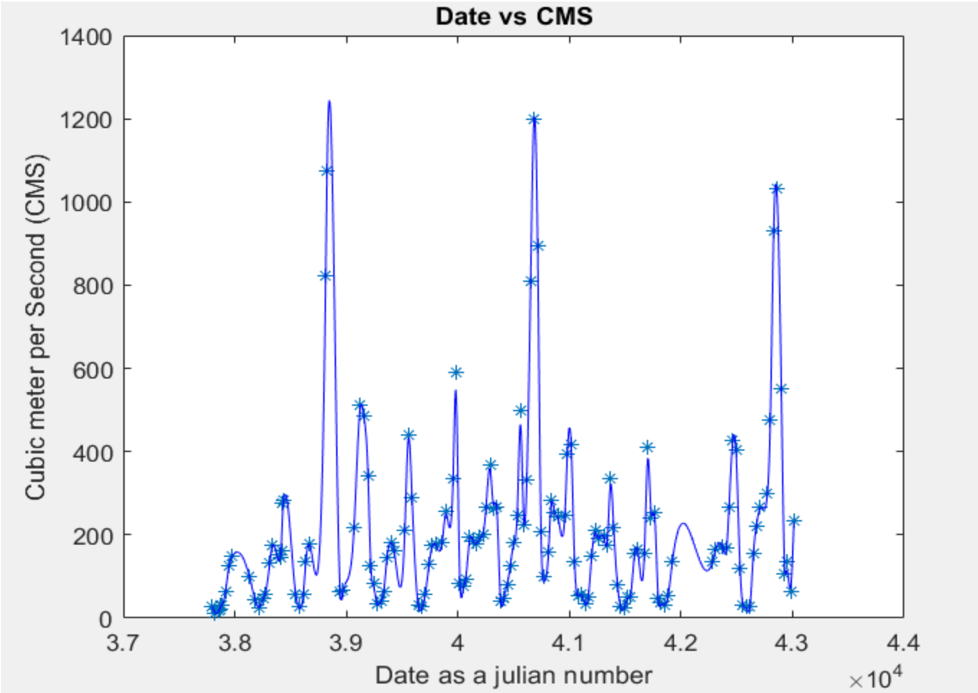


Figure 8. Curve Using Smoothing Parameter of 0.003

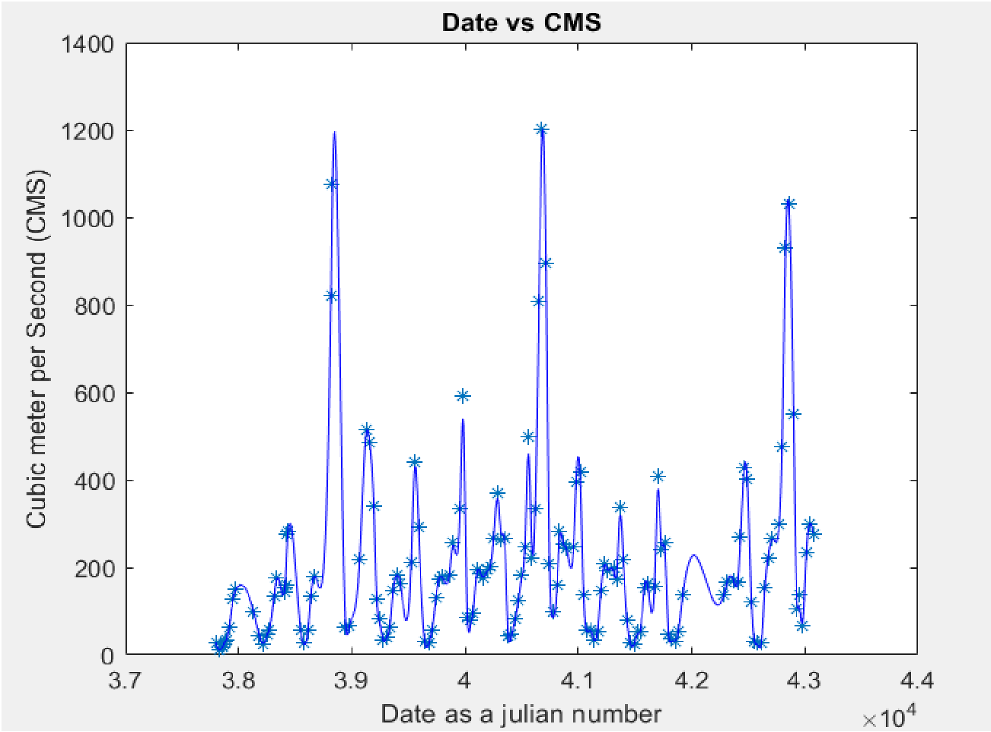


Figure 9.Curve Using Smoothing Parameter of 0.004

On the above figures, we tried using different values as a smoothing paramters to obtain reasonable daily bahaviour of our system without introducing unphysical behaviour. This methods solely depended upon our visual inspection. In figure 6, we used 0.001 as our smoothing parameter but it didn’t work out very well. We can see the curve missing lots of data points. In figure 7, we used 0.002 as our smoothing parameter. It was better than using 0.001 as a smoothing parameter. The curve was getting closer and was actually hitting more data points. Similarly we went for 0.003, 0.004 till we got to 0.009. The most promising curve we got was when we used 0.004 as our smoothing parameter. As we can see in figure 9, the curve is hitting almost every data points and it's not giving us any unnatural value as well. When we increased the value of the smoothing parameter, we were getting lots of unnatural values. The curve started to have lots of spikes as well. Figure 10 shows the behaviour of the curve when we chose 0.005 as our smoothing parameter. In the figure, we can see the curve was giving us some negative values for the flow rate of the river. The flow rate of the river can never take on negative values. There was also an increment in the value of the flowrate which almost hit 1500, which is not possible if we are considering Portneuf River. If we magnify our curve, it was not hitting almost all of the data points. The curve formed when using 0.004 as our smoothing parameter was the best curve, which was hitting almost all the data points and didn't contain any spikes and unnatural values. These are the main reasons we went with 0.004 as our smoothing paramter.

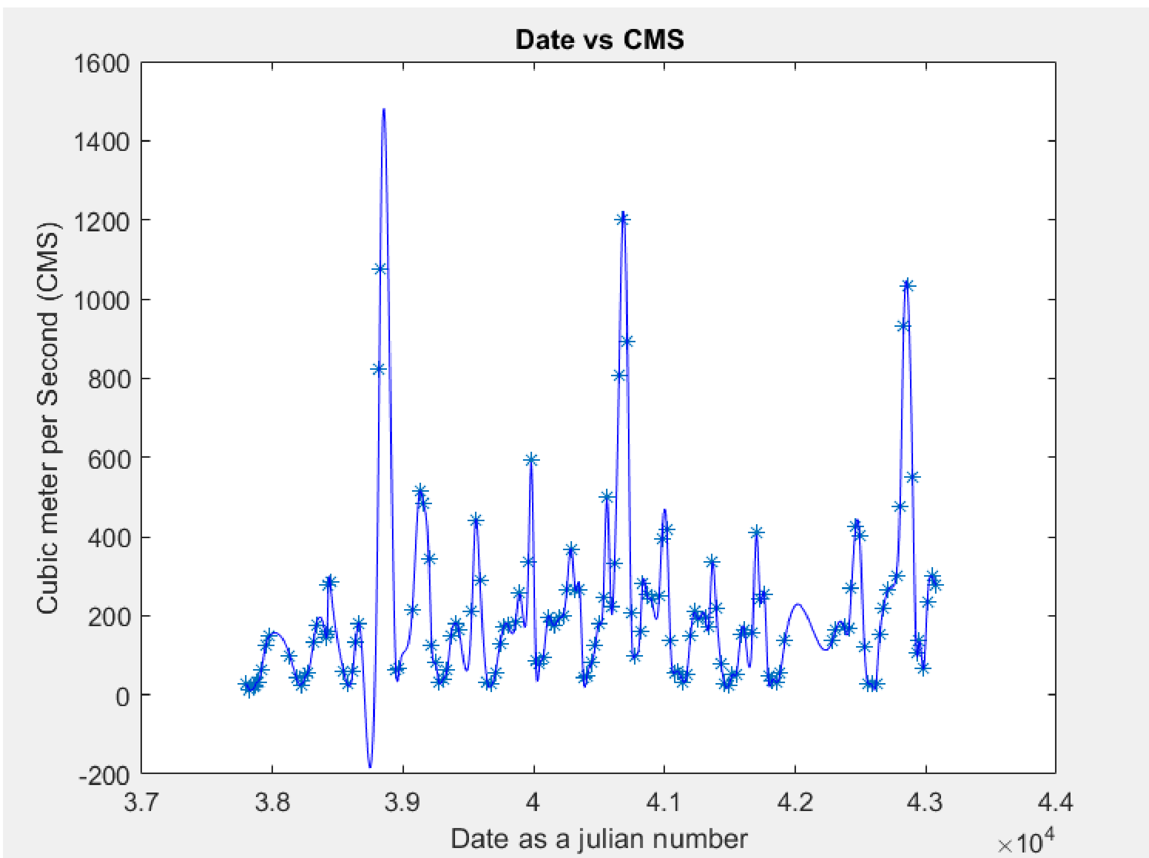


Figure 10. Curve Using Smoothing Parameter of 0.005

After we settled on a reasonable smoothing parameter value, we finally had callable functions for all of our I/O variables to produce uniform sample to use it in the System Identification routines we will use.

For generating daily, weekly and monthly data we created a MATLAB script which helped us generating data for each sampling period. In the script, we considered (Ts) as our sampling period; i.e., we get daily sampling when Ts=1, similarly when we consider Ts=7 and Ts=30 it would give us uniform weekly and monthly data. In the script, we first we load the data with smoothing parameter (lambda=0.004) for all smoothed variables. We also load ambient temperature data which contained the Julian dates of irregularly sampled data used in creating the smoothing spline approximation to input and output. Then we gave the script the sampling period value Ts. In the script we mentioned if the starting date of the Fichter dataset is greater than that of the ambient temperature data from NOAA, the script will change the time index of our data to account for this. This helped us while fetching Julian dates at regularly spaced Ts time intervals from ambient temperature data from NOAA. Also, this script helped us to evaluate the water temperature, flow rate, dissolved oxygen values from the smoothing spline using different values of Ts.

CHAPTER 5. MODEL ESTIMATION AND VALIDATION

System Identification is a methodology for building mathematical models of a dynamic system from the experimental data, i.e. using measurements of the system input/output signals to estimate the values of adjustable parameters in a given model structure [22]. The process of System Identification requires the selection of the model structure, and the choice and application of a method to estimate the value of adjustable parameters in the candidate model structure. In System Identification, researchers typically begin with the most tractable model structures, and if these fail to yield satisfactory results, proceed to more complex ones. Linear time-invariant (LTI) models, be they in discrete time or continuous time are always the first set of models one turns to. This is because the mathematical theory for LTI’s is completely understood, and the estimation of LTI model parameters can usually be framed as a least squares optimization problem (solvable by linear algebra), or via an iterative sequence of least square problems. The System Identification toolbox of MATLAB provides a complete suite of LTI discrete time models (provided the sampling is regular). After much preliminary testing (not reported here), we decided on focusing our modeling on two standard LTI models: the state space model and the transfer function model. Mathematically, these are more or less completely interchangeable. Moreover, both can estimate MIMO systems in the System Identification toolbox. This chapter will explain the state space model structure and the transfer function model structure and the results we obtained using these structures on the daily, weekly and monthly data from the Fichter station.

5.1 Models

As stated in the introduction, Kalman popularized the MIMO state space description of LTI's. In continuous time they take the form:

$$\dot{x}_{n\times 1} = A_{n\times n}x_{n\times 1} + B_{n\times k}u_{k\times 1}$$

$$y_{p\times 1} = C_{P\times n}\, x_{n\times 1}$$

$$x(0) = x_{0_{n\times 1}},\ t \geq 0$$

Where $x_{n\times 1}$(t) is the internal state of the system, $y_{p\times 1}$(t) is the observation, and $u_{k\times 1}$(t) is the input to the system. In discrete time, t ∈ (0,1,2,…) (i.e., in multiples of a constant sampling interval T_s), and the dynamics are:

$$X_{n\times 1} = A_{n\times n}x_t + B_{n\times k}u_t$$

$$y_t = C_{P\times n}\, x_t$$

$$x(0) = x_o$$

Where x_t is the $n \times 1$ internal state of the system at time t, y_t is the p×1 observation at time t, and u_t is the k×1 input, usually assumed to be either constant over [t,t+1], or piecewise linear and continuous.

In state space models, the parameters to be estimated are:

n : the necessary dimension of the internal state of the system,

$A_{n\times n}$, $B_{n\times k}$, and $C_{p\times n}$, the coefficient matrices for the dynamics.

Every LTI has an equivalent description in the frequency domain, either by Laplace transforms (continuous case) or by the z-transform (discrete case). This description is known as the Transfer Function (matrix). For each input u_j and each output y_i, the transfer function G_{ij} is such that:

$$Y_i = G_{ij} U_j$$

Where the uppercase variables represent the transforms of the respective lowercase time domain variables. It is well known that G_{ij} is a strictly proper rational function of the frequency variable (s or z^{-1}, depending on whether the system is in continuous or discrete time respectively), and all G_{ij} can be made to have the same common denominator, the characteristic polynomial for A. So, in MIMO transfer function models, one has to estimate p. k transfer functions of the form (for continuous time):

$$G_{ij}(s) = \frac{b_m \times s^m + b_{m-1} \times s^{m-1} + .. + b_1 \times s + b_o}{s^n + a_{n-1}\, S^{n-1} + .. + a_1\, S + a_o}$$

with m<n, as well as the internal state dimension n. In discrete time, an analogous expression results (just replace s by z^{-1}).

Fortunately, the System Identification toolbox has routines that estimate both continuous and discrete time MIMO state space and transfer function models. SSEST and TFEST will automatically run through a sequence of n values, and return the corresponding model with the best choice of n. The routine N4SID also handles MIMO state space models but uses a simpler algorithm than that used in SSEST. Throughout the remainder of this chapter, we will report on the performance of these three routines (SSEST, TFEST, N4SID) under various “data management” scenarios.

In this part of chapter five we will discuss the use of state space and transfer function models. In the appendix we specified what kind of System Identification method we want to use. We used SSEST for estimating a continuous-time space model system for data that was in the time-domain. Also, we used N4SID for estimating a discrete-time state-space model for our time-domain data. We used TFEST to estimate MIMO and SIMO transfer functions for our input-output in data. We also only show our results graphically in this chapter, but the actual models appear in the appendices.

5.1.1 Dividing Data into Two Portions

After getting a callable function (smoothing spline) representing our data, it was used to produce daily, weekly and monthly data with constant sampling interval of Ts = 1, 7, and 30. For doing System Identification, there should an estimation (testing) dataset and a validation dataset. For this section of chapter 5, the daily, weekly and monthly data was divided into two halves. The first half was used as a testing dataset and the second half was used as a validation dataset. To find out if the model was working, we compare the estimated model presented with the validation data. Ideally the fit on the validation data should match the performance of the estimation model on its dataset. The comparisons of the daily model using SSEST, weekly model using TFEST and monthly model using N4SID are shown in the figure below. The percentage given in all the figures is the "fit" statistic.

For a fit that "explains" the data, this "fit" statistic will be between 0% and 100%. Negative values of this statistic indicate that the fit has no explanatory values, and probably should be discarded.

The formula for this statistic is:

$$fit = 100\left(1 - \frac{norm(y-\hat{y})}{norm(y-mean(y))}\right), \quad \text{(Reported as a percentage)}$$

Here, y is the validation data, $\hat{y}$ is the estimated model's output, mean (.) is the average value, and norm (.) is the standard Euclidean norm. Hence, "fit" is similar to the "r-squared" statistic from multivariate statistics when it takes on negative values. So, when "fit" is positive, it gives the explanatory measure of the model, and when negative, shows how bad a model is.

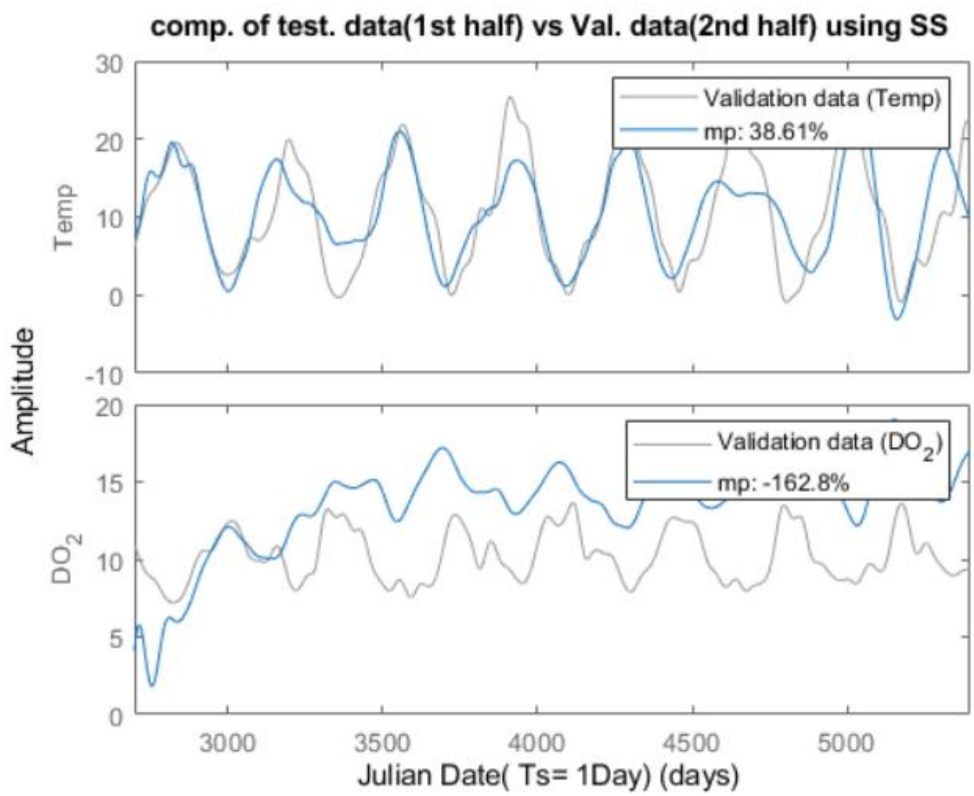


Figure 11. Comparison of Daily Testing Data with Validation data using SSEST

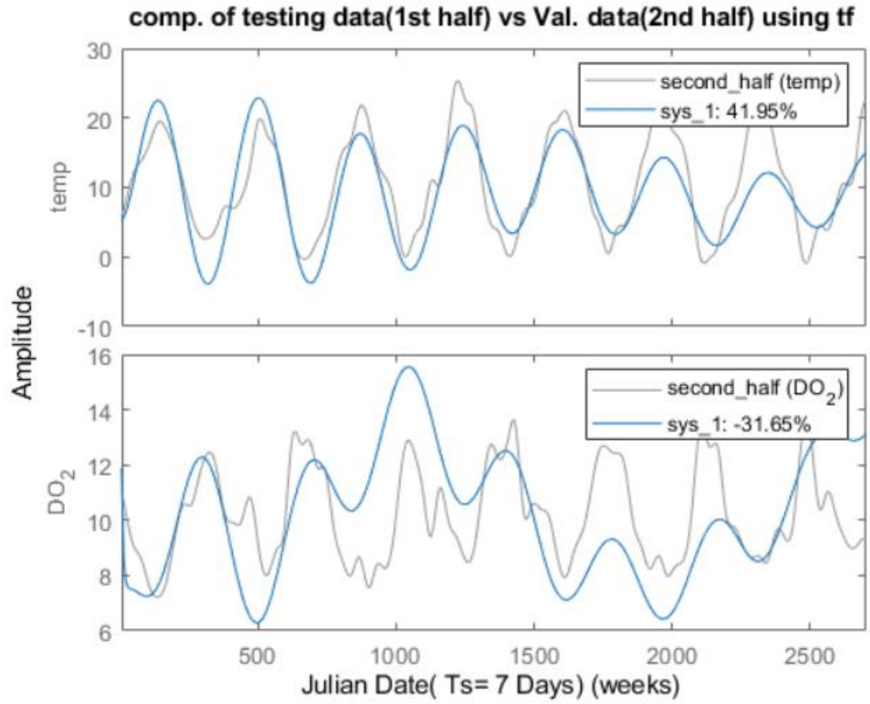


Figure 12. comparison of Weekly Testing Data with Validation data using TFEST

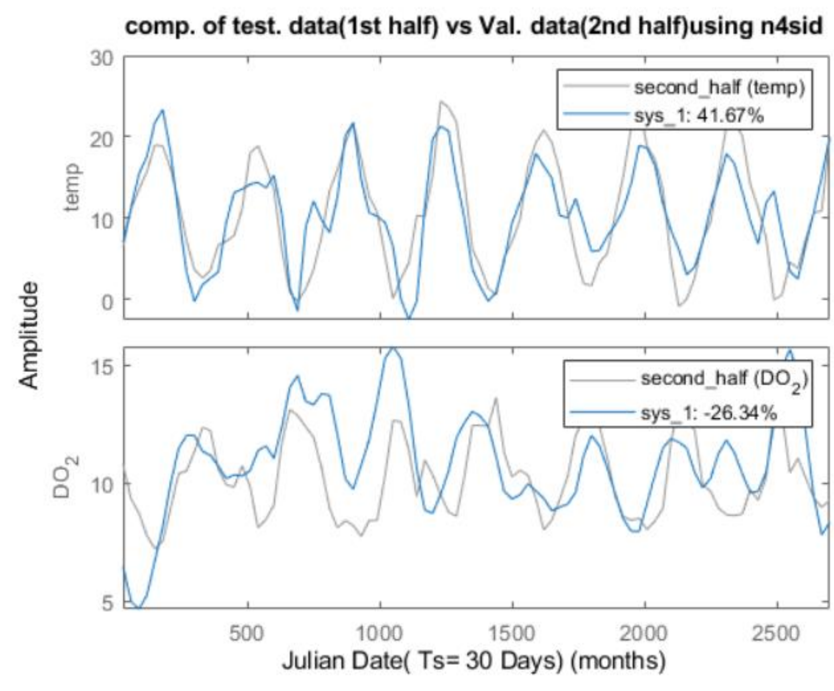


Figure 13. Comparison of Monthly Testing Data with Validation data using N4SID

First, we describe the performance of SSEST. For daily data we had 5,396 values for ambient temperature, dissolved oxygen, flow rate, and water temperature. We divided the data into two halves, so we have 2,698 values of testing dataset and the same number of values in the validation dataset. The daily model shown in figure 11 was computed using SSEST and was compared to the validation dataset. As we can see from the figure, the model didn't work out very well. There is a 38.61% match with the water temperature and a -162.8% "match" with the dissolved oxygen. Also, the model gave us a six-dimensional system with 91 free coefficients, which is not good. We also used SSEST on weekly and monthly data. For weekly, the fit to the estimation data went up and we got a 55.45% match with water temperature and a -6.326% "match" with the dissolved oxygen. For monthly data, we got a 59.02% match with water temperature and a -3.585% match with dissolved oxygen. Clearly the SSEST models are not providing a reasonable explanation of the system, especially with respect to the dissolved oxygen data.

Now we describe the performance of TFEST. The weekly model shown in figure 12 was computed using TFEST and compared to the validation dataset. As we can see from the figure, this model also didn't work out very well. But in comparison to SSEST, TFEST was working somewhat "better". For the weekly data, we had 41.95% match with the water temperature and a -31.65 "match" with dissolved oxygen. Similarly, while using TFEST on daily data, the fit percentage went up to 66.21% with water temperature and 5.378% with dissolved oxygen. But with the monthly data, the fit percentage went down just a little bit to 61.96% with the water temperature and 3.442% with the dissolved oxygen. TFEST is providing slightly better models for the water temperature, but unacceptable models for dissolved oxygen data.

Both results really disappointed us, but we were determined to perform more experiments with the data to try to obtain a better result. So, we finally tried using N4SID. The monthly model shown in the figure 13 was computed using N4SID and compared with the validation data. As we can see in figure 14, the fit percentage with the water temperature was 41.67% and the dissolved oxygen was -26.34%. for the daily data, the dissolved oxygen fit percentage went up to 15.34%. However, for the weekly data, we obtained totally unacceptable results. The fit was -476.2% for water temperature and -3053% for the dissolved oxygen. Figure 14, for the weekly data when N4SID was used, is given below. Clearly, the estimated system is unstable.

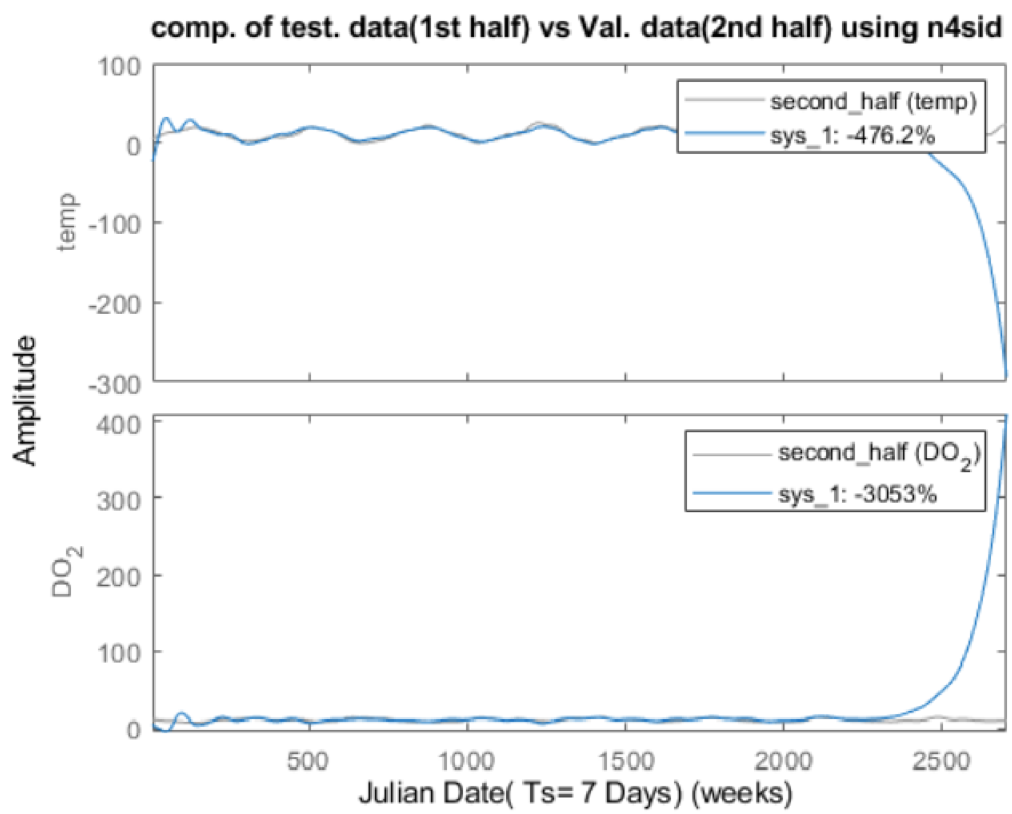


Figure 14. Comparison of Test. Data with Validation Data Using N4SID

5.1.2. Data Divided into three Portions

After the daily, weekly, and monthly data were compared using two different halves, we turned our eye to dividing the dataset into three portions. We took one third of the data as our estimation (testing) data whereas the other two thirds were used as validation data.

Since we got our "best" fit using N4SID when the data were divided into two halves, we tried using N4SID on our daily, weekly, and monthly data. When we use it for daily data, the comparison of first third with second and third portion didn't go well. We got a fit of 28.58% for water temperature and 12.82% for dissolved oxygen. Now we tried the same procedure with the weekly data. The fit of the estimated model on the validation data got even worse with a fit of 43.05% for water temperature and -22.37% for dissolved oxygen. The fit became even more worse obtained using the monthly data. The example of weekly data comparison of the model estimated on the first third and validated on the second third using N4SID is given below in figure 15. Clearly, the estimated model is unstable.

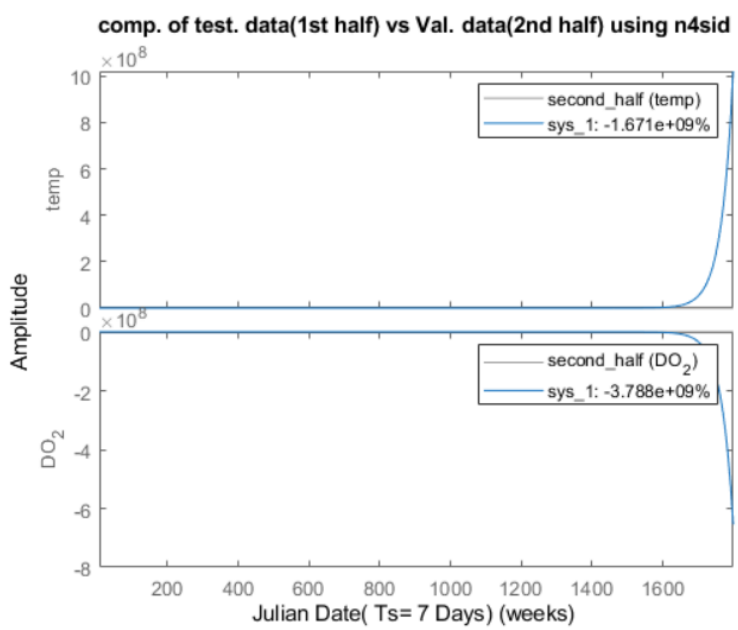


Figure 15. Comparison of Test. Data with Validation Data using N4SID

We then tried TFEST on the dataset divided into thirds. While using TFEST for the daily data, the fit went up to about 61.03% for water temperature and 6.75% for dissolved oxygen. When we tried using the weekly data, the fit was unacceptable. The fit obtained was -131.4% for water temperature and 6% for dissolved oxygen.

Similarly, we tried using SSEST with all the data divided into three thirds. But the estimates were again unacceptable. The fit obtained while using SSEST to the monthly data is given in figure 16 below.

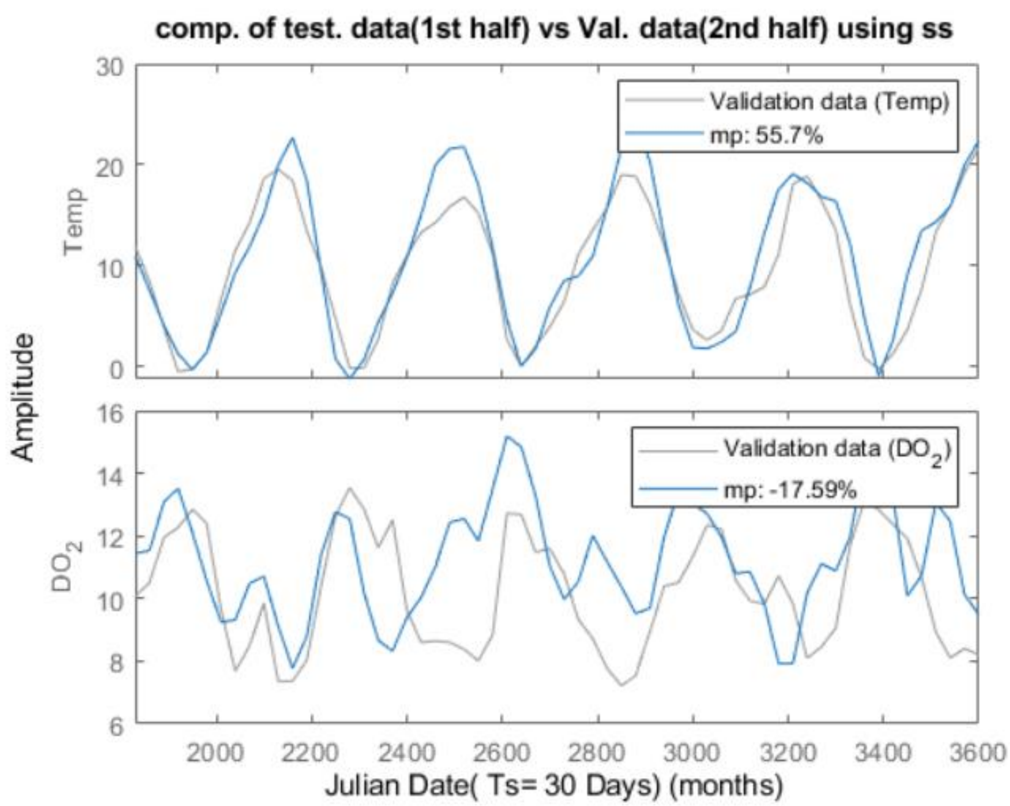


Figure 16. Comparison of Monthly Testing Data with Validation Data Using SSEST

5.1.3. Data Divided into Four Portions

After the comparison of the data when divided into three portions, our next experiment on the data was carried out by dividing the data into four quarters. As at least one of the results obtained while dividing the data into three thirds was better than the result obtained while dividing the data into two halves, we thought that estimating and validating on smaller datasets might be promising. When the daily data was divided into four quarters, each dataset contained 1,349 samples. Similarly, the weekly dataset contained 183 samples. If we were to divide the monthly data into four halves, each dataset would have consisted of 40 points. While doing the System Identification, the number of free parameters in the models would be greater than that of the sample points in monthly data, and any model would be over parameterized. This was the main reason why the System Identification of monthly dataset divided into four quarters was not attempted.

When the dataset was divided into four quarters, we first investigated the performance of N4SID. While using N4SID on the daily data, the best fit obtained was 52.91% for water temperature and 16.45% for dissolved oxygen. The 16.45% fit for dissolved oxygen was the best fit till now for this output variable. For the weekly data, the best fit obtained was 70.45% for water temperature and 19.49% for dissolved oxygen. Estimated using weekly data was performing somewhat better than using daily data. The results for the weekly data with the best fit is given in Figure 17.

Next, we investigated the performance of SSEST. While using SSEST on the daily data, the best fit obtained was 46.01% for water temperature and -21.74% for dissolved oxygen. On the weekly datasets, none of the estimates from N4SID were acceptable. For example, on the weekly datasets, the fit obtained was -17.86% for water temperature and -215.1% for dissolved oxygen.

The figure for the comparison carried out using SSEST to weekly data is given below in figure 18.

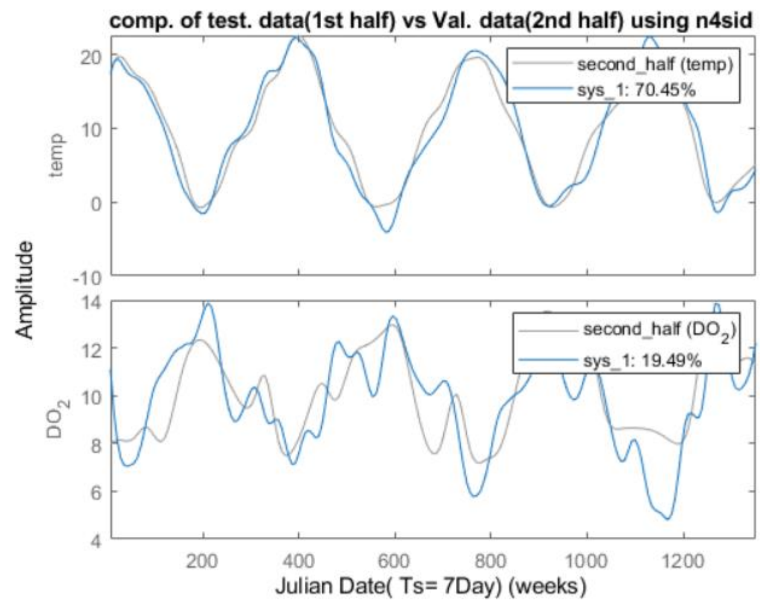


Figure 17. Comparison of Weekly Testing Data with Validation Data Using N4SID

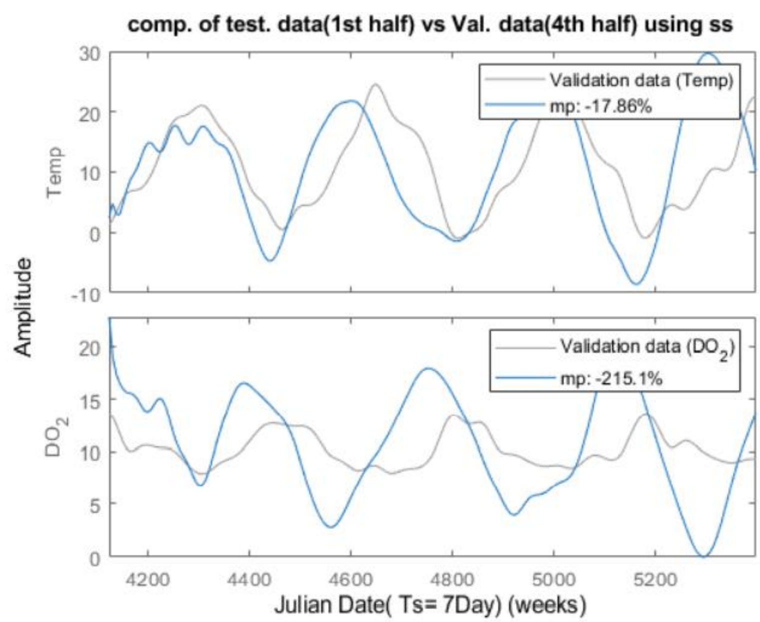


Figure 18. Comparison of Weekly Testing Data with Validation Data Using SSEST

Lastly, we investigated estimation using TFEST. On the daily datasets, the fit obtained was 73.18% for water temperature and 21.42% for dissolved oxygen. This was a ray of hope after the disastrous experiment conducted using SSEST. The fit percentage was going up. Again, when TFEST was used with the weekly dataset, the best fit was 76.76% for water temperature and 8.526% for dissolved oxygen. Figure 19 shows the best fit obtained while using TFEST on the daily data, and figure 20 shows the fit to weekly data. Clearly, we are doing well for the water temperature modeling, but not so well for dissolved oxygen.

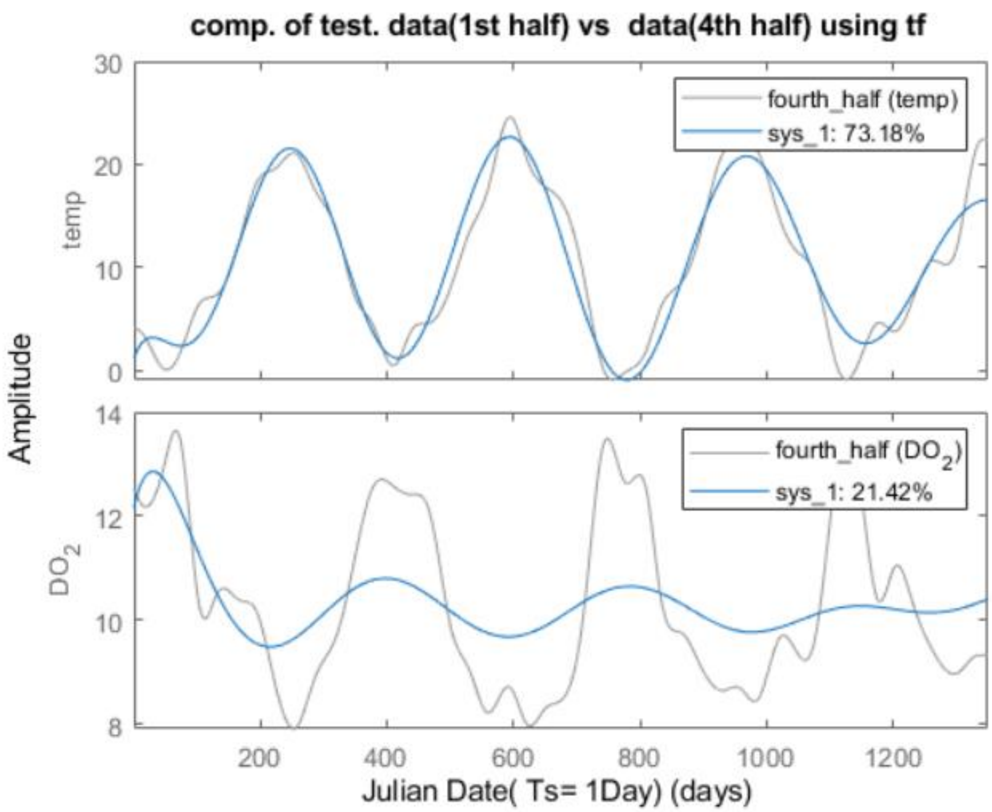


Figure 19. Comparison of Weekly Testing Data with Validation Data Using TFEST

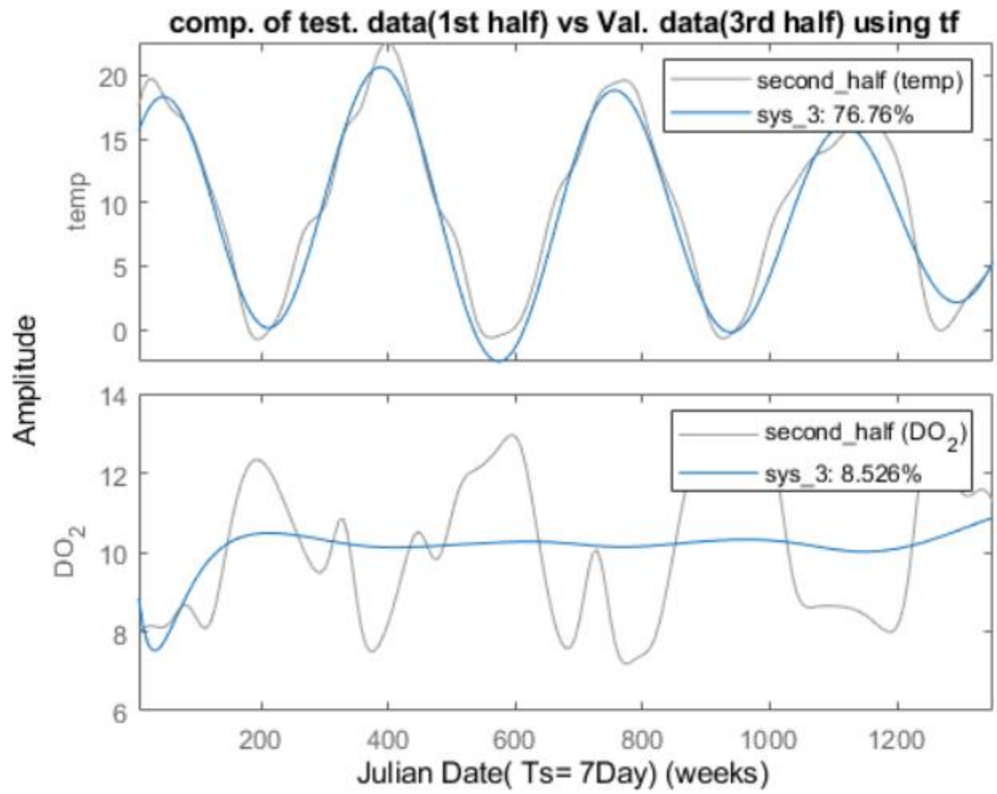


Figure 20. Comparison of Weekly Testing Data with Validation Data Using TFEST

5.1.4 Data Divided into Six Portions

From the previous section, we saw improvements (at least using TFEST) when using smaller estimation datasets. Hence, we now try dividing the data into six equal portions. Each sixth was used as testing data and validation data. In this section we report only on the “best” estimates for each routine (SSEST, N4SID, TFEST), searching through each sixth of the dataset used as an estimation set, and validated against the remaining five sixths.

When using SSEST for both the daily and weekly data, the fits were worse than the results obtained when dividing the data into four portions. The best fit obtained using SSEST to daily was 56.93% for water temperature and 15.99% for dissolved oxygen. But when the SSEST was used with the weekly data, the “best fit” obtained was -20.65% for water temperature and -1289% with the dissolved oxygen. Hence, SSEST was returning useless estimates.

Similarly, N4SID was applied to the daily and weekly datasets. The best fit obtained was better than that of using SSEST but wasn’t even near to when the data was divided into four portions. The best fit using N4SID to daily data was 54.12% for water temperature and 14.56% for dissolved oxygen. When used with the weekly datasets, the estimates were useless. Both fits obtained were on the range of -10^6. Figure 21 will show the worst percentage obtained when using N4SID on the weekly data. Clearly, the figure shows that the estimated model is unstable.

Lastly, we tried using TFEST on the daily and weekly data. The fit when used with the weekly data was not satisfactory, with the best fit obtained being 77.46% for water temperature and -37.07% with dissolved oxygen. But when TFEST was used with the daily dataset, the best fit obtained was 66.59% for water temperature and 56.69% for dissolved oxygen. This was the best percentage fit that was obtained till now. The figure 22 shows the best fit obtained till now.

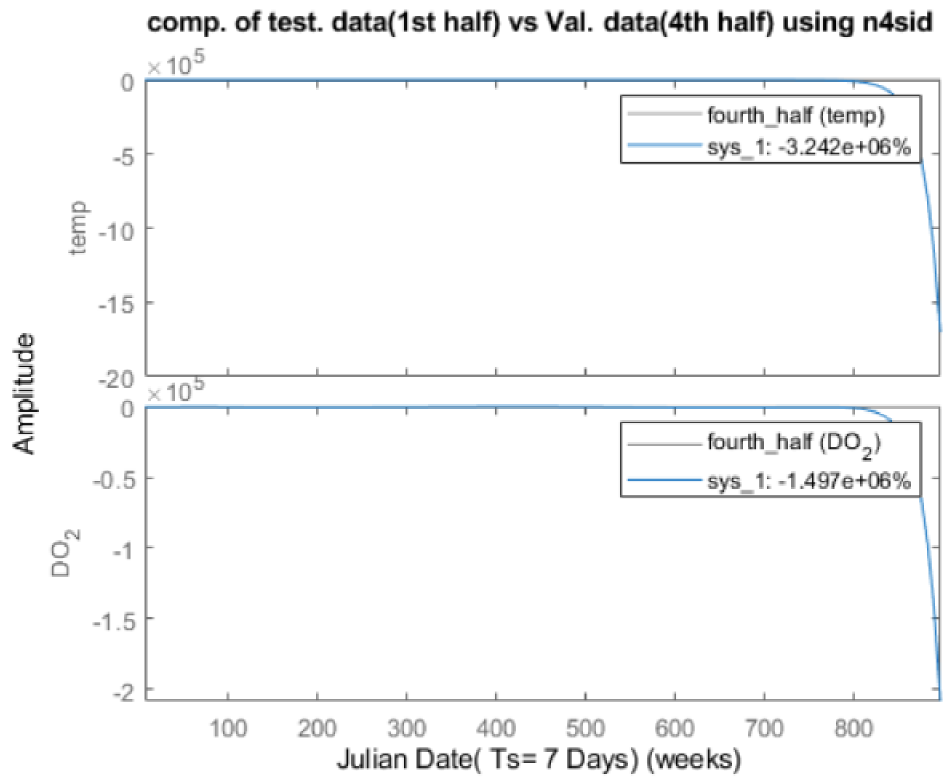


Figure 21. Comparison of Weekly Testing Data with Validation Data Using N4SID

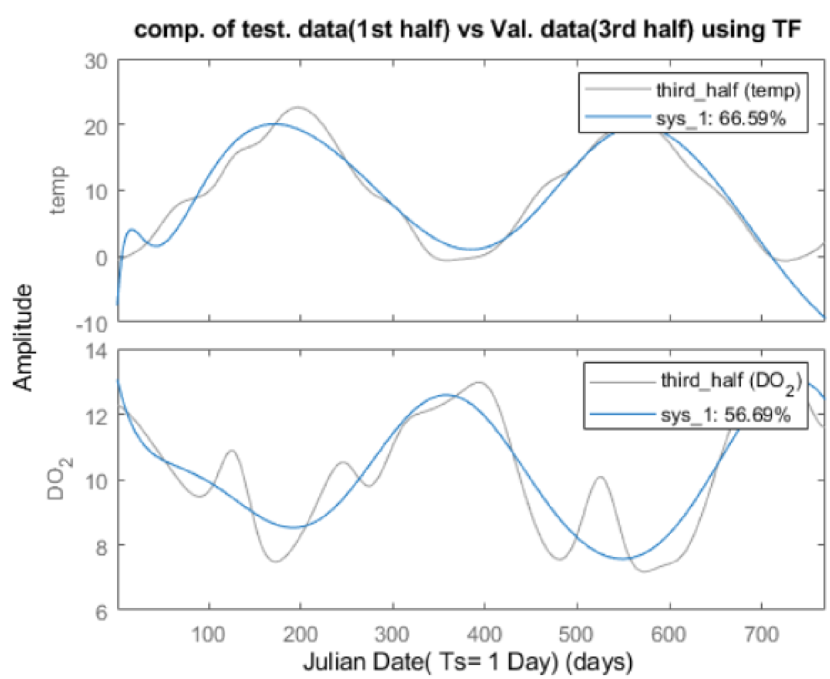


Figure 22. Comparison of Daily Testing Data with Validation Data Using TFEST

5.2 Model Excluding the First Three Years

The data provided from Idaho Department of Environmental Quality ranged from 1997 to 2017. This only applies to certain stations. For the Fichter station, the data sampling was from 2000 to 2017. Since the models that were obtained from the previous experiments were not satisfactory, the only option remained was to go back and analyze the provided data again. After analyzing the data that was provided, the first three years of approximately monthly data were very irregular and there were many missing values. So, further experimentation was carried after excluding the first three years of data. These experiments followed the same process as our previous experiments. First, we checked to be sure that our smoothing spline representations of the data on this smaller dataset still gave us reasonable callable functions for generating data on regular time grids (they did). Next, we used the spline representations to generate regularly spaced daily, weekly and monthly data. Daily, weekly, and monthly data was then divided into two, three, four and six portions. The divided data was analyzed and compared using N4SID, SSEST, and TFEST. Monthly data was only analyzed when the data was divided into two and three portions, for the reason stated in the previous section.

When the data was divided into two halves and analyzed using N4SID, the best fit obtained was 75.04% for water temperature and 46.56% for dissolved oxygen, and this was on the daily dataset. This was the best result obtained since the project begun. Next, the data was again analyzed using SSEST. The best fit obtained was a 78.99% fit for water temperature and 52.65% for the dissolved oxygen and this was on the daily dataset. Lastly when analyzed using TFEST, the best fit obtained was 73.38% for water temperature and 53.53% for dissolved oxygen on the daily dataset.

Next, we tried the analysis after dividing the dataset into three thirds. Upon using SSEST, N4SID, and TFEST on this data, the results obtained were inferior. The best result using SSEST obtained was a fit of 58.35% for water temperature and a fit of 6.68% for dissolved oxygen on the daily dataset. The other comparisons for N4SID and TFEST didn't go well, since the results obtained had negative values for the fits for dissolved oxygen.

Again, the data were divided into four quarters and analyzed using all three functions. When the data was analyzed using N4SID, the best fit obtained was 44% for the water temperature and 4% for dissolved oxygen, and this was for weekly data. This was worse than dividing the data into three thirds. The experiment went worse when analyzed and compared using SSEST. The best fit percentage obtained had negative fits for both the water temperature and dissolved oxygen, regardless of sampling period. Lastly, the data was again analyzed using TFEST. When TFEST was used on the weekly data, the result had positive fit values but was not quite satisfactory. However, when it was tested for the daily data, the best fit obtained was 76.84% for water temperature and 55.89% for dissolved oxygen.

Lastly, the data was divided into six portions and was tested using all the functions. When using SSEST and N4SID, the validation results started to get worse. The best fit obtained was 83.64% for water temperature and 8.046% for dissolved oxygen, on the daily dataset. Even though the data was being fit quite well for the water temperature, the fit for dissolved oxygen was far away from an acceptable value. When TFEST was used for the daily data, the best fit obtained was 81.17% for water temperature and 63.83% for dissolved oxygen. Also, when the weekly data was tested using TFEST, the best fit obtained was 84.42% for water temperature and 66.38% for dissolved oxygen. Figure 23 below shows the best fit that was ever obtained during this experiment and figure 24 shows the corresponding second best fit (TFEST on the daily data).

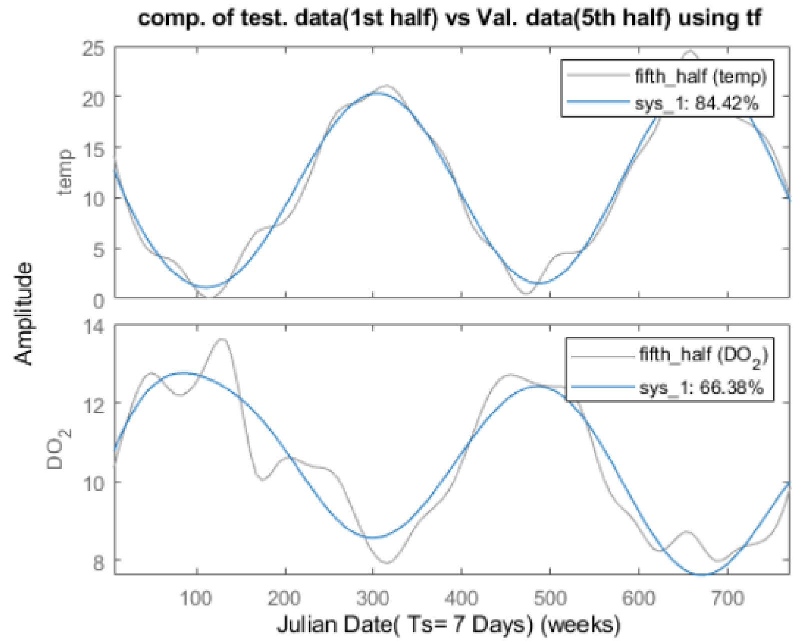


Figure 23. Comparison of weekly data excluding three years using TFEST

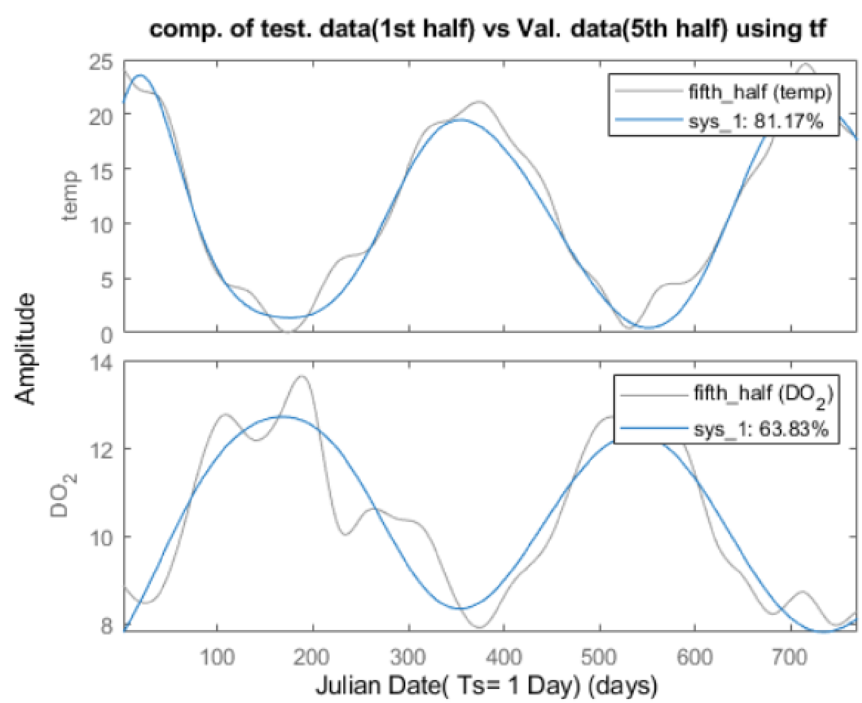


Figure 24. Comparison of Daily data excluding three years using TFEST

5.3 Excluding Flow Rate of the River

Previous experiments were carried out treating the system as a MIMO system, with ambient air temperature and flow rate of the river as inputs, and water temperature and dissolved oxygen as outputs. Water temperature fits were reasonable, but dissolved oxygen fits were not satisfactory. MIMO System Identification is more difficult theoretically and numerically than single input – multiple output (SIMO) identification, as the optimization routines behind N4SID, SSEST, and TFEST need to handle more parameters in the model, and the optimization criterion of final prediction error entails a “balancing act” between the inputs. Hence, we decided to see how a SIMO model might perform. Clearly ambient air temperature has a major influence on water temperature, and hence also on dissolved oxygen. The flow rate probably has a lesser effect on both outputs. So in this section, we tested whether just using ambient temperature as the single input to the system might improve the “overall” estimate: i.e., probably slightly decrease the goodness of fit for water temperature, but possibly increase the goodness of fit for dissolved oxygen.

As in previous experiments, we divided the regularly spaced data into daily, weekly, and monthly datasets. The daily, weekly, and monthly datasets were divided into two, three, four and six portions. All the datasets were then compared using the same estimation routines that were used in the previous experiments. At first two halves’ datasets were compared and analyzed. The best fit obtained was 67% for water temperature and 32.21% with the dissolved oxygen using SSEST on the daily dataset. The worst fit obtained was 26.56% for water temperature and -384% for dissolved oxygen using N4SID on the four quarters dataset. The results obtained from this experiment were sometimes equally good and sometimes quite poor.

Again, the three, four and six portions of the data were analyzed. During the analysis the comparison didn't improve, rather the best fit percentages were decreasing by quite a good margin. Also, the fit of the water temperature was also affected due to the use of single input and multiple output system as expected. In some experiments, dissolved oxygen fit percentage was hiking up but the percentage fit for the water temperature was equally dropping. The best fit obtained for three halves data was 57.63% for water temperature and 51.33% for dissolved oxygen using SSEST on the weekly dataset.

When we analyzed the data for four portions and six portions, the obtained result was not even close to what was obtained from the previous experiments. The best fit obtained for the four portions dataset was 68.81% for water temperature and -4.787% for dissolved oxygen. For the data with six portions the best fit was 23.2% with the water temperature and 11.16% with the dissolved oxygen. The figures shown below illustrate the best and worst results obtained while doing the experiment with a single input and multiple output system. Note that our "best" fit with a SIMO model (figure 25) does have the expected decrease in fit to water temperature, and does give a reasonable fit to dissolved oxygen, but suffers in that near date index 750, the predicted water temperatures are negative, an impossibility!

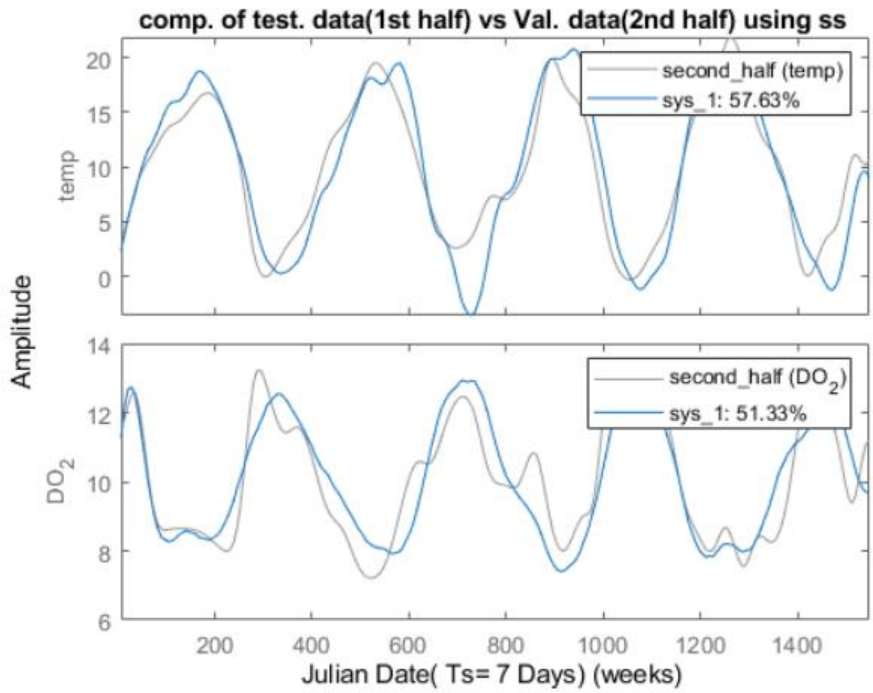


Figure 25. Comparison Using SSEST for SIMO System with Four Quarters

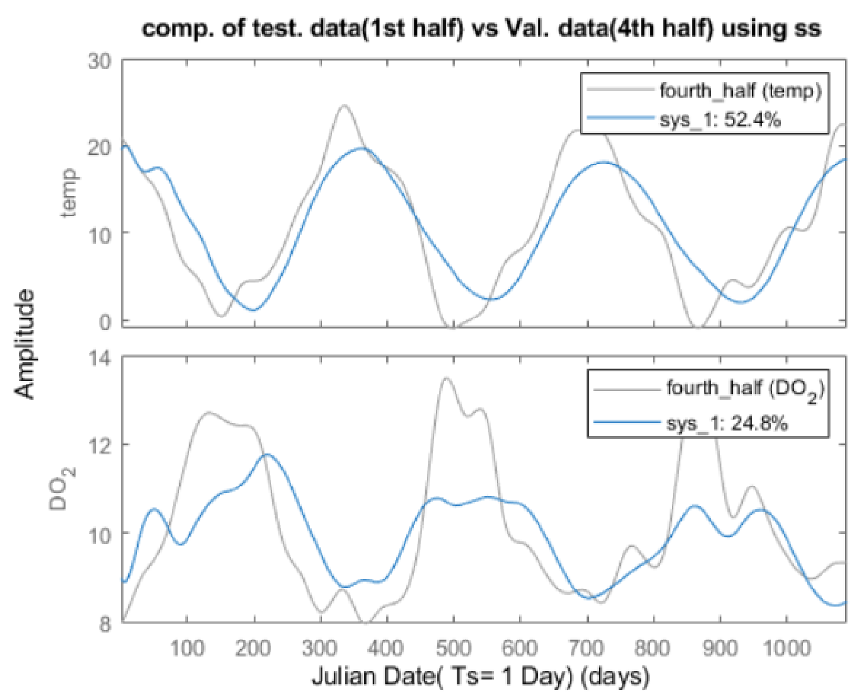


Figure 26. Comparison Using N4SID for SIMO System with Six Portions

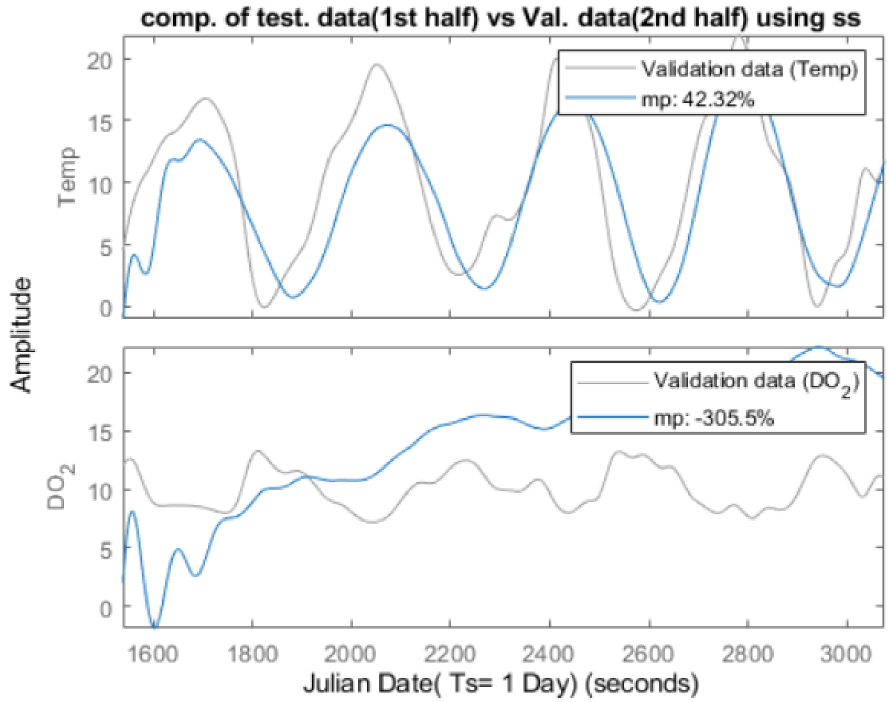


Figure 27. Comparison Using SSEST for SIMO System with two Halves

5.4 Model using the Average of the Data

This was a research project and new ideas kept coming up. In this section of chapter 5, the temporal average value of the output data was taken and subtracted from the data. That is the average value was taken as a new origin of the output data. Our rationale for doing this is as follows. From our previous experiments we suspect that the MIMO system we have been studying may be a nonlinear system. Since we are attempting to describe the dynamics using LTI models, such models will only be appropriate in the neighborhood of equilibrium points of the nonlinear system. We don't know the true equilibrium point(s) of the system, but since we observe annual oscillations of the output variables, we decided to center these oscillations about their means. This may give the linear model estimation routines a better chance of succeeding. It should be noted that since we are estimating linear models (in the output variable), the estimates can be translated back to their original coordinate system simply by adding back the means.

After taking the average and subtracting it from the all the sample points for the temperature of the river and dissolved oxygen, the data was again divided into daily, weekly and monthly data. The daily, weekly, and monthly data were again divided into two, three, four and six portions and compared using all the function like in previous experiments.

At first the data were analyzed for the normal dataset, which means without excluding flowrate and the first three years of the data. While using N4SID with the data that was divided into two halves, the best fit obtained was with the daily data with a fit of 42% for water temperature and 33.49% for dissolved oxygen. After that, the weekly and monthly data were tested which resulted in unstable models with the fit of $-7.6*10^5$% for water temperature and $-5.74*10^6$% for dissolved oxygen. Similarly, daily, weekly and monthly data were compared using SSEST and TFEST. The best result obtained when the data was divided into two halves was for weekly data when TFEST was used. The fit obtained was 67.8% for water temperature and 49.8% for dissolved oxygen. Figure 28 below shows the best result obtained when the data was divided into two halves, and TFEST was used.

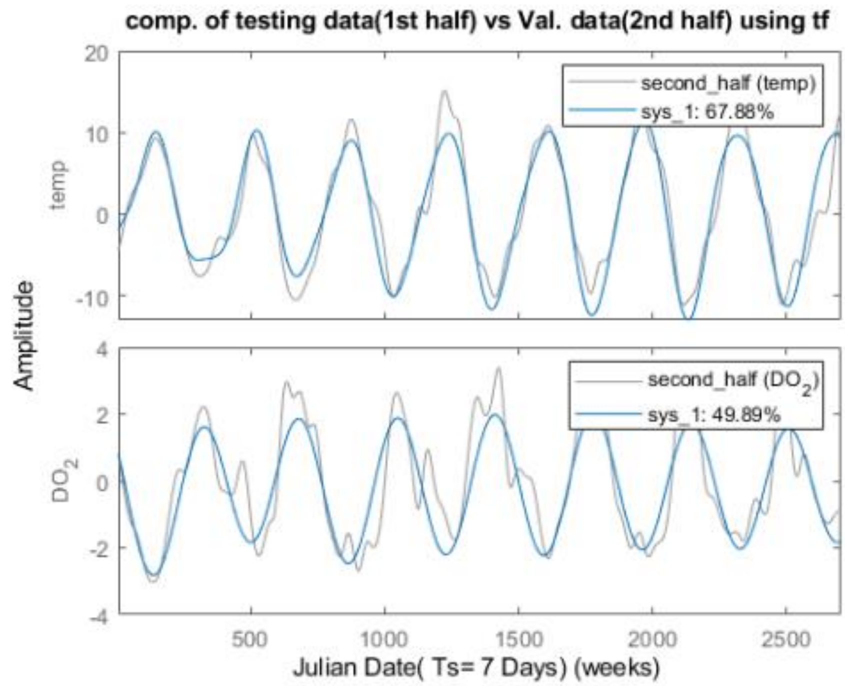


Figure 28. Comparison of Two Halves data Using TFEST

Likewise, similar experiments were carried out for the data that was divided into three, four, and six portions. For three thirds data, set the comparison result was not as expected. The best fit obtained was better than that obtained with the two halves datasets, and this was using daily dataset using TFEST, but the overall results were not quite satisfactory. There were lots of negative percentage fits obtained while doing the validations. The best fit obtained was 71.84% for water temperature and 53.66% for dissolved oxygen using TFEST. Similarly, the experiment was conducted for the four quarters data, the best fit obtained was while using TFEST on the weekly data with a fit of 73% for water temperature and 58% for dissolved oxygen. Again, when the sixths datasets were analyzed, the best fit was obtained when using TFEST with the daily data. The best fit obtained was 77.45% for water temperature and 54.79% for dissolved oxygen. The figure of some of best fits obtained are given below.

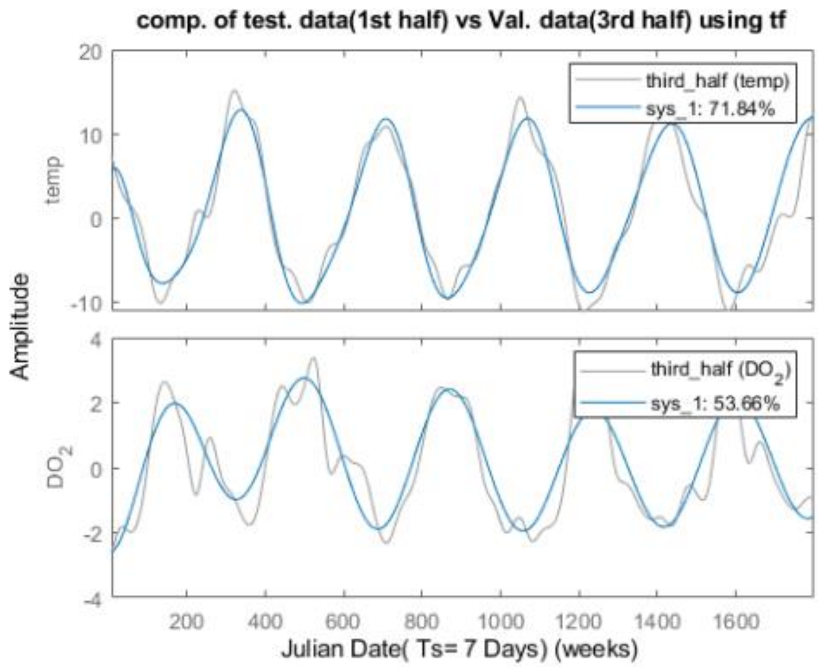


Figure 29. Comparison For Three Thirds Data Using TFEST.

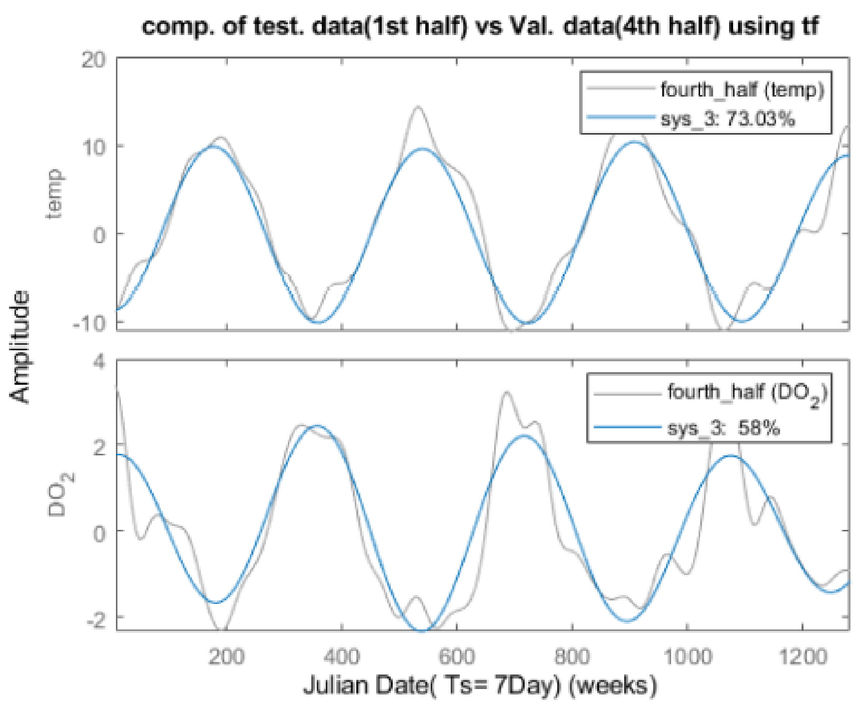


Figure 30. Comparison of Four Quarters Data Using TFEST

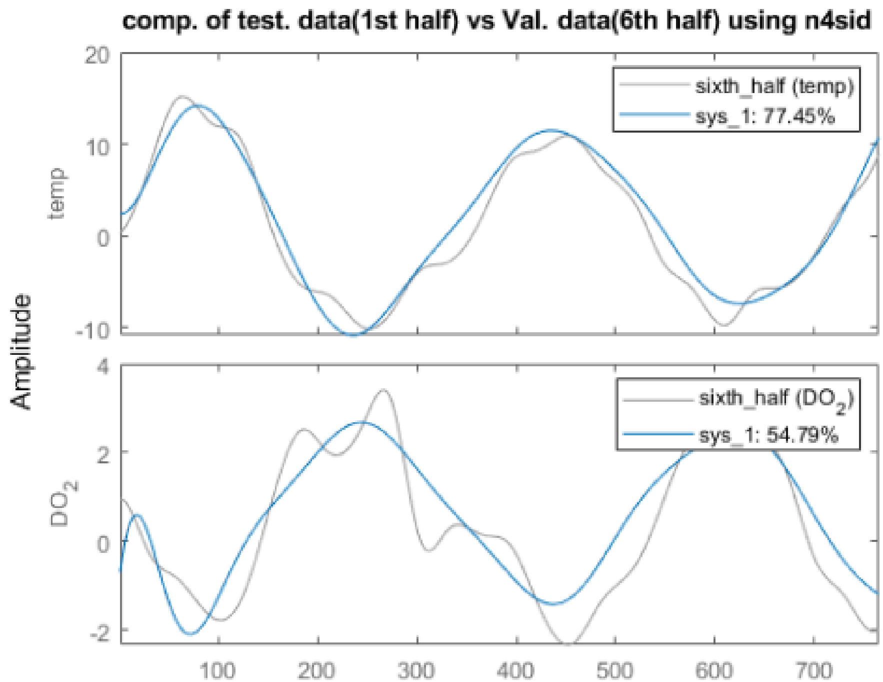


Figure 31. Comparison of Six Sixths Data Using N4SID

The experiment was again continued with the data excluding the first three years data due to lots of missing sample values for those years. The data was again divided into daily, weekly, and monthly datasets and each of these were divided into two, three, four and six portions. All the data were compared using N4SID, SSEST, and TFEST again. When the two halves' data were tested using N4SID, the best fit obtained was again for the weekly data. When N4SID was used for the weekly data, the fit obtained was 73.6% for water temperature and 44% for dissolved oxygen. Similarly, when SSEST was used for the weekly data, 63.7% was obtained for water temperature and 38% for dissolved oxygen and for TFEST, 67.6% for water temperature and 45.9% for dissolved oxygen, using the weekly data.

For the dataset divided into thirds, the best fit obtained was 83.6% for water temperature and 47.8% for dissolved oxygen, when TFEST was used on weekly data. When the experiment was conducted with the four quarters data, the best fit obtained was, 74.6% for water temperature and

57.6% for dissolved oxygen, when TFEST was used on the weekly data. The best fit obtained with the six sixths dataset was a fit if 86.63% for water temperature and 52.75% for dissolved oxygen, when TFEST was used on the weekly data. The figures of some of the best results obtained while conducting this experiment are given below.

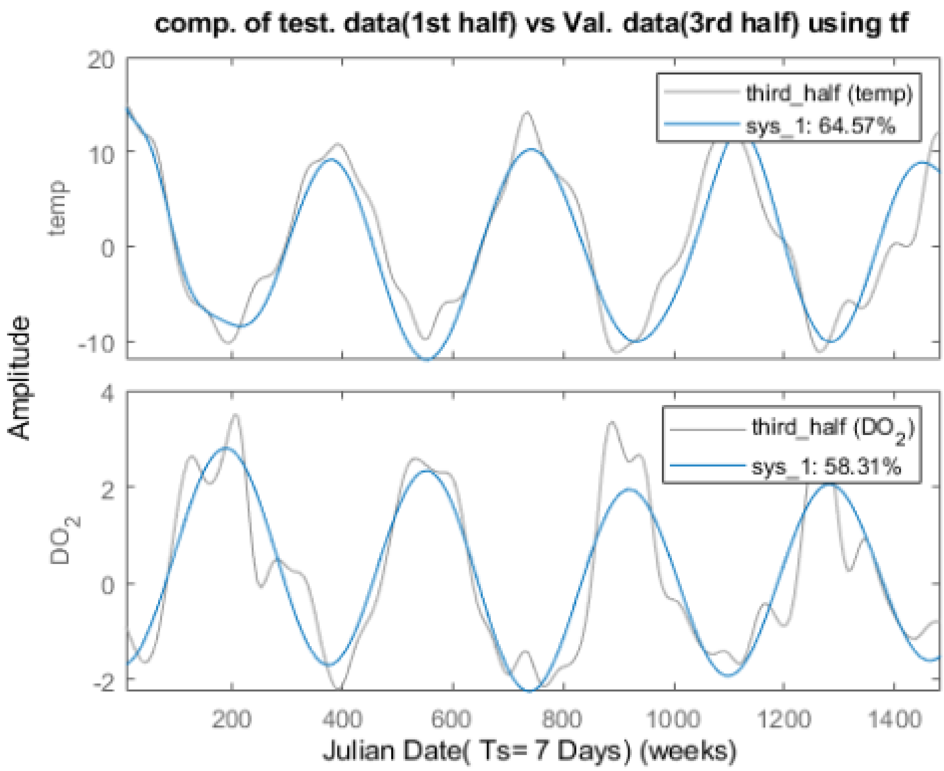


Figure 32. Comparison on Two Halves Data Using TFEST

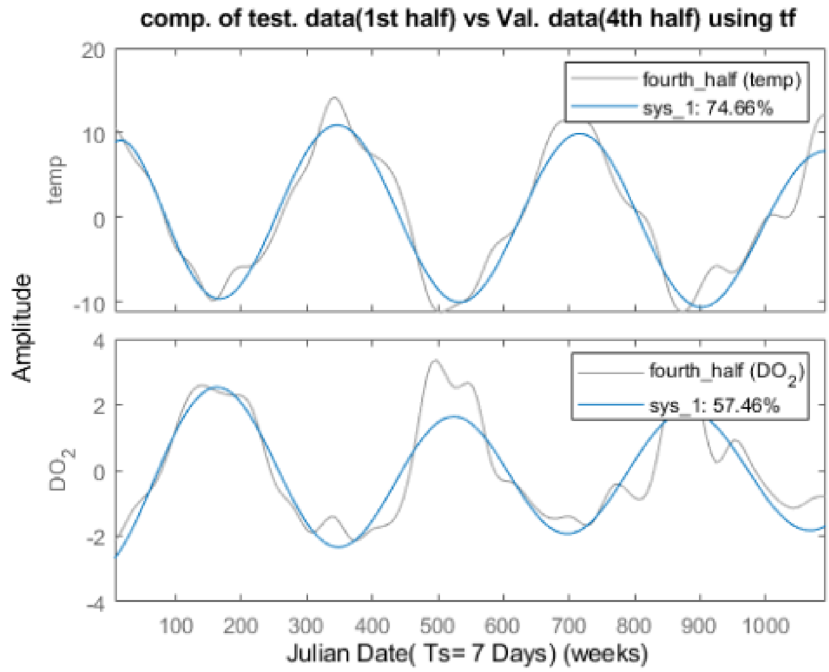


Figure 33. Comparison on Four Quarters Data using TFEST

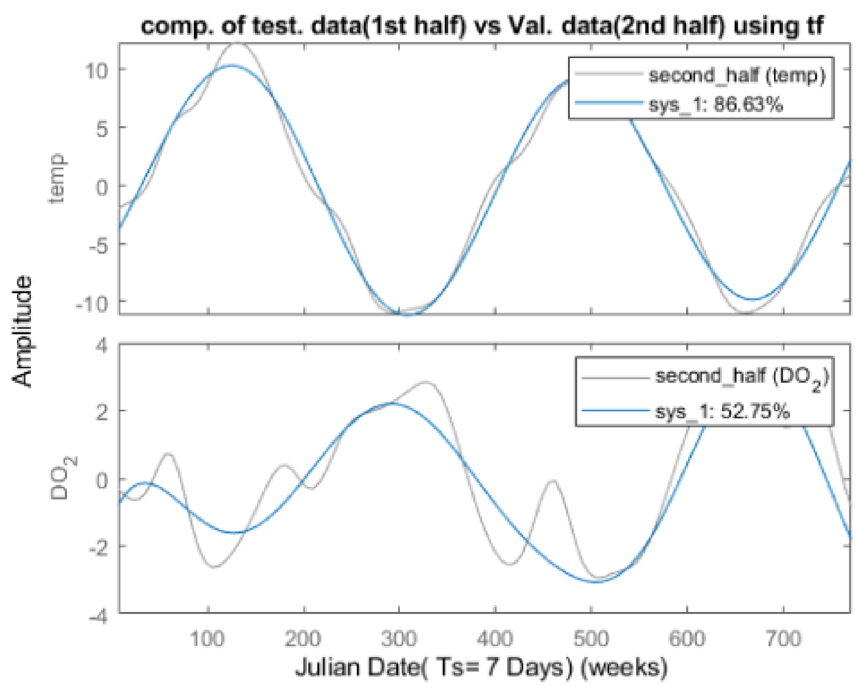


Figure 34. Comparison on Six Sixths Data Using TFEST

Similar experiments were again carried out using by excluding the flow rate of the Portneuf River. The results obtained were not satisfactory.

All of the experiments discussed in this chapter were conducted using MATLAB and the System Identification Toolbox. The best results obtained in these experiments were obtained using the transfer function estimator TFEST. The two state space estimation routines (N4SID, SSEST) underperformed TFEST, often had high dimensional parameterizations, and often gave negative fit values and unstable systems as estimates.

5.5 Use of CONSTID

CONTSID was the first toolbox entirely dedicated to continuous-time model identification from sampled data to be run with MATLAB. It was first released in 1999 (Garnier and Mensler, 1999) [26]. At that time, discrete time modeling was the most popular field in System Identification. CONSTID was mainly designed for estimating continuous time black box models without having to fully characterized the mathematics governing the system behavior [26]. This toolbox consists of standard tools for continuous time System Identification such as simple process, transfer function and state space models [26]. It also provides advanced tools like error-in-variables and closed-loop model estimation. The toolbox uses commands like TFSRIVC, PROCSRIVC, COE, SIDGPMF for estimating transfer function models, process models, polynomial models, and state space models respectively [26]. Several of these routines are capable of estimating MIMO systems. It also advocates using the simple System Identification flowchart which is shown in the figure below.

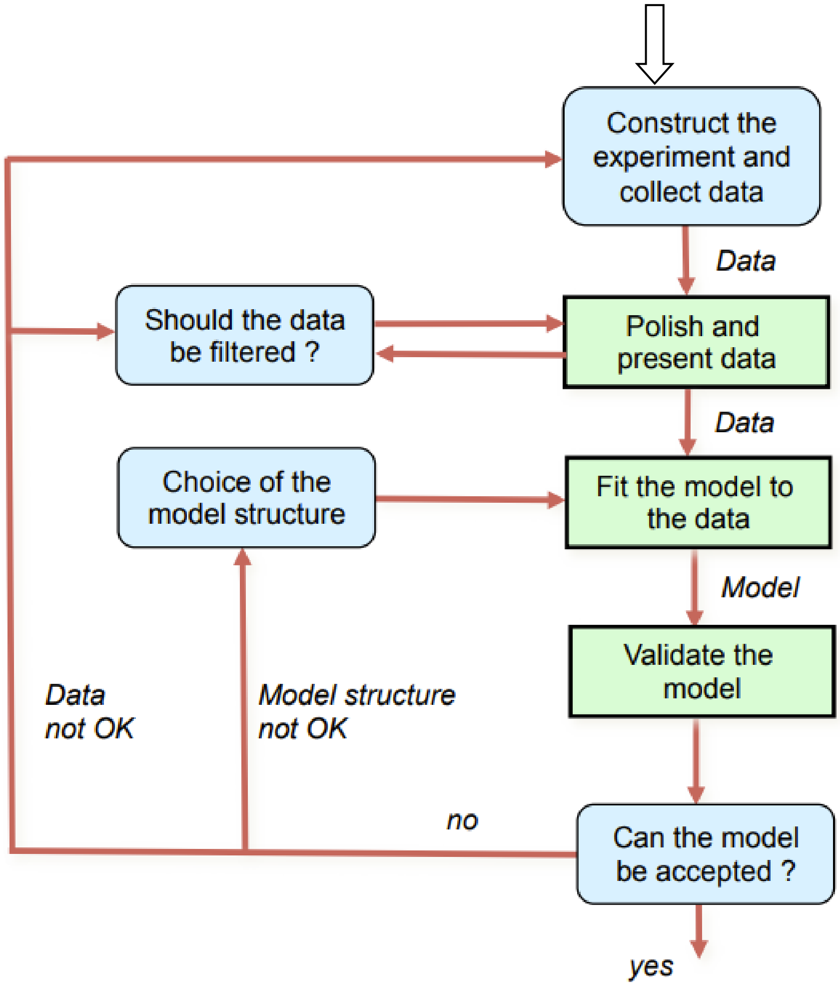


Figure 35. System Identification Procedure Suggested by CONSTID [26]

For our research, noisy and irregularly spaced data was provided by IDEQ, and it was "polished and presented" using smoothing splines in MATLAB. We know physically that our system is a continuous time system, and that is why we attempted to make use of CONTSID, as it was developed specifically for modeling continuous time system. It should be noted that the CONTSID toolbox is still in development, and in many cases the documentation and guidance for using its routines are vague on even or non-existent.

The first experiment we attempted was to estimate a continuous time transfer function, as we obtained the best results using the System Identification toolbox in MATLAB with TFEST. The corresponding CONTSID routine was COE (Continuous Output Error).

In experimenting with COE, we first attempted a SISO model with ambient temperature as the input, and water temperature as the output, using the mean centered, first three years of data excluded, weekly dataset, divided into two halves. COE gave essentially equivalent fits to the data as did TFEST. See the figure below for a comparison of the three fits on the weekly dataset (sys is the output model from the TFEST using the entire dataset as estimation data, sys-1 is the COE model using the first half of the dataset as the estimation data, and sys-2 is the COE model using the second half of the dataset the estimation data). All three are essentially equally good.

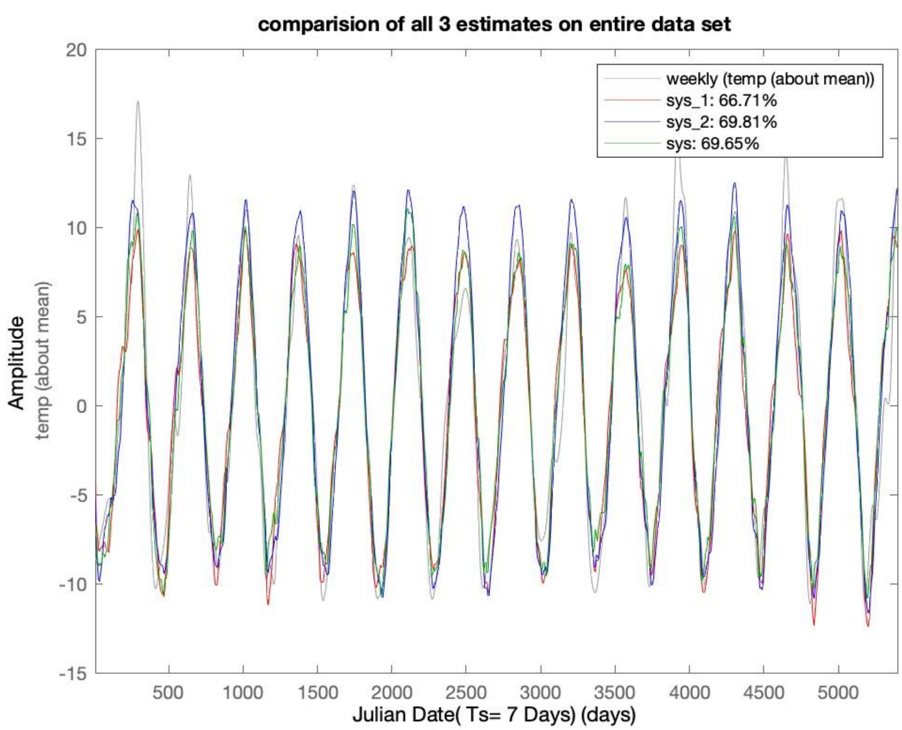


Figure 36. Comparison of Two Halves Weekly Data Using COE SISO

We then attempted to use COE for the MISO and MIMO modeling, but couldn't get COE to accept our model, getting run-time errors "incorrect model structure" or "unknown idmodel". As the documentation was lacking, we abandoned COE for computing transfer function estimates, as the TFEST model we obtained from the System Identification toolbox was equivalent, at least for SISO models, as those from COE.

The estimation didn't go according to our expectations. With SIDGPMF, the best result obtained was a fit of -0.7% for water temperature and -0.29% for dissolved oxygen. The figure of the validation is given below. Clearly, the estimate is giving us a model which says the best fit is the mean of the outputs and tracks none of the variations about the mean values.

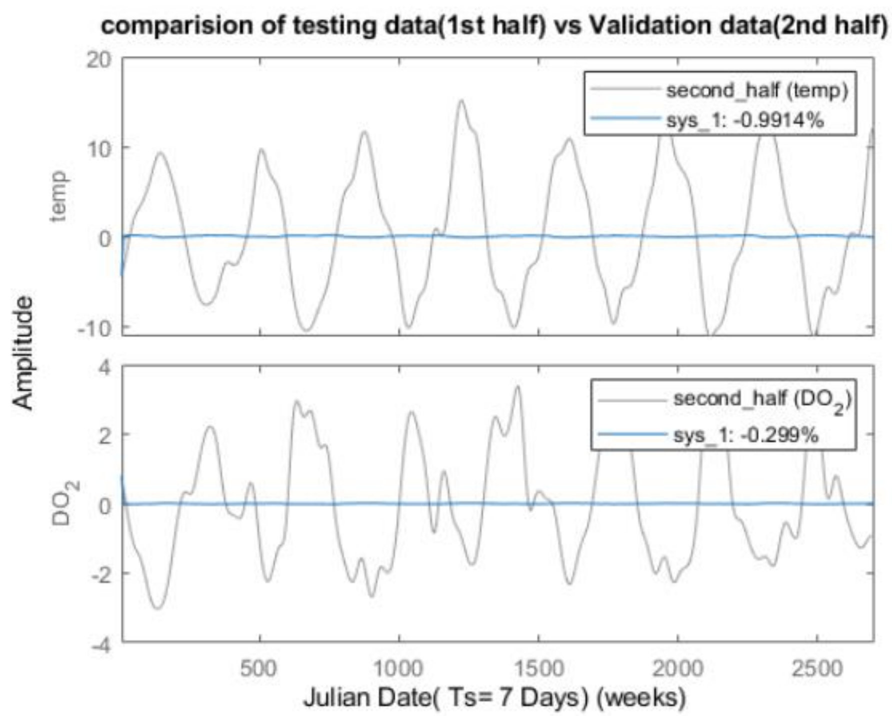


Figure 37. Comparison of Two Halves Weekly Data Using SIDGPMF

As the documentation for CONTSID was lacking in guidance on how to use the routines and how to adjust their calling parameters, we decided not to investigate the use of CONTSID any further until such guidance is available.

Chapter 6. CONCLUSIONS AND RECOMMENDATIONS

This chapter presents the summary of the research project and provides conclusions based on the findings throughout the research project. This research was set out to determine a suitable mathematical model for the Portneuf River ecosystem.

Management of river ecosystems is becoming increasingly important for now and into the future. In this paper, we tried to find a suitable mathematical model which can represent a small portion of the river ecosystem with the aim to improve the environmental outcomes. This project could have been a boon for all those aquatic and non-aquatic lives which depends on the river ecosystem. By experimental validation, it was found that the System Identification for a river ecosystem with the irregular data structure is very difficult. With a series of hundreds of experiments having been conducted, this research project was only partially successful.

First and foremost, nearly all System Identification routines need regularly spaced data in time. We attempted to overcome this problem (our raw dataset had irregular data) by substituting a callable smoothing spline approximation to the data, in order to produce regularly spaced samples. It appears that this approach was reasonable. We focused only on estimating LTI models to the data. This is always the first approach used in System Identification. In our recommendations, we suggest possible nonlinear grey-box modeling that may be appropriate.

The best LTI validation model obtained was when a transfer function model was used with data excluding first three years. The results obtained using state space models were not quite satisfactory. From our series of experiments, it can be concluded that the models using weekly datasets were far more suitable than when using the daily and monthly datasets. This might be because the weekly datasets had about 1000 sample points whereas daily data had almost five thousand sample points and monthly data had one hundred and sixty data points. From this research it can be concluded that even with a big dataset, System Identification might not be feasible. In smaller datasets there might arise the problem of too many free parameters. We also conclude that the irregularities in the original raw dataset can play a huge role in System Identification.

The research went through lots of hurdles (for instance, getting a huge negative percentage fit while doing the validation) but research was continued and at the end the project was partially successful. The most important part of this thesis was to use every single idea that comes to mind and trying it without worrying about the potential result. Sometimes these ideas had merit, and sometimes not. Nevertheless, future research and investigation should be conducted to discover the new mode of analysis in natural ecosystems. Some of the recommended research topics for future studies are:

1. As our LTI modeling did not give us highly accurate predictive models suitable for river management, and we know (physically, chemically, and biologically) that many of our variables involve dynamics that are either linear time varying linear with time delays, or truly nonlinear. Investigation into these types of models should be the primary focus of future research. Recommendations 2, 5, 6, and 7 below are to this end.
2. Conducting the same experiment with a suitable nonlinear grey-box model. This will help in better understanding of the elements that are present in the ecosystem which might directly and indirectly affect the river ecosystem. This approach is briefly described in section 6.1.
3. Conducting the same experiments with a different set of complete data, probably from another river. This may have lower irregularities in the data which means there will be a better chance of finding a suitable model.

4. Improve data quality: remote sensing at regularly spaced short time intervals (e.g., hourly, or daily) should be feasible for some of the variables in the dataset.
5. Conducting the same experiment using “machine learning” (non-linear models) to produce better predictive models.
6. Conducting the same experiment using “neural networks” (non-linear models) to produce better predictive models.
7. Linking “upstream” data to “downstream” models, when performing the estimation process, see section 6.2.

6.1 Grey Box Models

Grey-box models of a dynamic system entail specification of the form of the dynamics, parametrized by a finite set of parameters. The form of the system dynamics can be linear or non-linear. Grey-box models are based on either physical principles or intuition, guided by physical, chemical, or biological principles.

For our system, one possible non-linear grey-box model would be as follows (we give only the grey-box model for water temperature, a similar model can be given for dissolved oxygen).

Let,

$u_1(t)$ denote flowrate (m^3/sec),

$u_2(t)$ denote ambient temperature (^oC),

And $x_1(t)$ denote water temperature (^oC).

Then the grey-box model, based on Newton’s Law of Cooling is:

$$x_1 = f_1(u_1) \cdot (u_2 - x_1) \cdot H(x_1)$$

Where,

H(x) is the "Heaviside" function,

And, $f_1(u_1) = k_1 e^{-\alpha u_1}$

This model reflects that liquid water can't get cooler than 0^0C (that's the role of $H(x_1)$) and that the faster the water is flowing, the lower the heat transfer rate (that's the role of $f_1(u_1)$). So, this is a non-linear model with two free parameters: k_1 and $\propto$. The MATLAB System Identification toolbox has grey-box model estimation routines (GREYEST) which utilize non-linear optimization routines in conjunction with an ODE integrator (ODE45) to attempt to find the unknown parameters. This should be attempted for our system.

6.2 Linking Upstream Data

We know that our flow data (cms) obtained by smoothing spline imputation is flawed: the flow of the Portneuf River from Chesterfield Reservoir downstream is (with the exception of the winter and spring time natural flows and runoff events) nearly completely controlled by the "canal company" (irrigation) managers. Our smoothing spline estimates give a C^2 smooth approximation to the monthly data. The more realistic scenario is a sequence of "step-function" inputs (nearly discontinuous) overlaid with noisy fluctuations. See figure (36) for the annual flow, and figure (37) for the irrigation season flow, at the Topaz Station.

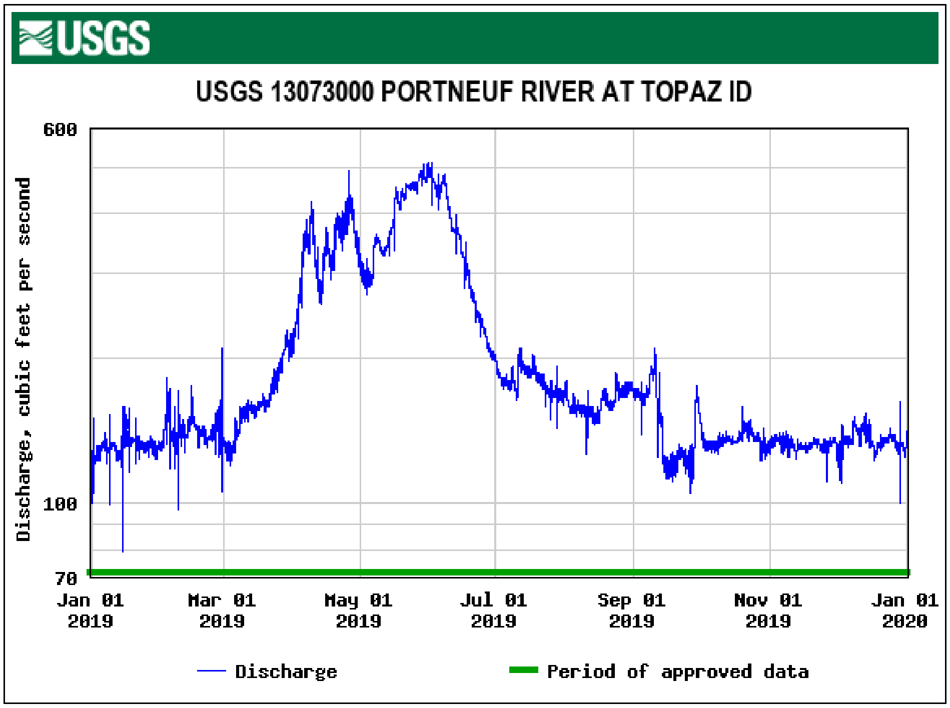


Figure 38. Topaz Annual Flow.

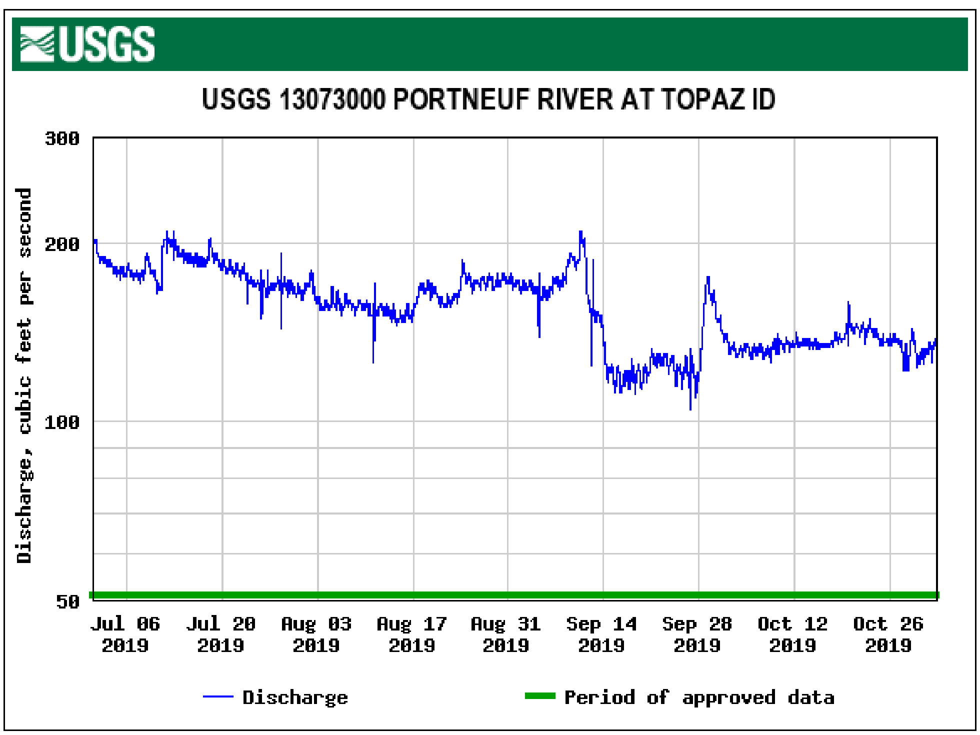


Figure 39. Topaz Irrigation Flow

In the annual flow at Topaz, we can clearly see the natural “spring runoff” from March through June. In the irrigation season figure, we can see the piecewise step control determined by the canal company (overlaid with natural fluctuations).

So, one suggestion for further research is to use an upstream flow gauge measurement for flow rates (daily) as another non-local input to the downstream system. This would introduce time delays into all of the models and the nature of these delays would need to be investigated. Transfer function models can easily handle time delays, so this approach may be manageable.

REFERENCES

[1] Kerschen G, Worden K, Vakakis AF, Golinval J-C. Past, present and future of nonlinear System Identification in structural dynamics. *Mechanical Systems and Signal Processing*. 2006;20(3):505-592. doi:10.1016/j.ymssp.2005.04.008

[2] M. Gevers, A personal view of the development of System Identification, IEEE Control Systems Magazine, 2006

[3] Coghill GM, King RD, Srinivasan A. Qualitative System Identification from Imperfect Data. 2011. doi:10.1613/jair.2374

[4] Xing Wang, Hill TL, Neild SA, Shaw AD, Haddad Khodaparast H, Friswell MI. Model updating strategy for structures with localised nonlinearities using frequency response measurements. *Mechanical Systems and Signal Processing*. 2018;100:940-961. doi:10.1016/j.ymssp.2017.08.004

[5] https://www.mathworks.com/help/ident/gs/about-system-identification.html

[6] Ljung, L. (2008). Perspectives on System Identification. IFAC Proceedings Volumes, 41(2), 7172–7184. https://doi.org/10.3182/20080706-5-kr-1001.01215

[7] Wigren T, Schoukens J. Three free datasets for development and benchmarking in nonlinear System Identification. 2013 European Control Conference (ECC), Control Conference (ECC), 2013 European. July 2013:2933-2938. Accessed July 6, 2022. https://search-ebscohost-com.libpublic3.library.isu.edu/login.aspx?direct=true&db=edseee&AN=edseee.6669201&site=eds-live&scope=site

[8] Nasir HA, Weyer E. Comparison of prediction error methods and subspace identification methods for rivers. 2013 Australian Control Conference, Control Conference (AUCC), 2013 3rd Australian. November 2013:415-420. doi:10.1109/AUCC.2013.6697309

[9] Nasir HA, Weyer E. System Identification of the upper part of Murray river. 2014 European Control Conference (ECC), Control Conference (ECC), 2014 European. June 2014:1355-1360. doi:10.1109/ECC.2014.686238

[10] J. Sjöberg and L. Ljung, "Overtraining, regularization, and searching for minimum in neural networks" Proc. Symp. On Adaptive systems in Control and Signal Processing. Grenoble, Switzerland. 1992.

[11] L. Ljung, System Identification – Theory For The User. 2nd Ed., Prentice Hall, Upper Saddle River, N.J., 1999.

[12] Farms, Rivers and Market Project. (2012, Dec. 25) [Online]. Available: http://www.frm.unimelb.edu.au/default.htm

[13] I. Mareels, E. Weyer, S. K. Ooi, M. Cantoni, Y. Li, and G. Nair, "Systems engineering for irrigation systems: Successes and challenges," Annu. Rev. Control, vol. 29, no. 2, pp. 191–204, Aug. 2005.

[14] H. Linke, "A model-predictive controller for optimal hydro-power utilization of river reservoirs," in Proc. IEEE Multi Conf. Syst. Control, Sept. 2010, pp. 1868–1873

[15] M. Maxwell and S. Warnick, "Modelling and identification of the Sevier River system," in Proc. Amer. Control Conf., Jun. 2006, pp. 5342–5347.

[16] “Spline Models for observational Data”, by Grace Wahba, CBMS-WSF Regional Conference Series in Applied Mathematic, SIAM, 1990. Isbn: 978-0-89871-244-5

[17] https://idwr.idaho.gov/

[18] https://en.wikipedia.org/wiki/Portneuf_River_(Idaho)

[19] https://www.visitpocatello.com/things-to-do/portneuf-river/

[20] https://river.pocatello.us/

[21] P.C. Young, A. Castelletti, and F. Pianosi. The data-based mechanistic approach in hydrological modelling. Topics on System Analysis and Integrated Water Resource Management, pages 27–48, 2007

[22] M. Papageorgiou and A. Messmer. Flow control of a long river stretch. Automatica, 25(2):177–183, 1989.

[23] B. Sohlberg and M. Sernfalt. Grey box modelling for river control. Journal of Hydroinformatics, 4:265–280, 2002

[24] W.E. Larimore. Canonical Variate Analysis in identification, filtering, and adaptive control. In Proceedings of the 29th IEEE Conference on Decision and Control, 1990, pages 596–604.

[25] X. Litrico. Robust IMC flow control of SIMO dam-river open channel systems. IEEE Transactions on Control Systems Technology, 10(3):432–437, 2002.

[26] Garnier H, Gilson M. CONTSID: a Matlab toolbox for standard and advanced identification of black-box continuous-time models. *IFAC - Papers Online*. 2018;51(15):688-693. doi:10.1016/j.ifacol.2018.09.203

APPENDICES

Appendix will show all the experiment carried out for this project.

Appendix 1. Regular System Identification

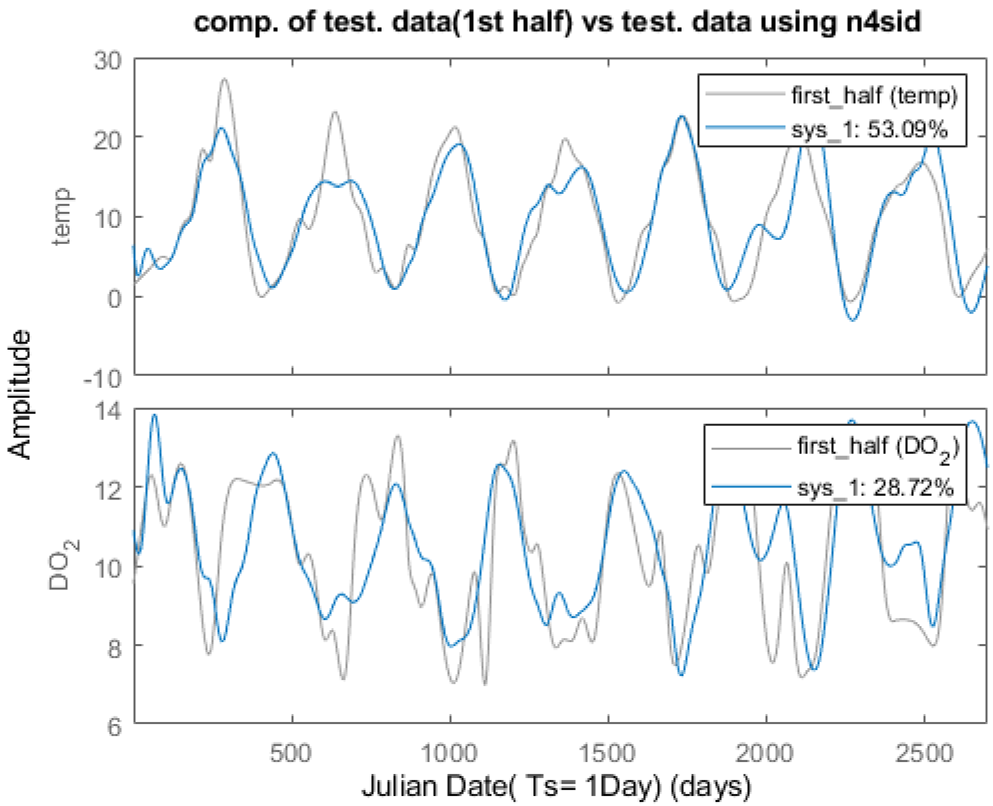
1. *Data Into Two halves using N4SID*

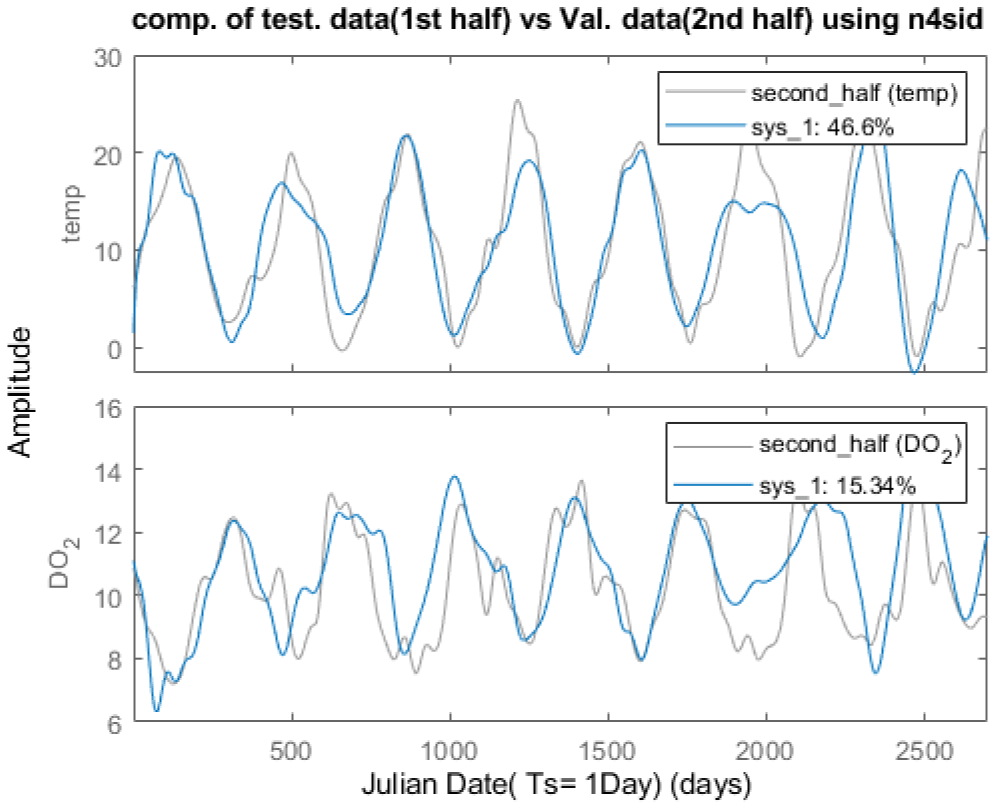
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of daily data using n4sid for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%SS = State Space
close all;
clc;
clear;
load fichter_daily.mat;% load the fichter daily dataset
y=[Temp,DO_2];%load the output of the system
u=[Ambtemp_time,cms];%load the input of the system
daily=iddata(y,u,Ts);%converting input and out into iddata
%giving names to inputs and outputs
daily.inputname(1)={'Ambtemp'};
daily.inputname(2)={'cms'};
daily.timeunit='days';
daily.outputname(1)={'temp'};
daily.outputname(2)={'DO_2'};
sys=n4sid(daily,4)
first_half= iddata(y(1:2698,:),u(1:2698,:),Ts);%converting first half to iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='days';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y(2699:5396,:),u(2699:5396,:),Ts);%converting second half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='days';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
sys_1=n4sid(first_half);%using n4sid to first half
sys_2=n4sid(second_half);%using n4sid to second half
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using n4sid')
xlabel('Julian Date( Ts= 1Day)')
```

```
figure ()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using n4sid')
xlabel('Julian Date( Ts= 1Day)')
```

```
sys =
  Discrete-time identified state-space model:
    x(t+Ts) = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
              x1          x2          x3          x4
   x1      1.001   6.246e-05    0.007631    -0.01349
   x2    0.01048      0.9985     0.04458     0.01391
   x3   -0.01413    -0.01209      0.9944    0.006725
   x4    0.01204    -0.01099   -0.003987      0.9863

  B =
          Ambtemp          cms
   x1  -3.855e-08    1.043e-05
   x2  -4.627e-07     3.11e-05
   x3  -7.197e-06    1.719e-05
   x4  -2.772e-05   -7.825e-05

  C =
             x1      x2      x3      x4
   temp   -84.58  -192.7  -4.606  -0.766
   DO_2    209.2    15.3   1.133  -1.293

  D =
         Ambtemp      cms
   temp        0        0
   DO_2        0        0

  K =
             temp        DO_2
   x1  0.0002307    0.003112
   x2   -0.00342   -0.001268
   x3   -0.01649    0.008991
   x4   -0.01296    -0.05392

Sample time: 1 days

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 40
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using N4SID on time domain data "daily".
```

```
Fit to estimation data: [99.55;99.11]% (prediction focus)
FPE: 2.407e-07, MSE: 0.001262
```





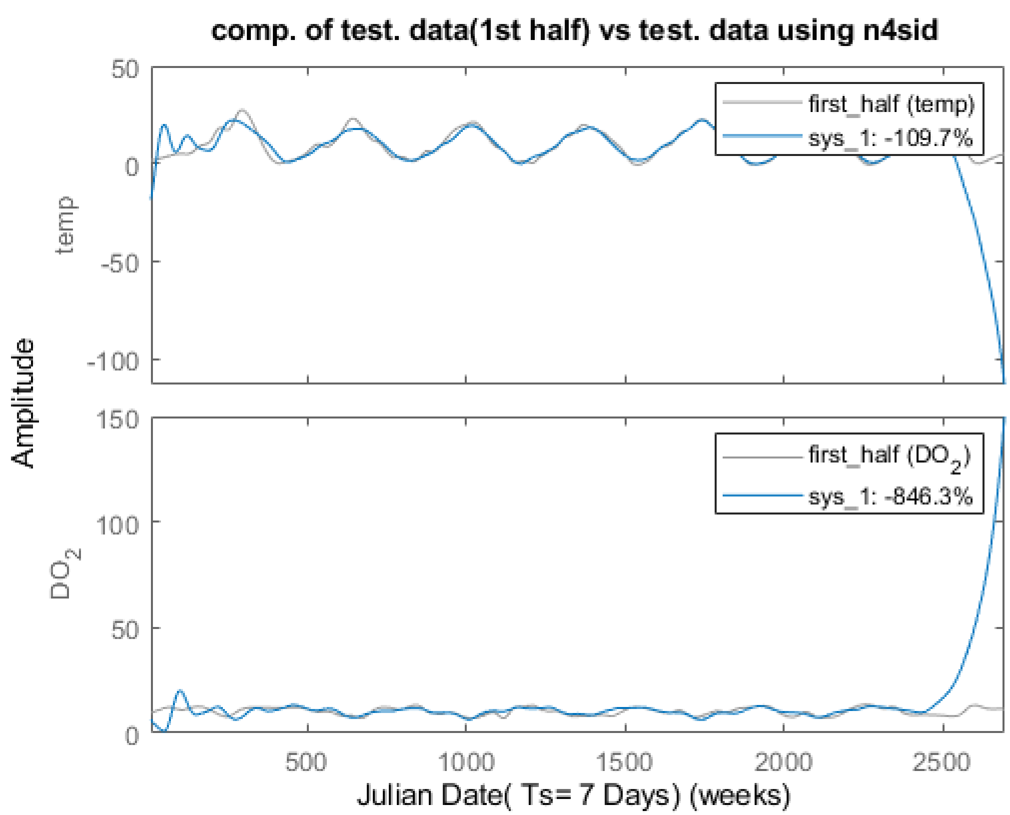
Published with MATLAB® R2020b

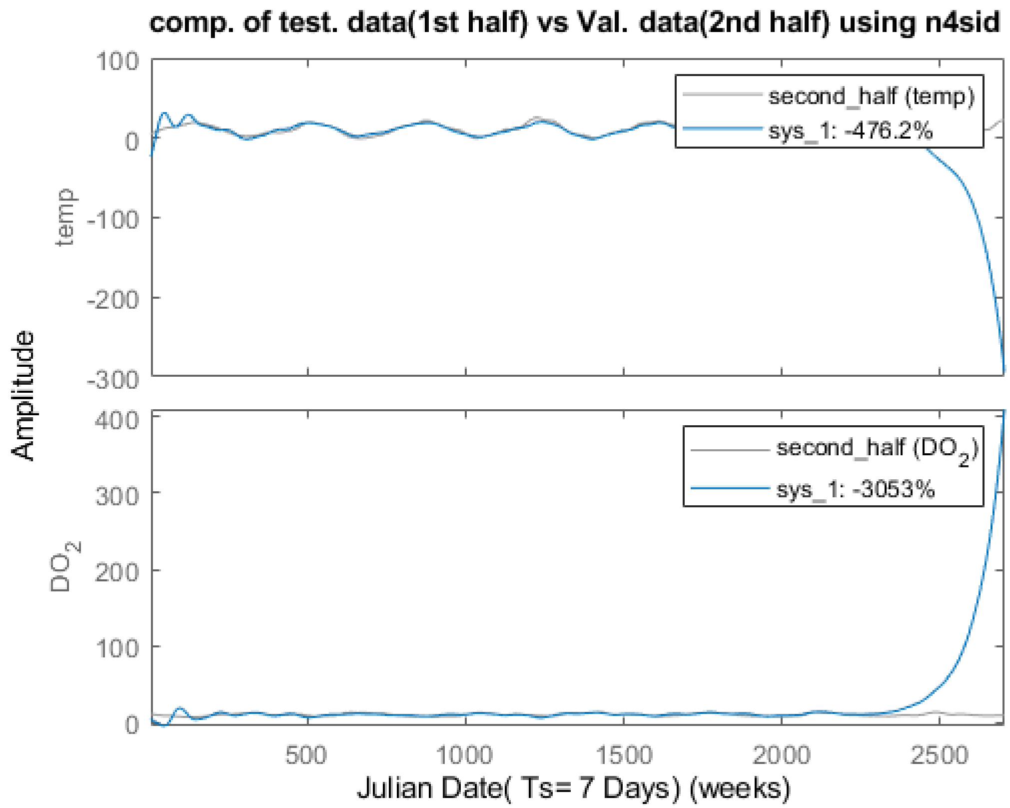
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of weekly data using n4sid for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%SS = State Space
close all;
clc;
clear;
load fichter_weekly.mat;% load the fichter weekly dataset
y7=[Temp,DO_2];%load the output of the system
u7=[Ambtemp_time,cms];%load the input of the system
weekly=iddata(y7,u7,Ts);%converting input and out into iddata
%giving names to inputs and outputs
weekly.inputname(1)={'Ambtemp'};
weekly.inputname(2)={'cms'};
```

```
weekly.timeunit='weeks';
weekly.outputname(1)={'temp'};
weekly.outputname(2)={'DO_2'};
sys=n4sid((weekly),4)
first_half= iddata(y7(1:385,:),u7(1:385,:),Ts);%converting first half into iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='weeks';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y7(386:771,:),u7(386:771,:),Ts);%converting second half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='weeks';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
sys_1=n4sid((first_half));%using n4sid to first half data
sys_2=n4sid((second_half));%using n4sid to second half data
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using n4sid')
xlabel('Julian Date( Ts= 7 Days)')
figure ()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using n4sid')
xlabel('Julian Date( Ts= 7 Days)')
```

```
sys =
  Discrete-time identified state-space model:
    x(t+Ts) = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
             x1         x2         x3         x4
   x1    0.9983   0.007329   -0.01075     -0.147
   x2  -0.09743     0.8573     -0.266   -0.02123
   x3  -0.01655     0.1927     0.8768    0.07508
   x4    0.1153    0.07499    -0.1613     0.4361

  B =
           Ambtemp          cms
   x1    1.83e-05  -0.0009669
   x2   0.0001622   -0.002542
   x3     -0.0011   -0.001453
   x4  -0.0008112    -0.00684

  C =
             x1       x2       x3       x4
   temp  -22.25    68.19   -9.469   0.4193
   DO_2   69.36   -12.64    1.649   -3.749

  D =
```

```
         Ambtemp      cms
   temp        0        0
   DO_2        0        0

  K =
            temp       DO_2
   x1   0.002594    0.01475
   x2    0.01285    0.00252
   x3   -0.02334  -0.005484
   x4   0.001885   -0.01745

Sample time: 7 weeks

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 40
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using N4SID on time domain data "weekly".
Fit to estimation data: [91.71;86.22]% (prediction focus)
FPE: 0.02004, MSE: 0.3985
```





Published with MATLAB® R2020b

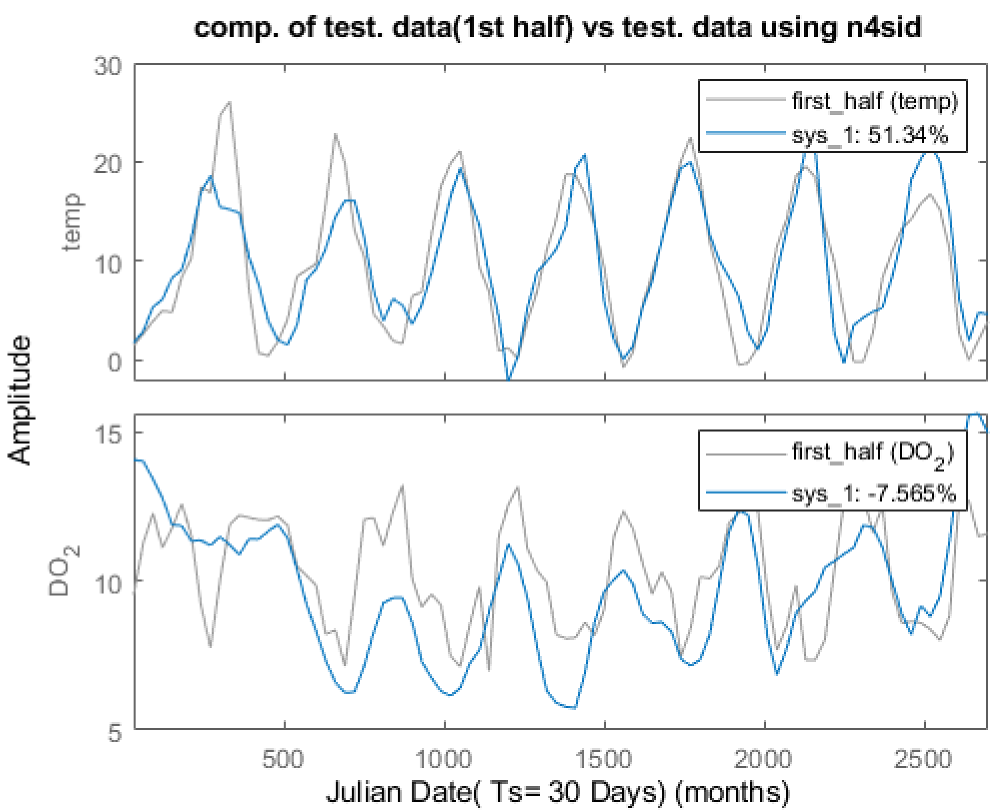
A) System Identification for Monthly Data

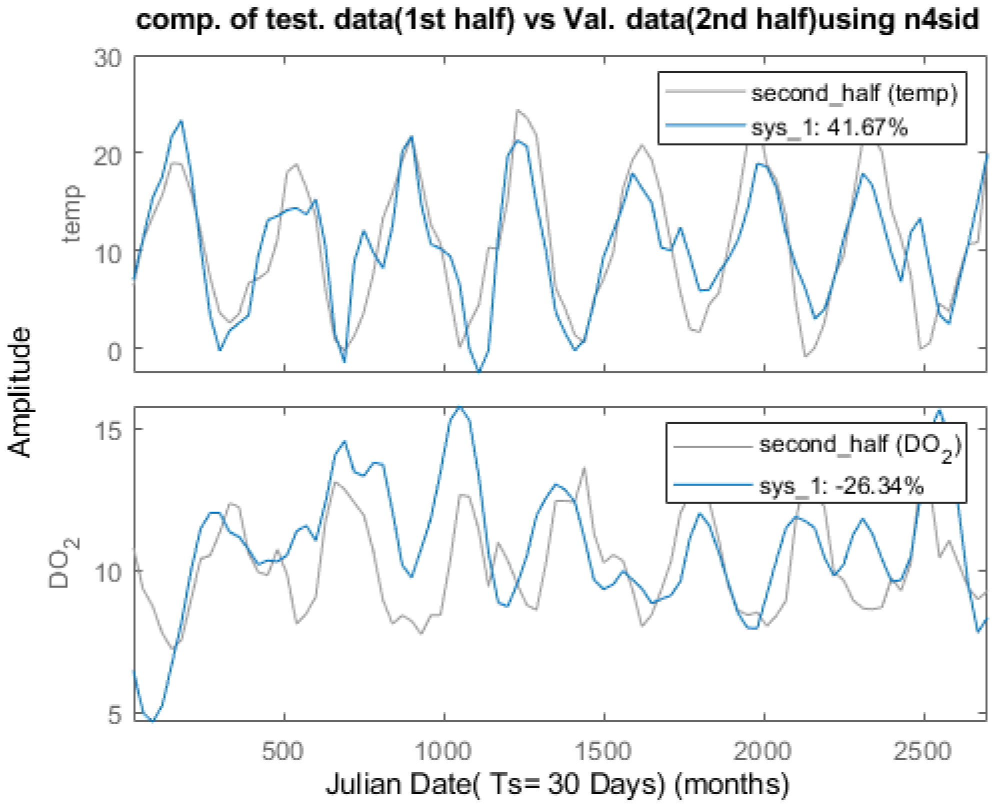
```
Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of monthly data using n4sid for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%SS = State Space
close all;
clc;
clear;
load fichter_monthly.mat;% load the fichter monthly dataset
y30=[Temp,DO_2];%load the output of the system
u30=[Ambtemp_time,cms];%load the input of the system
monthly=iddata(y30,u30,Ts);%converting input and out into iddata
%giving names to inputs and outputs
monthly=iddata(y30,u30,Ts);
```

```
monthly.inputname(1)={'Ambtemp'};
monthly.inputname(2)={'cms'};
monthly.timeunit='months';
monthly.outputname(1)={'temp'};
monthly.outputname(2)={'DO_2'};
sys=n4sid(monthly,4)
first_half= iddata(y30(1:90,:),u30(1:90,:),Ts);
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='Months';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y30(91:180,:),u30(91:180,:),Ts);
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='Months';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
sys_1=n4sid(first_half);
sys_2=n4sid(second_half);
figure ()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using n4sid')
xlabel('Julian Date( Ts= 30 Days)')
figure ()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half)using n4sid')
xlabel('Julian Date( Ts= 30 Days)')
```

```
sys =
  Discrete-time identified state-space model:
    x(t+Ts) = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
             x1         x2         x3         x4
   x1    0.9848    -0.1762    0.01964    0.08597
   x2    0.4343      0.743    -0.3995    -0.4319
   x3    0.1716     0.3682     0.8973    -0.2979
   x4  -0.03486    -0.1655    0.06417     0.3188

  B =
           Ambtemp           cms
   x1   -0.0006036     -0.002103
   x2   -0.0002874       0.00179
   x3    0.0003002      0.006488
   x4     -0.00139      0.001364

  C =
             x1       x2       x3       x4
   temp   2.766   -30.67     4.86   -7.341
   DO_2   14.36    6.376    5.125    3.425
```

```
  D =
         Ambtemp       cms
   temp        0         0
   DO_2        0         0

  K =
            temp        DO_2
   x1   0.002996      0.0083
   x2   -0.02022   -0.003607
   x3   0.002613     0.01622
   x4  -0.003646    0.002051

Sample time: 30 months

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 40
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using N4SID on time domain data "monthly".
Fit to estimation data: [70.97;39.31]% (prediction focus)
FPE: 6.065, MSE: 5.292
```





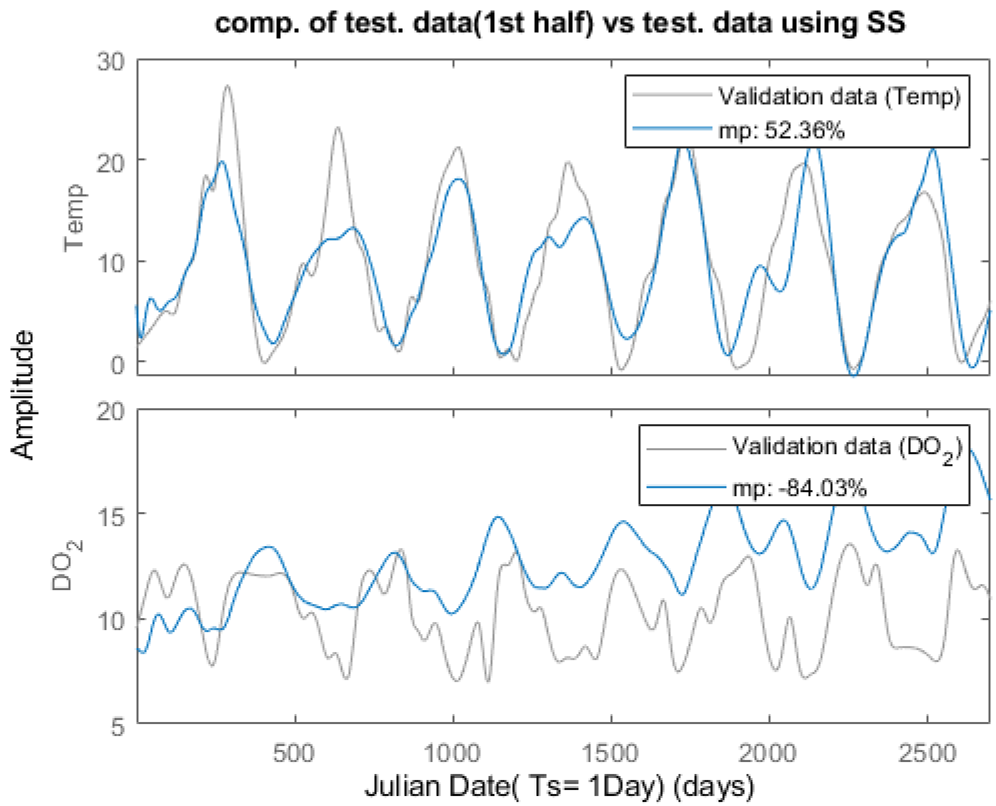
Published with MATLAB® R2020b

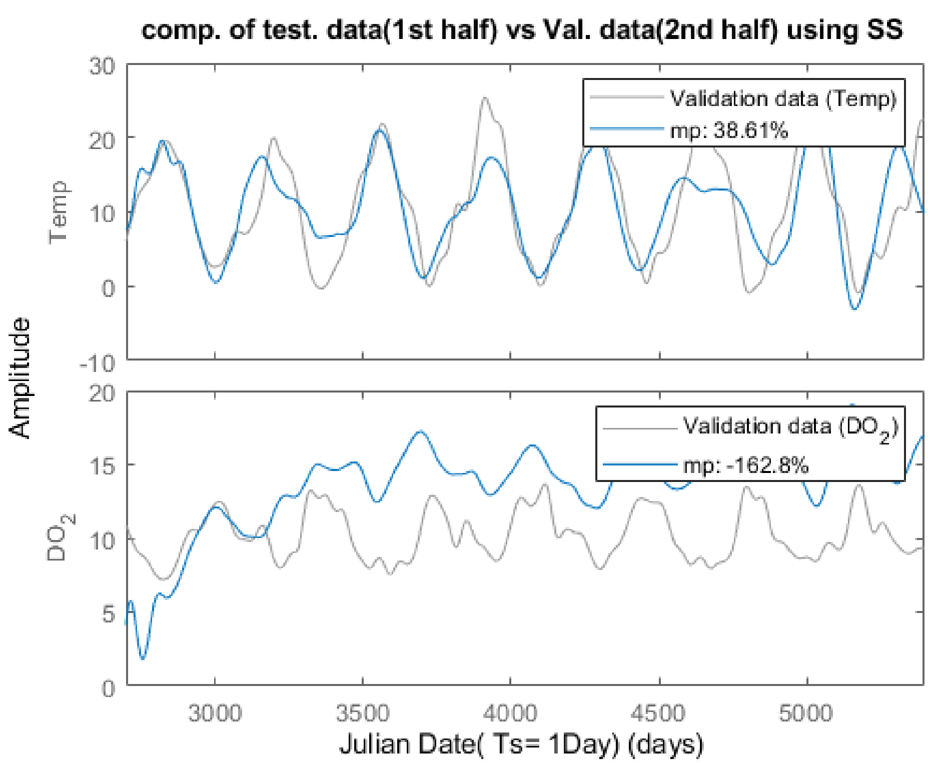
2. Data into Two Halves using SSEST

```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of daily data using state space for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%SS = State Space
close all;
clc;
clear;
load fichter_daily.mat;% load the fichter daily dataset
Daily=iddata([Temp,DO_2],[Ambtemp_time,cms],Ts); % Providing Inputs and outputs
Daily.InputName={'Ambtemp_time','cms'};% giving names to input
Daily.Outputname={'Temp';'DO_2'};% giving names to output
Daily.timeunit='days';% time unit
mp=ssest(Daily(1:2698))%This code will give state space structure to the first half
figure(7)
compare(Daily(1:2698),mp);%comparision of Testing data with testing data
title('comp. of test. data(1st half) vs test. data using SS')
xlabel('Julian Date( Ts= 1Day)')
figure(8)
compare(Daily(2699:5396),mp);%comparision of testing data with validation data
title('comp. of test. data(1st half) vs Val. data(2nd half) using SS')
xlabel('Julian Date( Ts= 1Day)')
```

```
mp =
  Continuous-time identified state-space model:
      dx/dt = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
                x1         x2         x3         x4         x5         x6
   x1      0.02807   0.005039   -0.07792     0.1246   -0.01054  -0.006166
   x2      0.03713   -0.02028    -0.1039    -0.1825    0.03211   -0.01607
   x3      0.02531  -0.003328    0.01891    0.02028  -0.004969     0.1355
   x4     -0.00171   0.008849   -0.02369    0.03188     0.1902     0.1026
   x5   -0.0009122  0.0005877    0.02511   -0.01186   -0.06021 -0.0005935
   x6     0.008679   -0.01363  -0.007812   -0.01776    0.01771   0.009759
   x7      0.01804   -0.01384    0.02515    0.02452   0.002296     0.1046

                x7
   x1     -0.01299
   x2     -0.02826
   x3       0.0353
   x4      0.04071
```

```
   x5     0.03655
   x6     -0.1269
   x7    -0.08821

  B =
       Ambtemp_time           cms
   x1     4.808e-09     7.461e-05
   x2     9.277e-09      0.000531
   x3     6.881e-07      0.000229
   x4     1.521e-07      6.75e-05
   x5     7.632e-07     2.297e-05
   x6    -4.903e-07    -1.405e-05
   x7    -2.969e-06     5.035e-05

  C =
                x1         x2         x3         x4         x5         x6
   Temp     -80.85       -109      2.727      1.358   0.002637     0.0747
   DO_2      154.8      2.616    -0.9803     0.8493    0.04113  -0.007785

                x7
   Temp  -0.004353
   DO_2    0.00189

  D =
         Ambtemp_time           cms
   Temp             0             0
   DO_2             0             0

  K =
           Temp      DO_2
   x1  0.002623   0.01335
   x2  -0.02969  -0.01248
   x3    0.1025  -0.08896
   x4   0.06582    0.1434
   x5    0.1238    0.7021
   x6     1.703   -0.3545
   x7   -0.9226    0.4348

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 91
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using SSEST on time domain data.
Fit to estimation data: [99.99;99.99]% (prediction focus)
FPE: 3.566e-15, MSE: 1.889e-07
```





PublishedwithMATLAB®R2020b

```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of weekly data using state space for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%SS = State Space
close all;
clc;
clear;
load fichter_weekly;%load fichter weekly data
weekly=iddata([Temp,DO_2],[Ambtemp_time, cms],Ts);%Putting input and output
weekly.InputName={'Ambtemp_time';'cms'};%giving names to inputs
weekly.Outputname={'Temp';'DO_2'};%giving names to output
weekly.timeunit='weeks';%defining time unit
mp=ssest(weekly(1:386))%this code will give state space structure to first half
figure(8)
compare(weekly(1:386),mp);%comparision of Testing data with testing data
title('comp. of testing data(1st half) vs testing data')
xlabel('Julian Date( Ts= 7 Days)')
figure(9)
compare(weekly(387:771),mp);%comparision of testing data with validation data
title('comparision of testing data(1st half) vs Validation data(2nd half)')
xlabel('Julian Date( Ts= 7 Days)')
```

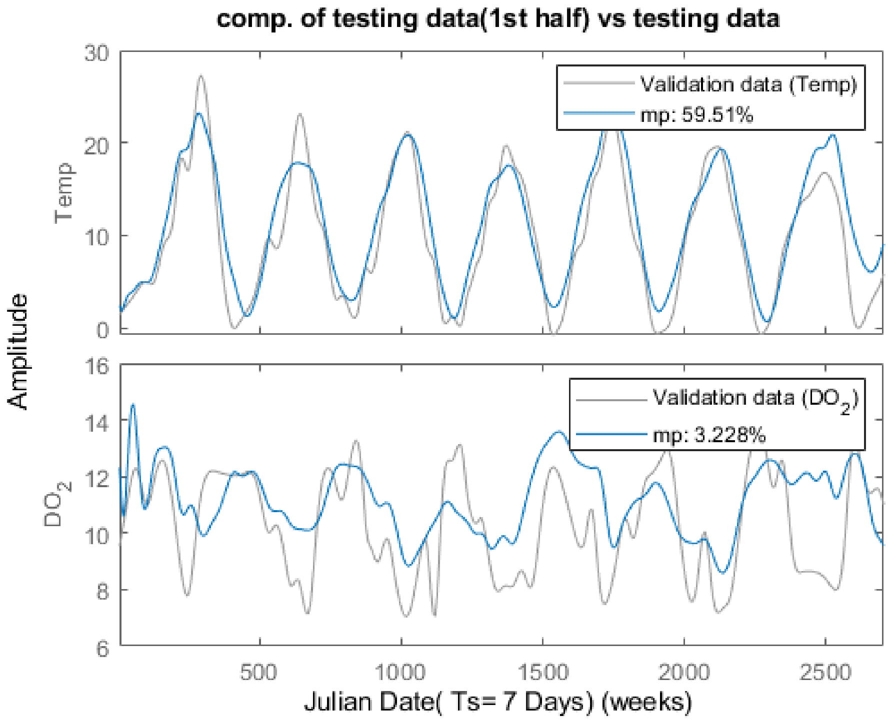
```
mp =
  Continuous-time identified state-space model:
      dx/dt = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
```

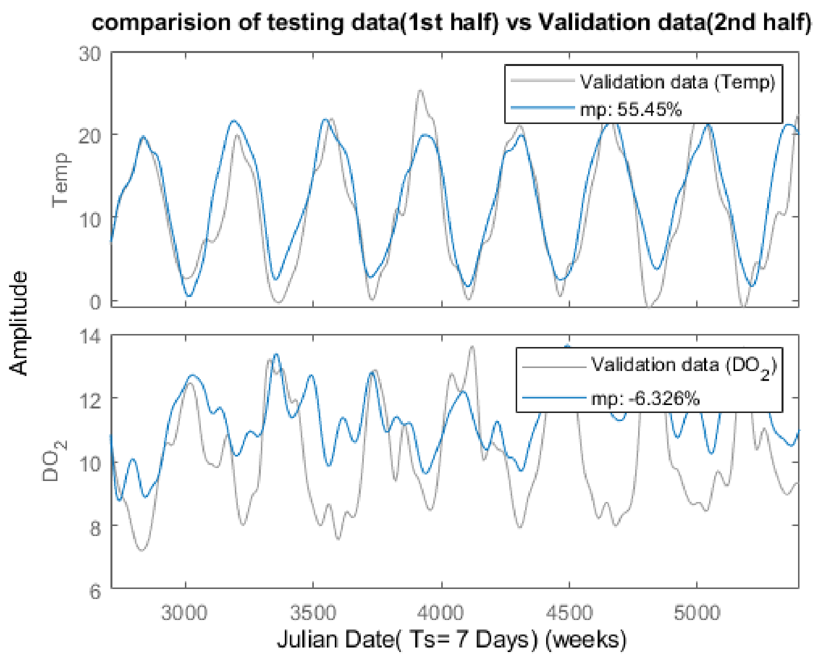
	x1	x2	x3	x4	x5	x6
x1	-0.02156	0.01368	-0.01897	-0.01753	-0.01122	1.09e-05
x2	-0.05126	0.01826	-0.04277	0.01302	-0.01297	0.009952
x3	0.01978	0.0075	0.01965	0.04217	-0.07488	-0.04987
x4	0.03868	-0.03512	-0.004	-0.003896	-0.06235	0.04112
x5	-0.04706	0.003492	0.08078	0.02503	-0.07322	-0.0003862
x6	-0.007414	0.005757	0.06376	-0.04407	-0.1028	-0.08296
x7	-0.04235	0.01824	0.02404	-0.03628	-0.03273	-0.001463
x8	-0.03895	0.009639	0.05886	0.01227	-0.1196	-0.09464
x9	0.003358	-0.006623	0.02077	-0.01662	-0.1239	-0.09235
x10	0.003169	-0.005518	-0.03388	0.05086	0.08703	0.1091

	x7	x8	x9	x10
x1	0.002305	-0.004828	-0.007049	-0.01642
x2	0.006113	-0.001794	-0.005163	-0.002751
x3	0.02222	0.006813	0.0007023	-0.01021
x4	0.03113	0.007449	-0.01652	-0.0004781

```
   x5      0.03657   -0.006154    -0.01316    0.002619
   x6      0.03981    -0.03459     0.01371    -0.07546
   x7      0.01074      0.0116    -0.04736     0.02106
   x8      0.09017   -0.003798     0.05387      0.0496
   x9       0.1039    -0.09215     0.05876     -0.0532
   x10     -0.0613    -0.01787    -0.04937    -0.04104

  B =
      Ambtemp_time           cms
   x1      1.342e-05     2.922e-06
   x2     -2.854e-05    -0.0009697
   x3      4.275e-05     -0.001312
   x4      0.0003697     0.0003883
   x5       0.001557    -0.0001603
   x6     -1.538e-05    -0.0002074
   x7       0.002787     -0.001298
   x8       0.002086     -0.005304
   x9      0.0001312     -0.004098
   x10     -0.001837      0.003983

  C =
              x1        x2        x3        x4        x5        x6        x7
   Temp    -7.294     43.91    -5.077     2.993    0.3676      1.04    0.1572
   DO_2     40.04    -11.27    -1.159    -2.723    0.5255   -0.2988   -0.2717

              x8        x9       x10
   Temp   -0.3128    -0.187  -0.08663
   DO_2    0.1076    0.1442   0.06142

  D =
       Ambtemp_time           cms
   Temp             0             0
   DO_2             0             0

  K =
            Temp       DO_2
   x1    0.001266   0.003792
   x2    0.004945  -0.001379
   x3   -0.009429   -0.03538
   x4   -0.002179    -0.0234
   x5    -0.02662    0.08255
   x6     0.05224   -0.06436
   x7    -0.08588    0.08641
   x8    -0.01759  -0.001117
   x9     0.02252   -0.05599
   x10    0.01058   -0.01522

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 160
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.
```

```
Status:
Estimated using SSEST on time domain data.
Fit to estimation data: [98.7;97.63]% (prediction focus)
FPE: 2.141e-05, MSE: 0.01019
```



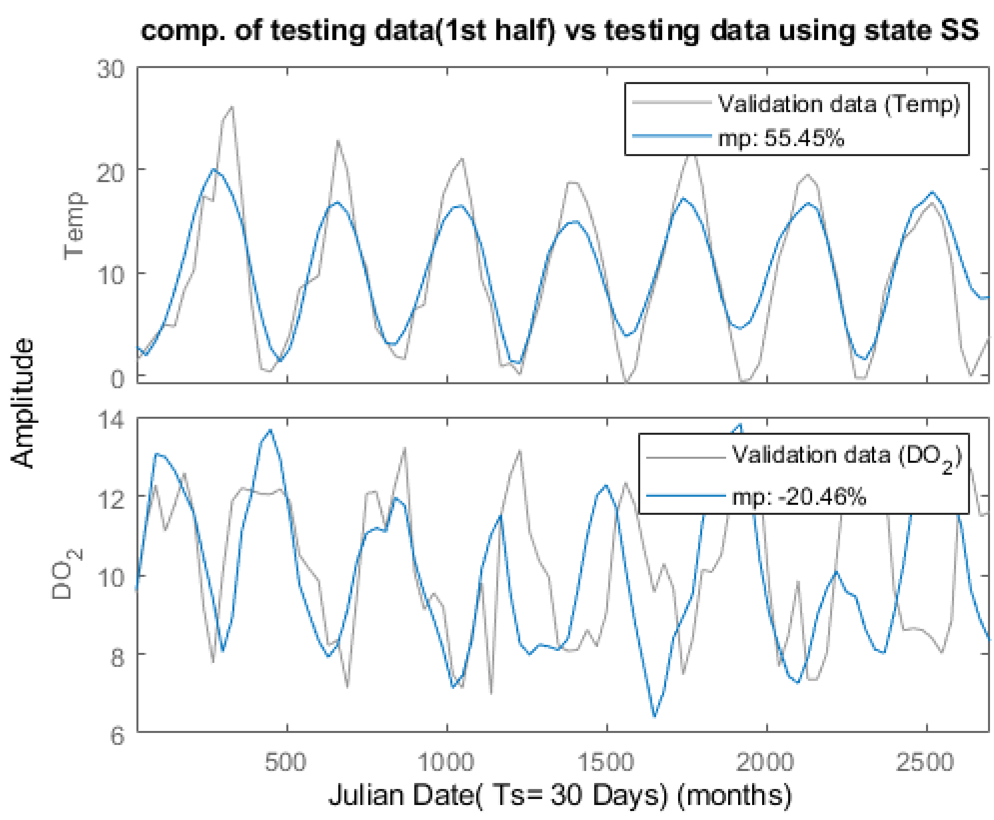


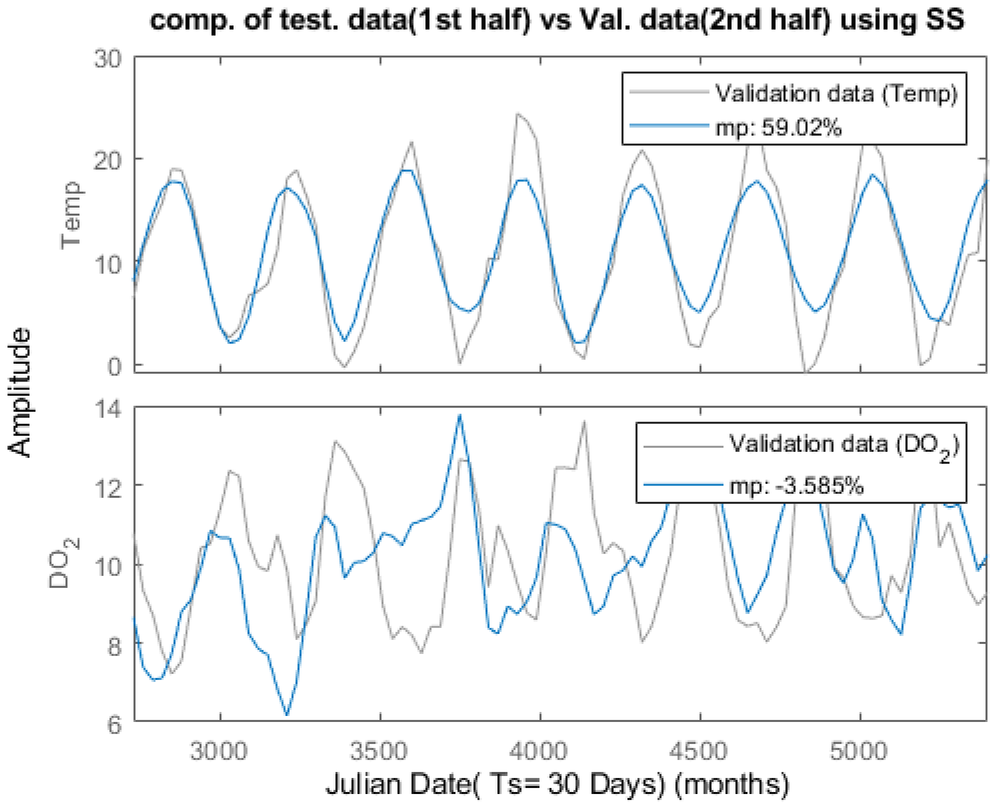
PublishedwithMATLAB®R2020b

```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of monthly data using state space for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%SS = State Space
close all;
clc;
clear;
load fichter_monthly.mat;% load the fichter monthly dataset
monthly=iddata([Temp,DO_2],[Ambtemp_time,cfs],Ts);%providing inputs and outputs
monthly.InputName={'Ambtemp_time';'cfs'};%giving name to inputs
monthly.Outputname={'Temp';'DO_2'};%giving names to output
monthly.timeunit='months';%defining timeunit
mp=ssest(monthly(1:45),4)%using ssest to monthly data
figure()
compare(monthly(1:90),mp);
title('comp. of testing data(1st half) vs testing data using state SS')
xlabel('Julian Date( Ts= 30 Days)')
figure()
compare(monthly(91:180),mp);
title('comp. of test. data(1st half) vs Val. data(2nd half) using SS')
xlabel('Julian Date( Ts= 30 Days)')
```

```
mp =
  Continuous-time identified state-space model:
      dx/dt = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
               x1          x2          x3          x4
   x1    0.002916    0.005355    -0.01767    0.004638
   x2    0.007281      0.0118    -0.03467     0.05468
   x3     0.01272     0.00284     -0.0103    0.008953
   x4    0.003695    -0.01819   -0.009386    -0.03928

  B =
       Ambtemp_time            cfs
   x1    -0.0001339     -0.0002814
   x2     0.0001744     -2.932e-05
   x3     3.435e-05     -0.0002831
   x4     0.0002836      0.0004966

  C =
             x1        x2        x3        x4
   Temp     8.41     5.429     14.41   -0.4989
   DO_2   -11.11     3.989    -2.678     -3.77
```

```
  D =
         Ambtemp_time           cfs
   Temp             0             0
   DO_2             0             0

  K =
             Temp         DO_2
   x1   6.047e-05    -0.000787
   x2    0.001041     0.003995
   x3    0.001021   -8.064e-05
   x4   0.0002493    -0.002633

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 40
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using SSEST on time domain data.
Fit to estimation data: [70.97;-6.386]% (prediction focus)
FPE: 48.85, MSE: 7.804
```



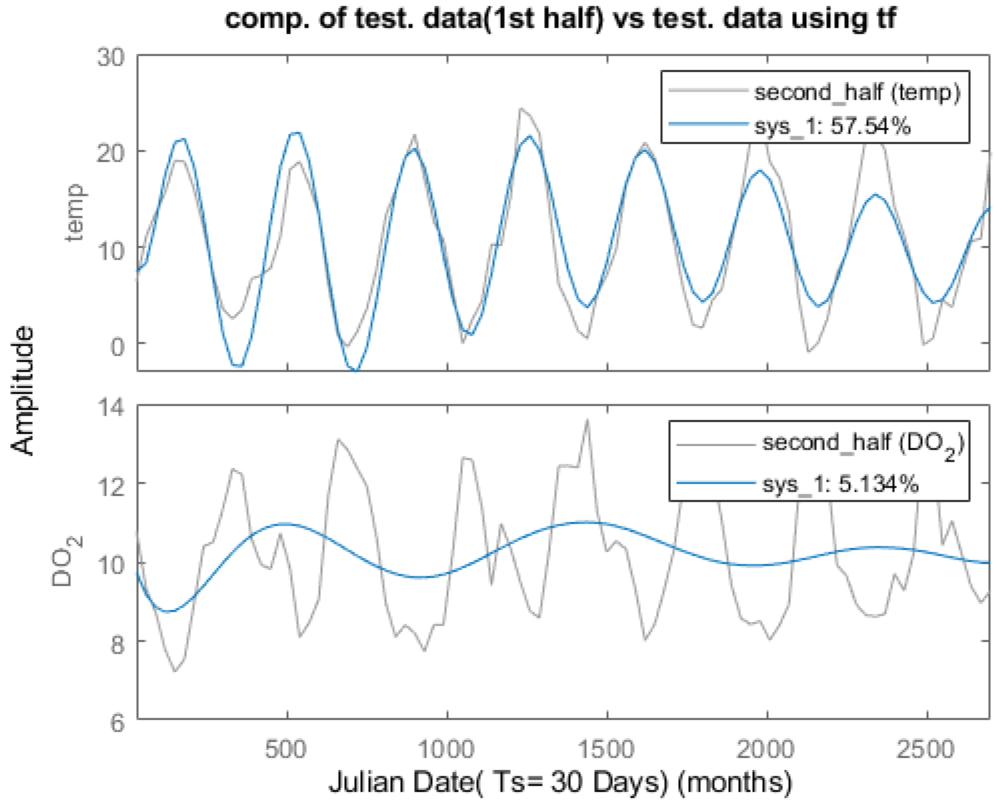


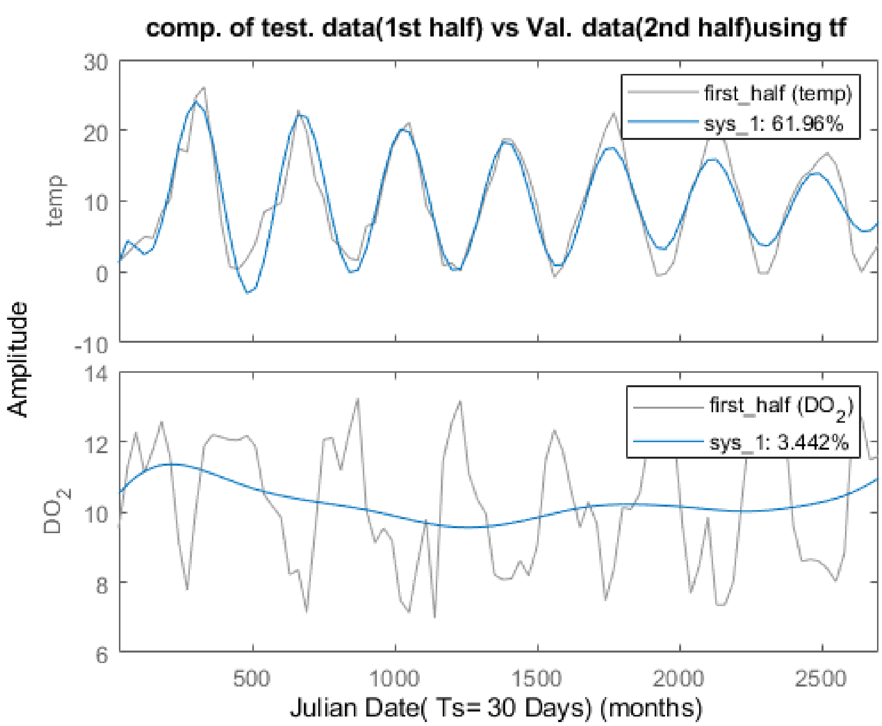
PublishedwithMATLAB®R2020b

3. Data into Two halves using TFEST

```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of monthly data using transfer function for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%tf= transfer function
close all;
clc;
clear;
load fichter_monthly.mat;% load the fichter monthly dataset
y30=[Temp,DO_2];%load the output of the system
u30=[Ambtemp_time,cms];%load the input of the system
monthly=iddata(y30,u30,Ts);%converting input and output into iddata
%giving names to inputs and outputs
monthly.inputname(1)={'Ambtemp'};
monthly.inputname(2)={'cms'};
monthly.timeunit='months';
monthly.outputname(1)={'temp'};
monthly.outputname(2)={'DO_2'};
sys=tfest(monthly,4)
first_half= iddata(y30(1:90,:),u30(1:90,:),Ts);
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='months';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y30(91:180,:),u30(91:180,:),Ts);
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='months';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
sys_1=tfest(first_half,4,0);
sys_2=tfest(second_half,4,0);
figure ()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs test. data using tf')
xlabel('Julian Date( Ts= 30 Days)')
figure ()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs Val. data(2nd half)using tf')
xlabel('Julian Date( Ts= 30 Days)')
```

```
sys =

  From input "Ambtemp" to output...
            0.003106 s^3 - 1.957e-05 s^2 + 1.391e-07 s + 7.669e-12
   temp:  ----------------------------------------------------------
          s^4 + 0.01075 s^3 + 0.0002844 s^2 + 5.88e-07 s + 1.329e-09

            0.001185 s^3 + 1.296e-05 s^2 + 3.371e-07 s - 7.788e-12
   DO_2:  -----------------------------------------------------------
          s^4 + 0.003818 s^3 + 0.0002747 s^2 + 9.334e-07 s + 1.42e-10

  From input "cms" to output...
            -0.01283 s^3 + 0.0004934 s^2 - 1.967e-07 s + 3.401e-09
   temp:  ----------------------------------------------------------
          s^4 + 0.02308 s^3 + 0.0001956 s^2 + 2.06e-07 s + 1.666e-09

             0.001685 s^3 + 3.445e-05 s^2 - 5.245e-07 s + 7.578e-11
   DO_2:  -------------------------------------------------------------
          s^4 + 0.0006167 s^3 + 0.0002767 s^2 + 1.209e-07 s + 4.854e-10

Continuous-time identified transfer function.

Parameterization:
   Number of poles: [4 4;4 4]   Number of zeros: [3 3;3 3]
   Number of free coefficients: 32
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "monthly".
Fit to estimation data: [66.72;10.88]%
FPE: 22.21, MSE: 7.883
```



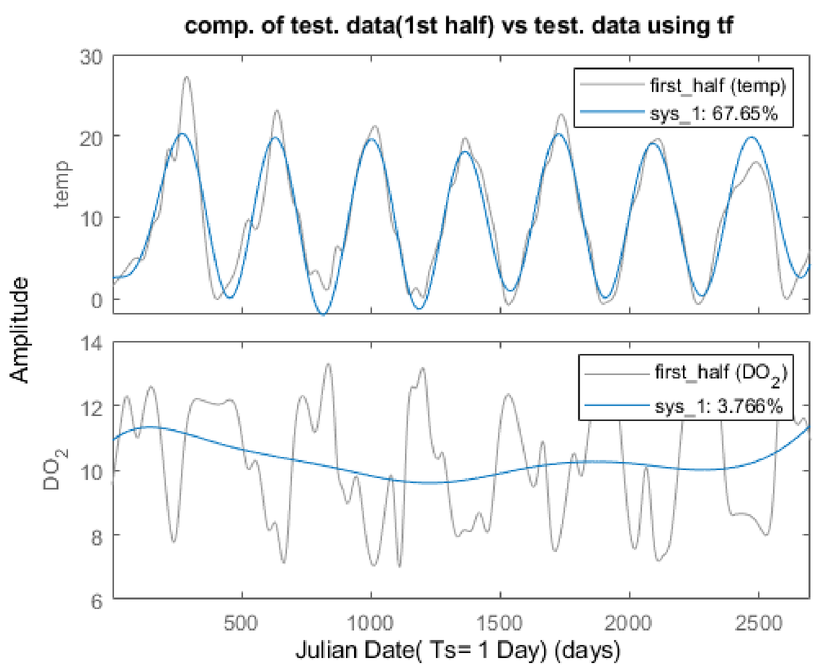


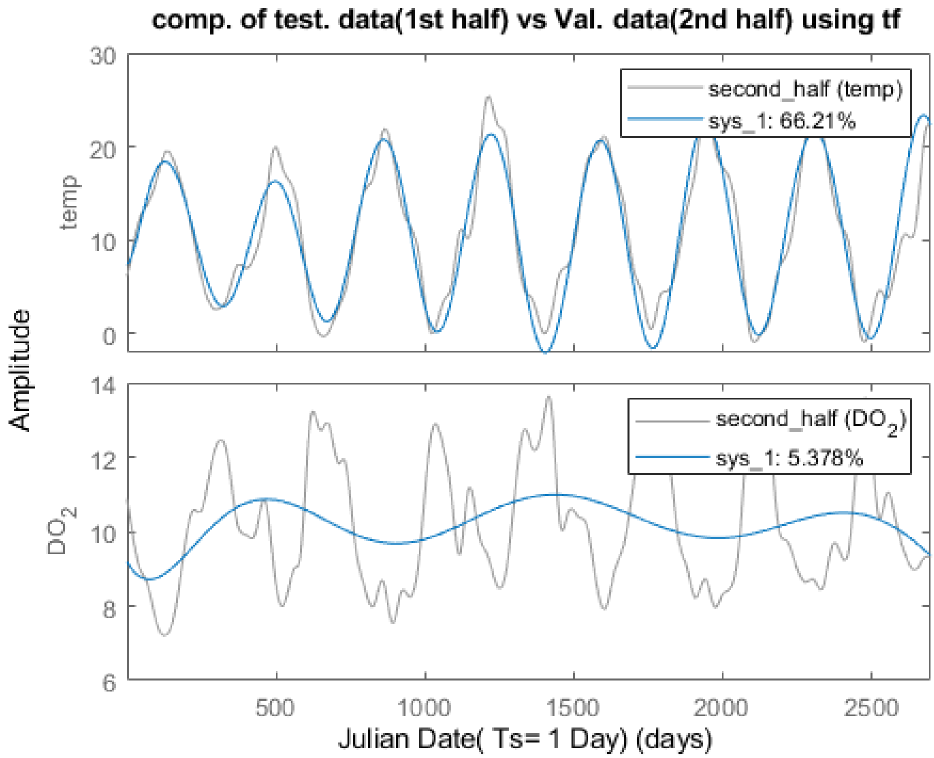
PublishedwithMATLAB®R2020b

```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of daily data using transfer function for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%tf= transfer function
close all;
clc;
clear;
load fichter_daily.mat;% load the fichter daily dataset
y=[Temp,DO_2];%load the output of the system
u=[Ambtemp_time,cms];%load the input of the system
daily=iddata(y,u,Ts);%converting input and out into iddata
%giving names to inputs and outputs
daily=iddata(y,u,Ts);
daily.inputname(1)={'Ambtemp'};
daily.inputname(2)={'cms'};
daily.timeunit='days';
daily.outputname(1)={'temp'};
daily.outputname(2)={'DO_2'};
sys=tfest(daily,4)
first_half= iddata(y(1:2698,:),u(1:2698,:),Ts);%converting first half into iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='days';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y(2699:5396,:),u(2699:5396,:),Ts);%converting second half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='days';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
sys_1=tfest(first_half,4,0);%using tf to first half
sys_2=tfest(second_half,4,0);%using tf to second half
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using tf')
xlabel('Julian Date( Ts= 1 Day)')
figure ()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using tf')
xlabel('Julian Date( Ts= 1 Day)')
```

```
sys =

  From input "Ambtemp" to output...
```

```
             0.01953 s^3 - 2.357e-05 s^2 + 5.162e-06 s + 2.212e-09
   temp:  -----------------------------------------------------------
          s^4 + 0.07611 s^3 + 0.0003599 s^2 + 2.217e-05 s + 7.154e-09

            0.001807 s^3 + 3.179e-05 s^2 + 9.522e-09 s + 3.442e-11
   DO_2:  -----------------------------------------------------------
          s^4 + 0.01446 s^3 + 0.0001513 s^2 + 9.161e-08 s + 2.391e-10

  From input "cms" to output...
            -0.003386 s^3 - 4.24e-05 s^2 + 8.691e-07 s - 2.207e-09
   temp:  -----------------------------------------------------------
          s^4 + 0.01157 s^3 + 0.0003086 s^2 + 3.208e-06 s + 4.333e-09

             0.005464 s^3 + 4.463e-05 s^2 + 1.036e-07 s + 6.057e-10
   DO_2:  ------------------------------------------------------------
          s^4 + 0.003354 s^3 + 0.0003148 s^2 + 7.572e-09 s + 7.055e-10

Continuous-time identified transfer function.

Parameterization:
   Number of poles: [4 4;4 4]   Number of zeros: [3 3;3 3]
   Number of free coefficients: 32
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "daily".
Fit to estimation data: [72.47;32.15]%
FPE: 5.147, MSE: 5.151
```





PublishedwithMATLAB®R2020b

Appendix 2. System Identification with Average Data

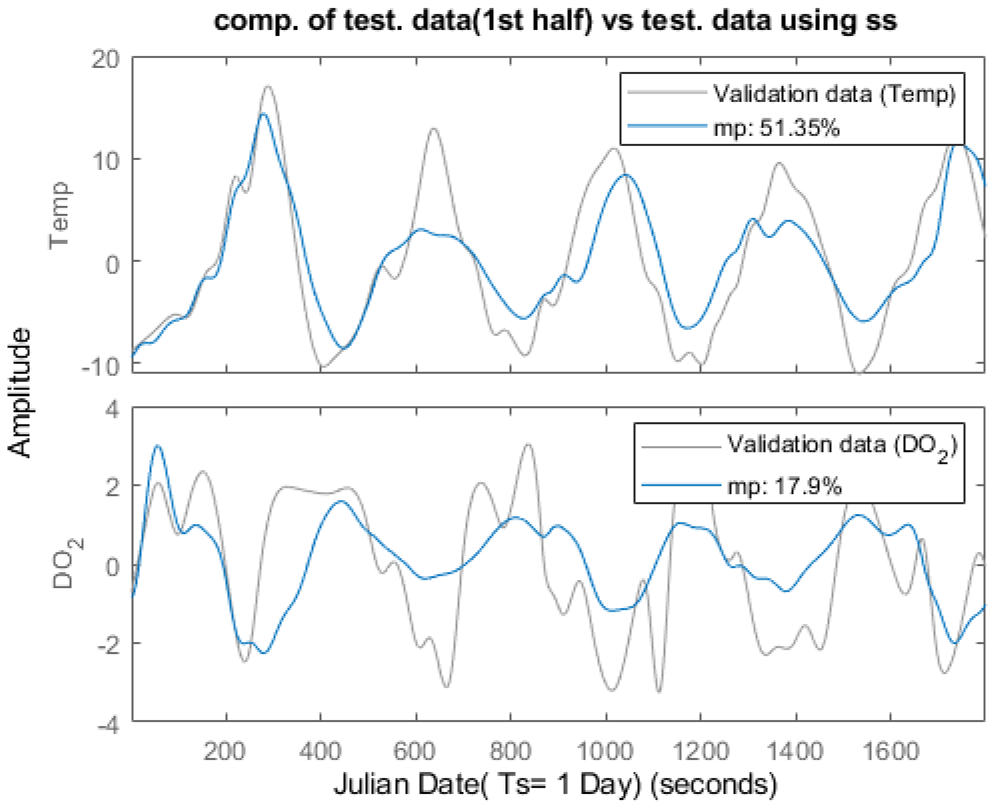
1. *Data Into Three Portions using SSEST*

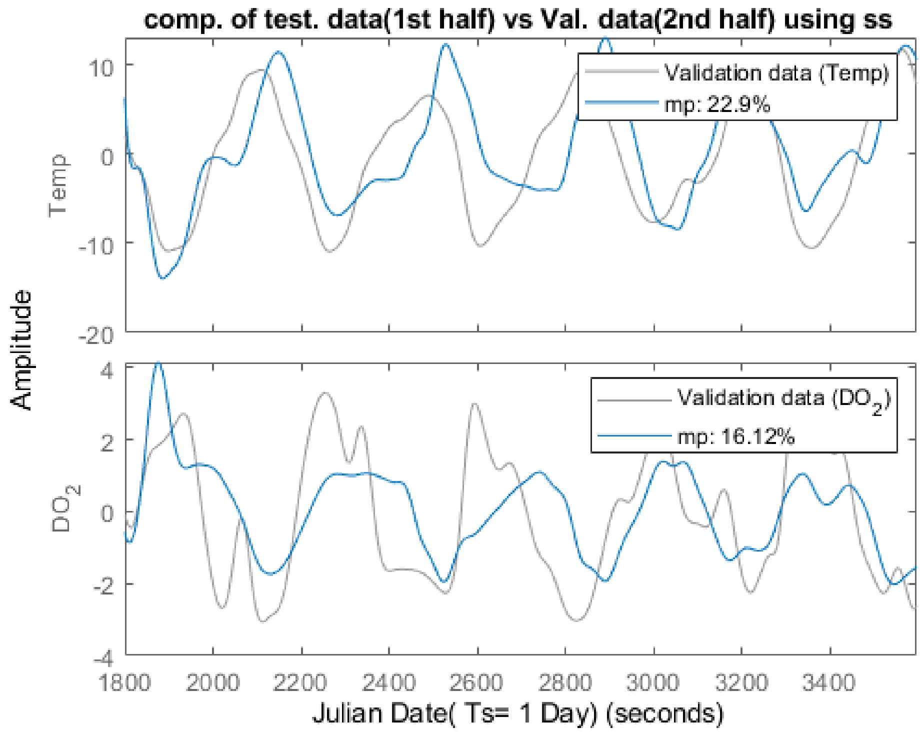
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of daily data using state space for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%SS = State Space
close all;
clc;
clear;
load fichter_daily.mat;%loading fichter dataset
y=[Temp,DO_2];
u=[Ambtemp_time,cms];
y_avg=y-mean(y);%average of the output
Daily=iddata(y_avg,u,Ts); % Providing Inputs and outputs
Daily.InputName={'Ambtemp_time','cms'};%giving names to inputs
Daily.Outputname={'Temp';'DO_2'};%giving names to outputs
daily.timeunit='days';%time unit
mp=ssest(Daily(1:1798))% giving state space structure to 1st half
figure(7)
compare(Daily(1:1798),mp);
title('comp. of test. data(1st half) vs test. data using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure(8)
compare(Daily(1799:3597),mp);
title('comp. of test. data(1st half) vs Val. data(2nd half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure(9)
compare(Daily(3598:5396),mp);
title('comp. of test. data(1st half) vs Val. data(3rd half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
```

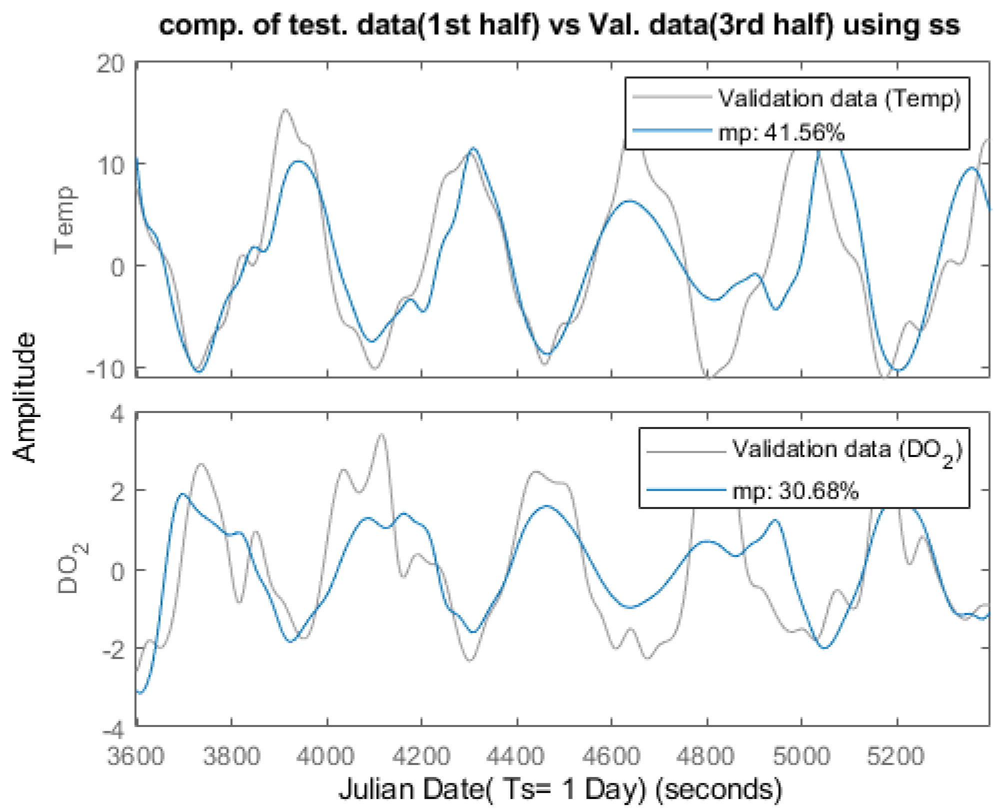
```
mp =
  Continuous-time identified state-space model:
      dx/dt = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A = 
             x1         x2         x3         x4         x5         x6
   x1    0.0658   0.009455    -0.1068  0.0007907   0.008347   -0.01173
   x2    0.0236    0.02631    0.06685    -0.2378    0.01261     0.0302
```

```
   x3    0.03896    0.01365    0.03846    0.01242    -0.1135    0.07159
   x4    0.02792    0.01478     0.0127    0.01776      0.195     0.1179
   x5  -0.007656  -0.004318   0.008206   -0.02033   -0.04057   -0.02373
   x6   -0.01221  -0.002806   -0.04419  -0.005242    -0.0374   -0.07563
   x7    0.02657    0.01776    0.03106  -0.001689   -0.07932     0.0566

              x7
   x1    -0.0311
   x2      0.055
   x3   -0.00217
   x4    0.03287
   x5    0.06741
   x6    -0.1388
   x7    -0.1371

  B =
       Ambtemp_time           cms
   x1       1.9e-08     0.0005627
   x2    -6.622e-09      0.000148
   x3     1.007e-06      0.000878
   x4     -1.54e-07     0.0003845
   x5     1.399e-06     0.0001767
   x6    -2.586e-06    -0.0008459
   x7    -1.575e-06     6.299e-05

  C =
                x1         x2         x3         x4         x5         x6
   Temp     -152.5     -35.11      2.369     0.9818   -0.01582    0.06481
   DO_2      37.06     -44.23    -0.7207     0.7823    0.03296  -0.001735

                x7
   Temp  -0.003526
   DO_2   0.002363

  D =
         Ambtemp_time           cms
   Temp             0             0
   DO_2             0             0

  K =
             Temp       DO_2
   x1  -0.008728    0.02134
   x2   -0.02161    -0.1098
   x3     0.1235     -0.101
   x4    0.06376     0.1789
   x5    -0.2817      1.114
   x6      1.849    -0.2581
   x7     -0.664     0.5228

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 91
```

```
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using SSEST on time domain data.
Fit to estimation data: [100;99.94]% (prediction focus)
FPE: 1.084e-13, MSE: 1.196e-06
```







PublishedwithMATLAB®R2020b

```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of weekly data using state space for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%SS = State Space
close all;
clc;
clear;
load fichter_weekly;% load fichter dataset
y7=[Temp,DO_2];
u7=[Ambtemp_time,cms];
y7_avg=y7-mean(y7);%average of the output
weekly=iddata(y7_avg,u7,Ts); % Providing Inputs and outputs
weekly.InputName={'Ambtemp_time';'cms'};%giving names to inputs
weekly.Outputname={'Temp';'DO_2'};%giving names to ouputs
daily.timeunit='weeks';%time unit
mp=ssest(weekly(1:257))%using ssest into first half
figure(8)
compare(weekly(1:257),mp);
title('comp. of test. data(1st half) vs test. data using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure(9)
compare(weekly(258:514),mp);
title('comp. of test. data(1st half) vs Val. data(2nd half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure(10)
compare(weekly(515:771),mp);
title('comp. of test. data(1st half) vs Val. data(3rd half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
```

```
mp =
  Continuous-time identified state-space model:
      dx/dt = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
                x1          x2          x3          x4          x5          x6
   x1      0.02581   0.0007716    -0.05289    -0.03632     0.01017   -0.003205
   x2     -0.04433    -0.02039    -0.02199     0.01518    -0.01262    -0.01505
   x3      0.03809   0.0007794    -0.01447    0.001269     0.04421   0.0003124
   x4      0.04136   -0.008127       0.005    -0.02139   -0.001289     0.01968
   x5     -0.01911     0.01393    -0.02615   -0.009729     -0.0243   -0.003787
   x6     -0.01362  -0.0004649    -0.04263    -0.03317    0.008639    -0.01975
   x7      0.02025   -0.003109    0.009057     0.06175    -0.02633  -6.672e-05
   x8    -0.004257  -0.0004721    0.002027    -0.02039     0.02494    -0.01344
   x9      0.00041     0.05324     0.06162     -0.0192     0.04608      0.1072
```

x10	0.001824	0.009493	0.02623	0.01034	0.02995	-0.006961

	x7	x8	x9	x10
x1	-0.0171	0.0112	-0.0103	-0.03306
x2	0.0375	-0.03563	0.02299	-0.01663
x3	-0.02591	-0.0324	-0.00303	-0.0315
x4	-0.04432	-0.01122	0.005151	-0.03244
x5	-0.02379	0.01141	-2.862e-06	0.001147
x6	0.05627	0.1041	-0.06392	0.03372
x7	-0.01966	0.03353	-0.02562	0.008284
x8	-0.06064	-0.0266	0.00323	0.03847
x9	-0.08019	0.09922	-0.02827	-0.01271
x10	-0.006535	-0.02939	-0.02209	-0.01574

B =

	Ambtemp_time	cms
x1	-6.979e-05	0.0006973
x2	-1.947e-05	-0.00284
x3	-0.0007252	-0.003588
x4	-0.0001635	0.003849
x5	0.0001913	0.002347
x6	0.001761	0.0005583
x7	2.809e-05	0.002042
x8	-0.0007114	-0.001672
x9	7.757e-06	0.004782
x10	0.0008575	0.0007576

C =

	x1	x2	x3	x4	x5	x6	x7
Temp	19.97	40.28	-5.523	1.466	0.1269	-0.8691	-0.1388
DO_2	12.96	-15.66	-0.2306	-2.34	-0.3951	0.09874	0.4358

	x8	x9	x10
Temp	-0.2488	-0.1302	-0.3906
DO_2	0.1916	0.0345	0.1243

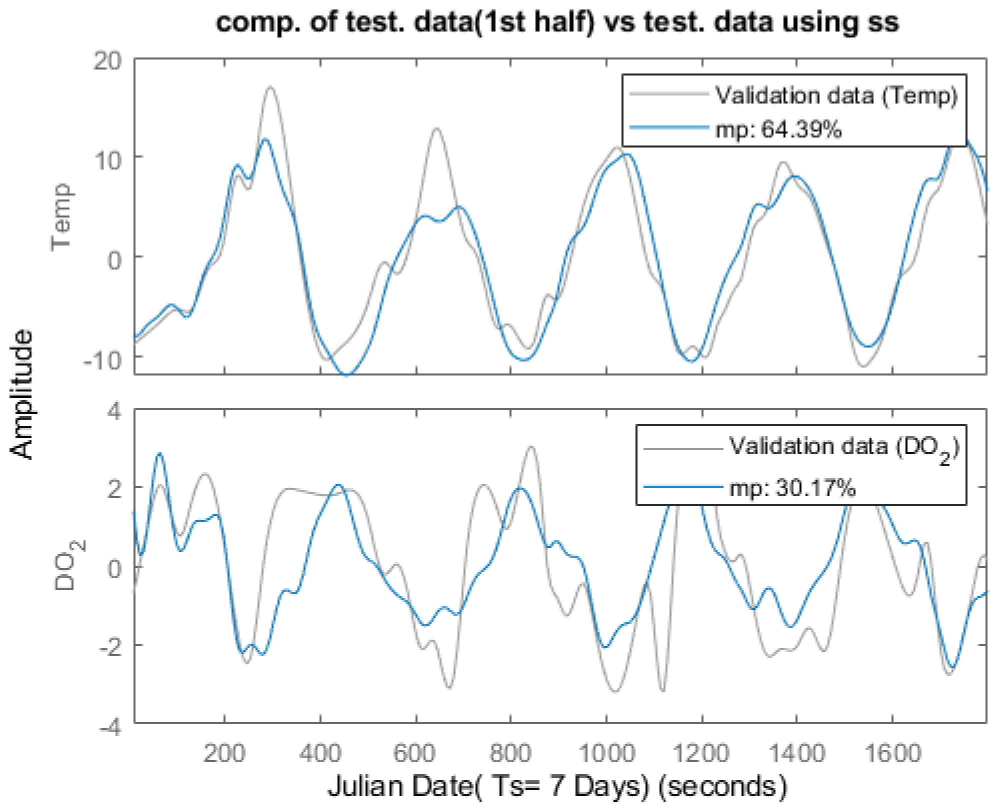
D =

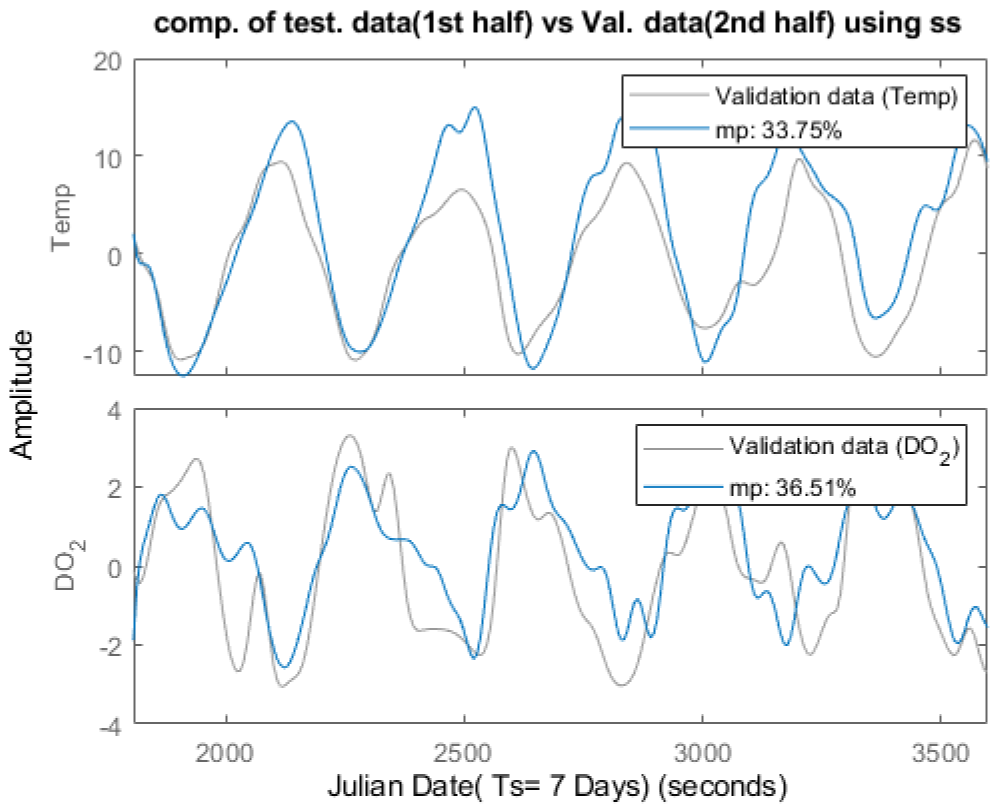
	Ambtemp_time	cms
Temp	0	0
DO_2	0	0

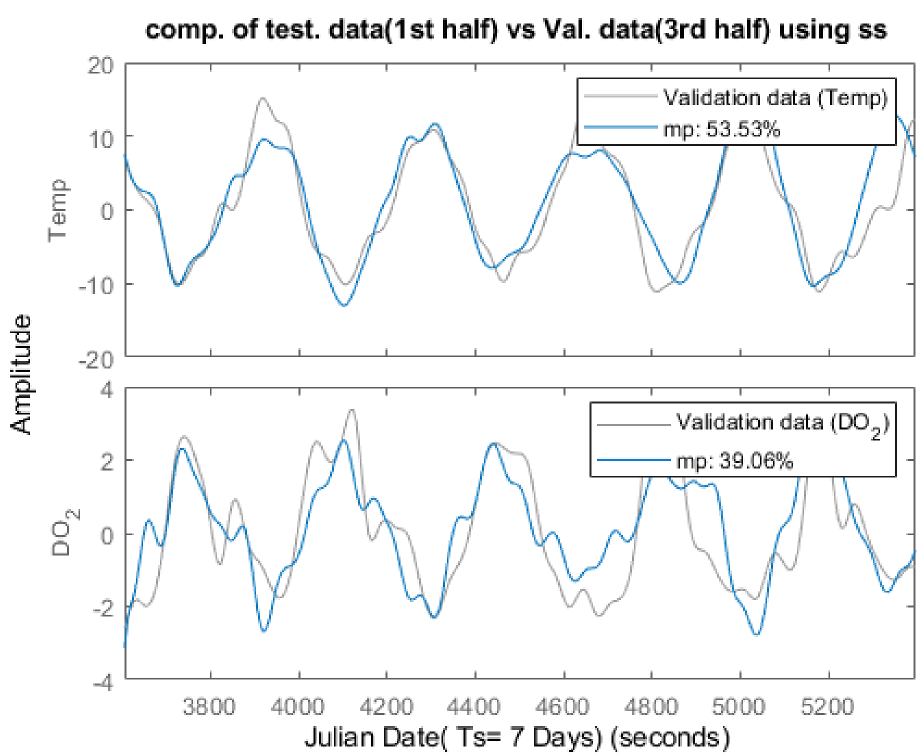
K =

	Temp	DO_2
x1	0.017	-0.002688
x2	-0.001282	-0.005067
x3	0.05442	-0.1227
x4	0.06303	-0.09516
x5	0.1007	-0.01366
x6	-0.2657	0.2586
x7	-0.1081	-0.07075
x8	0.2235	-0.06137
x9	0.385	0.1501
x10	-0.1041	0.1563

```
Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 160
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using SSEST on time domain data.
Fit to estimation data: [98.67;97.57]% (prediction focus)
FPE: 2.587e-05, MSE: 0.01107
```





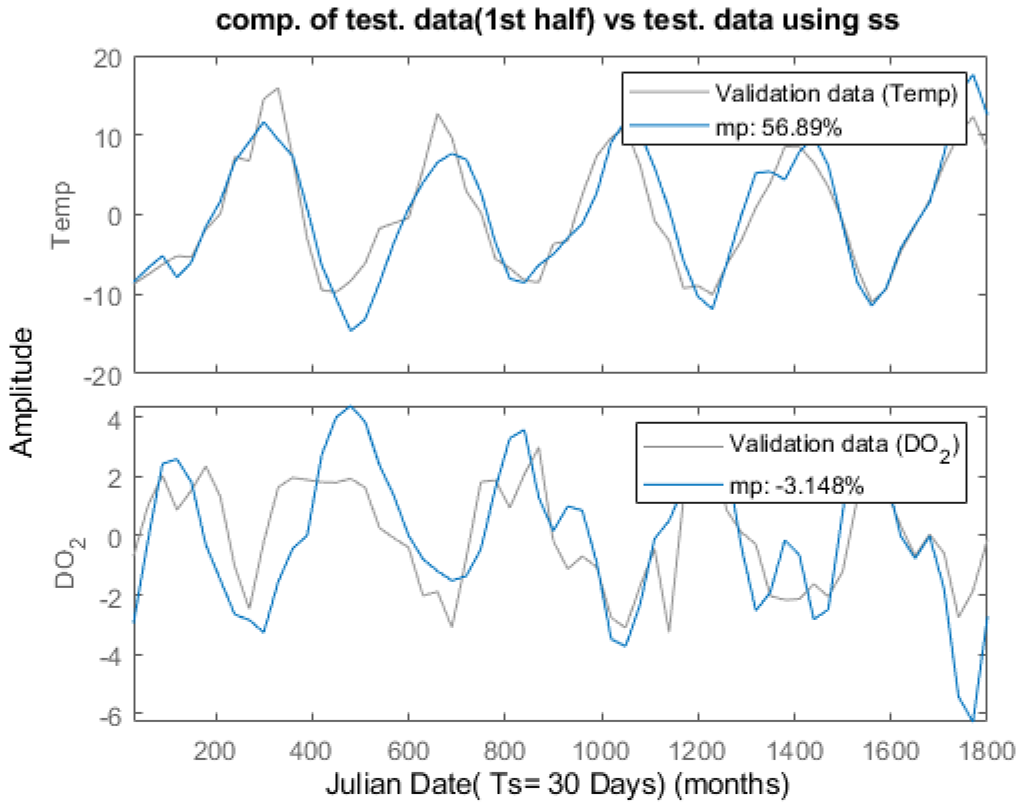


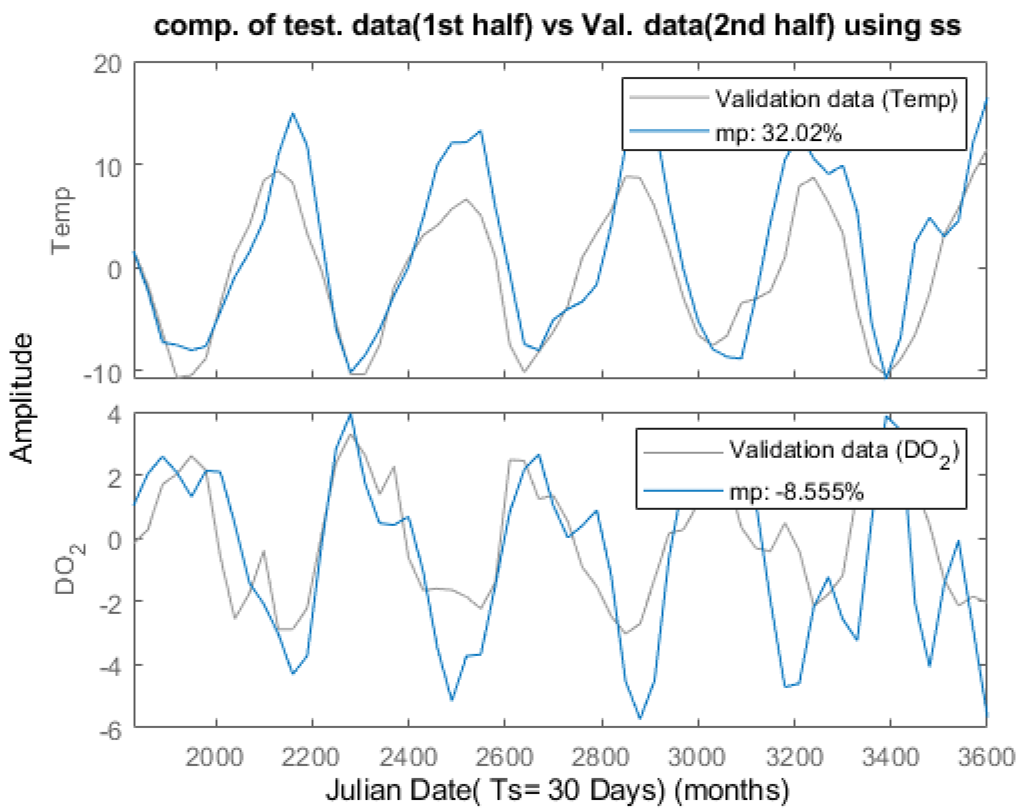
PublishedwithMATLAB®R2020b

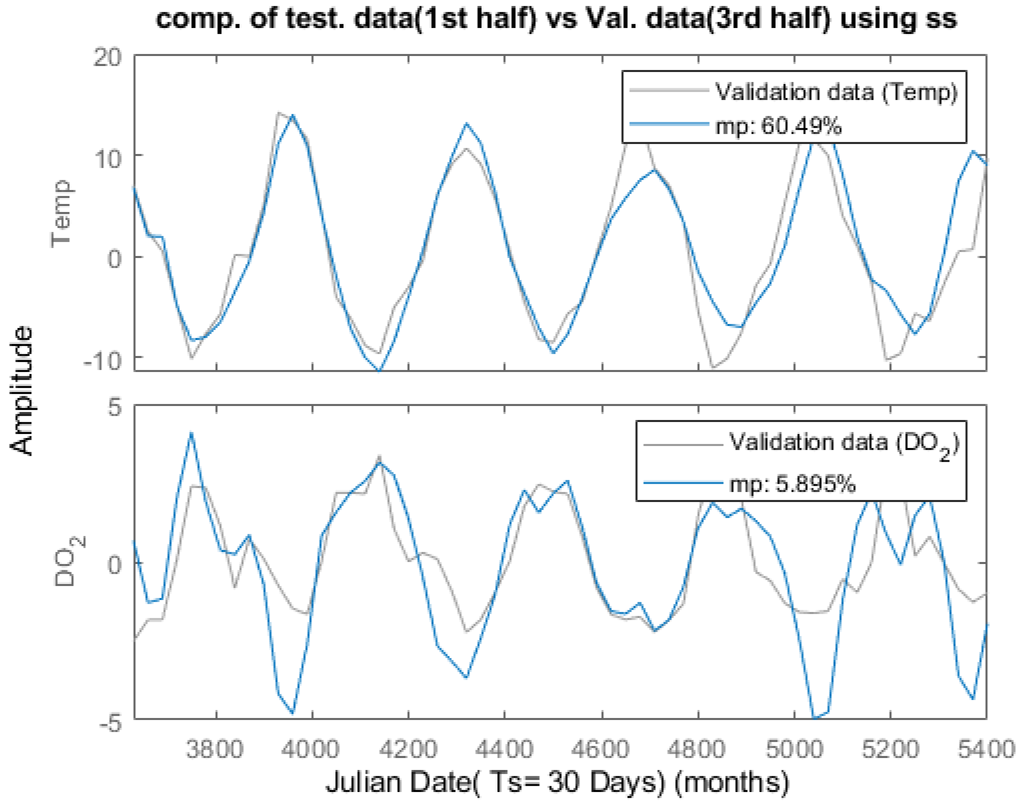
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of monthly data using state space for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%SS = State Space
close all;
clc;
clear;
load fichter_monthly.mat;% load fichter dataset
y30=[Temp,DO_2];
u30=[Ambtemp_time,cms];
y30_avg=y30-mean(y30);%average of the output
monthly=iddata(y30_avg,u30,Ts); % Providing Inputs and outputs
monthly.InputName={'Ambtemp_time';'cms'};%giving names to inputs
monthly.Outputname={'Temp';'DO_2'};%giving names to outputs
monthly.timeunit='months';%time unit
mp=ssest(monthly(1:60))%using ssest to first half
figure(8)
compare(monthly(1:60),mp);
title('comp. of test. data(1st half) vs test. data using ss')
xlabel('Julian Date( Ts= 30 Days)')
figure(9)
compare(monthly(61:120),mp);
title('comp. of test. data(1st half) vs Val. data(2nd half) using ss')
xlabel('Julian Date( Ts= 30 Days)')
figure(10)
compare(monthly(121:180),mp);
title('comp. of test. data(1st half) vs Val. data(3rd half) using ss')
xlabel('Julian Date( Ts= 30 Days)')
```

```
mp =
  Continuous-time identified state-space model:
      dx/dt = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
                x1         x2         x3         x4         x5         x6
   x1  -0.002383    0.01277   -0.02064    0.02313   0.001265    -0.0106
   x2   -0.02878   -0.03802    0.01535    0.04409    -0.1129    0.04106
   x3    0.01846   -0.00833   0.003991  -0.004462    0.02538     0.0256
   x4   0.006058    0.02215  0.0003029   -0.04872     0.1974   -0.03874
   x5    0.01121   -0.00525  0.0003982   -0.02041   -0.05627    0.01658
   x6  0.0007241   -0.01495   -0.04893    0.04529   -0.03761   0.006997

  B =
       Ambtemp_time            cms
```

```
   x1      0.000151     -0.002848
   x2       0.00243      -0.02028
   x3    -0.0003903      0.005407
   x4     -0.003641       0.03172
   x5     0.0002845     -0.003561
   x6     0.0009047      -0.01292

  C =
              x1       x2       x3       x4       x5       x6
   Temp   -2.604   -10.34     1.23   -1.362    4.465   -3.541
   DO_2   -1.335    4.055  -0.5157    2.472    3.366   -1.883

  D =
         Ambtemp_time           cms
   Temp             0             0
   DO_2             0             0

  K =
             Temp        DO_2
   x1   7.392e-05    0.001505
   x2   0.0007754     0.02034
   x3  -0.0004833    -0.00416
   x4   -0.004749    -0.02848
   x5    0.002523    0.007934
   x6    0.002345     0.00908

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 72
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using SSEST on time domain data.
Fit to estimation data: [73.58;42.07]% (prediction focus)
FPE: 20.77, MSE: 4.695
```


comp. of test. data(1st half) vs test. data using ss
Amplitude
Temp
Validation data (Temp)
mp: 56.89%
20
10
0
-10
-20
DO_2
Validation data (DO_2)
mp: -3.148%
4
2
0
-2
-4
-6
200
400
600
800
1000
1200
1400
1600
1800
Julian Date(Ts= 30 Days) (months)


comp. of test. data(1st half) vs Val. data(2nd half) using ss
Amplitude
Temp
Validation data (Temp)
mp: 32.02%
20
10
0
-10
DO_2
Validation data (DO_2)
mp: -8.555%
4
2
0
-2
-4
-6
2000
2200
2400
2600
2800
3000
3200
3400
3600
Julian Date(Ts= 30 Days) (months)



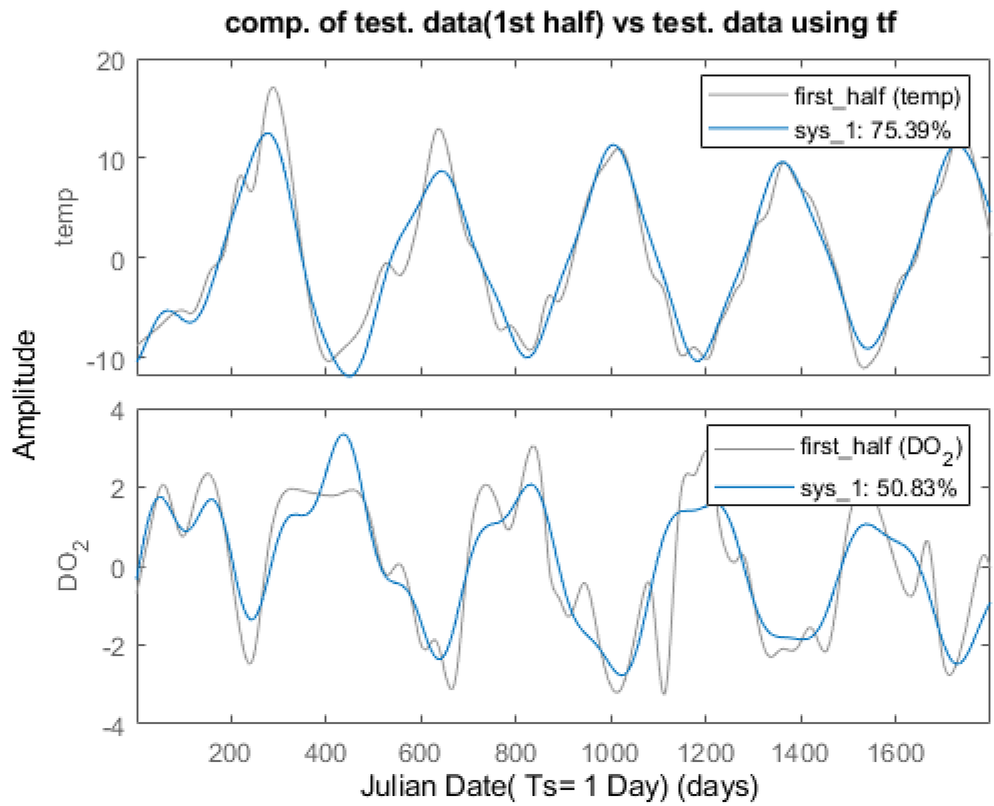
PublishedwithMATLAB®R2020b

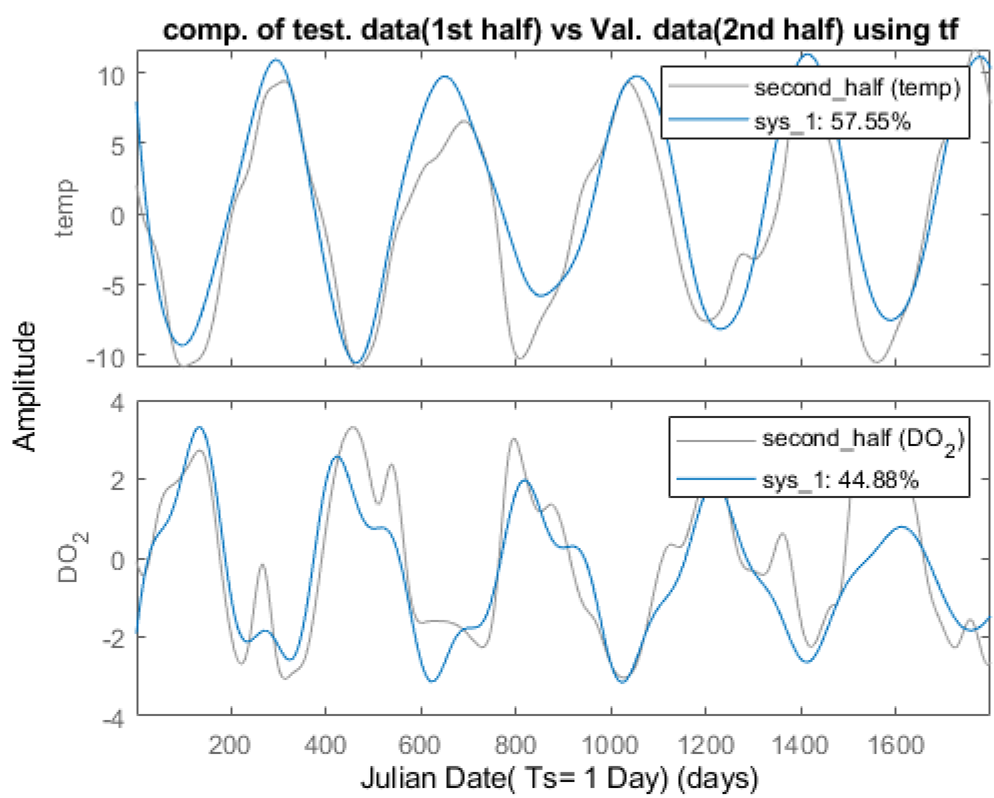
2. Data Into Three Portions Using TFEST

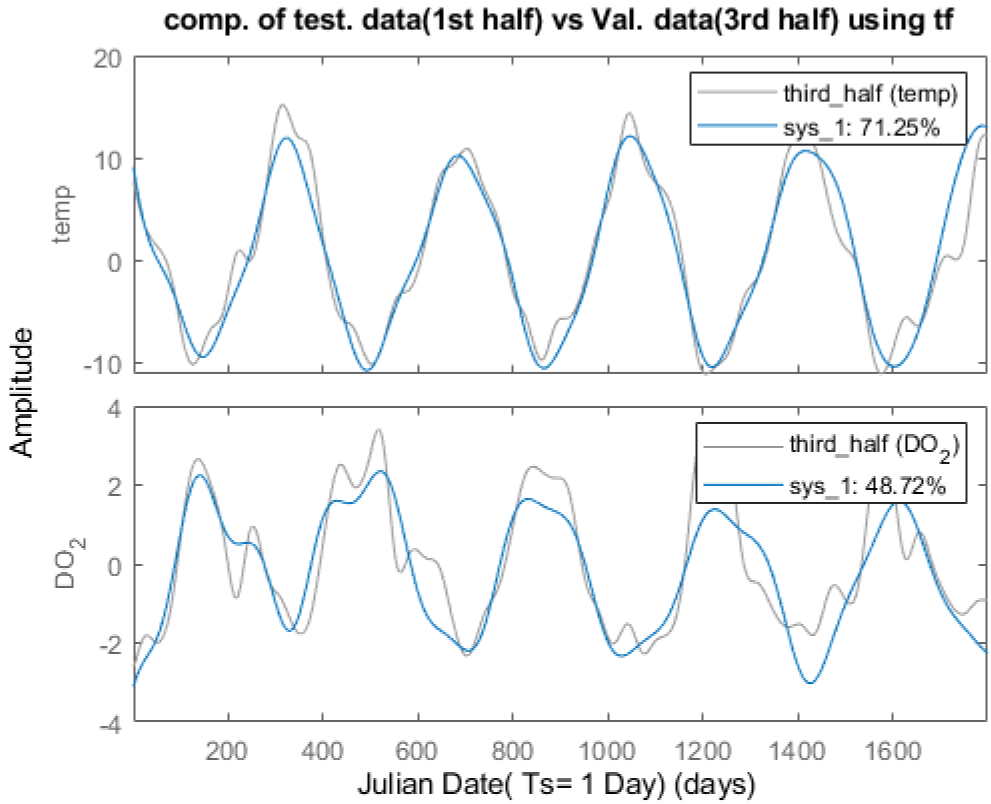
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of Daily data using n4sid for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%tf = transfer function
close all;
clc;
clear;
load fichter_daily;%load fichter daily dataset
% use tfest
y=[Temp,DO_2];%load the output of the system
u=[Ambtemp_time,cms];%load the input of the system
y_avg=y-mean(y);%average of the output
daily=iddata(y_avg,u,Ts);%converting input and output into iddata
%giving names to inputs and outputs
daily.inputname(1)={'Ambtemp'};
daily.inputname(2)={'cms'};
daily.timeunit='days';
daily.outputname(1)={'temp'};
daily.outputname(2)={'DO_2'};
sys=tfest(daily,4,0)
first_half= iddata(y_avg(1:1798,:),u(1:1798,:),Ts);%converting 1st half into iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='days';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y_avg(1799:3597,:),u(1799:3597,:),Ts);%converting 2nd half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='days';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y_avg(3598:5396,:),u(3598:5396,:),Ts);%converting 3rd half into iddata
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='days';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
sys_1=tfest(first_half,4,0);% using tfest to first half
sys_2=tfest(second_half,4,0);%using tfest into 2nd half
sys_3=tfest(third_half,4,0);%using tfest into 3rd half
```

```
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using tf')
xlabel('Julian Date( Ts= 1 Day)')
figure ()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using tf')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) using tf')
xlabel('Julian Date( Ts= 1 Day)')
```

```
sys =

  From input "Ambtemp" to output...
                                  -9.358e-09
   temp:  -----------------------------------------------------------
          s^4 + 0.001761 s^3 + 0.001048 s^2 + 6.402e-07 s + 2.159e-07

                                   2.766e-10
   DO_2:  ------------------------------------------------------------
          s^4 + 0.004818 s^3 + 0.0005125 s^2 + 8.279e-07 s + 5.427e-08

  From input "cms" to output...
                                   2.055e-08
   temp:  ------------------------------------------------------------
          s^4 + 0.009113 s^3 + 0.0004705 s^2 + 2.544e-06 s + 5.287e-08

                                   -3.225e-09
   DO_2:  ------------------------------------------------------------
          s^4 + 0.007126 s^3 + 0.0005625 s^2 + 2.332e-06 s + 7.617e-08

Continuous-time identified transfer function.

Parameterization:
   Number of poles: [4 4;4 4]   Number of zeros: [0 0;0 0]
   Number of free coefficients: 20
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "daily".
Fit to estimation data: [69.67;47.48]%
FPE: 3.562, MSE: 5.406
```


comp. of test. data(1st half) vs test. data using tf
first_half (temp)
sys_1: 75.39%
first_half (DO2)
sys_1: 50.83%
Amplitude
temp
DO2
Julian Date(Ts= 1 Day) (days)


comp. of test. data(1st half) vs Val. data(2nd half) using tf
second_half (temp)
sys_1: 57.55%
second_half (DO2)
sys_1: 44.88%
Amplitude
temp
DO2
Julian Date(Ts= 1 Day) (days)

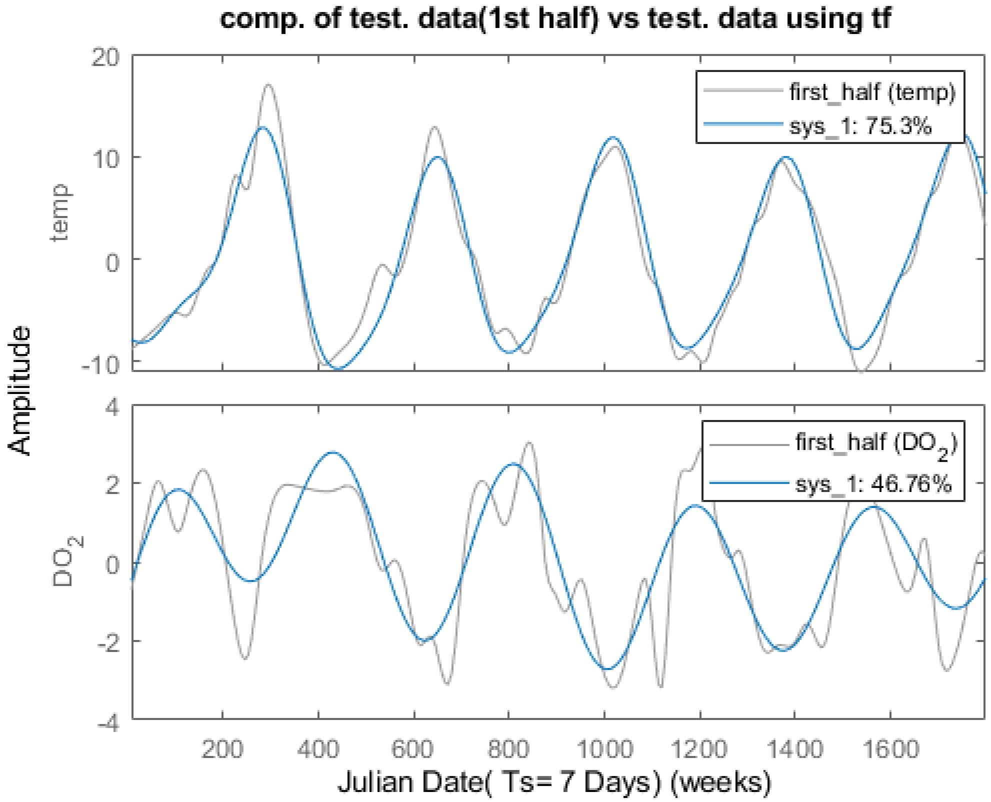


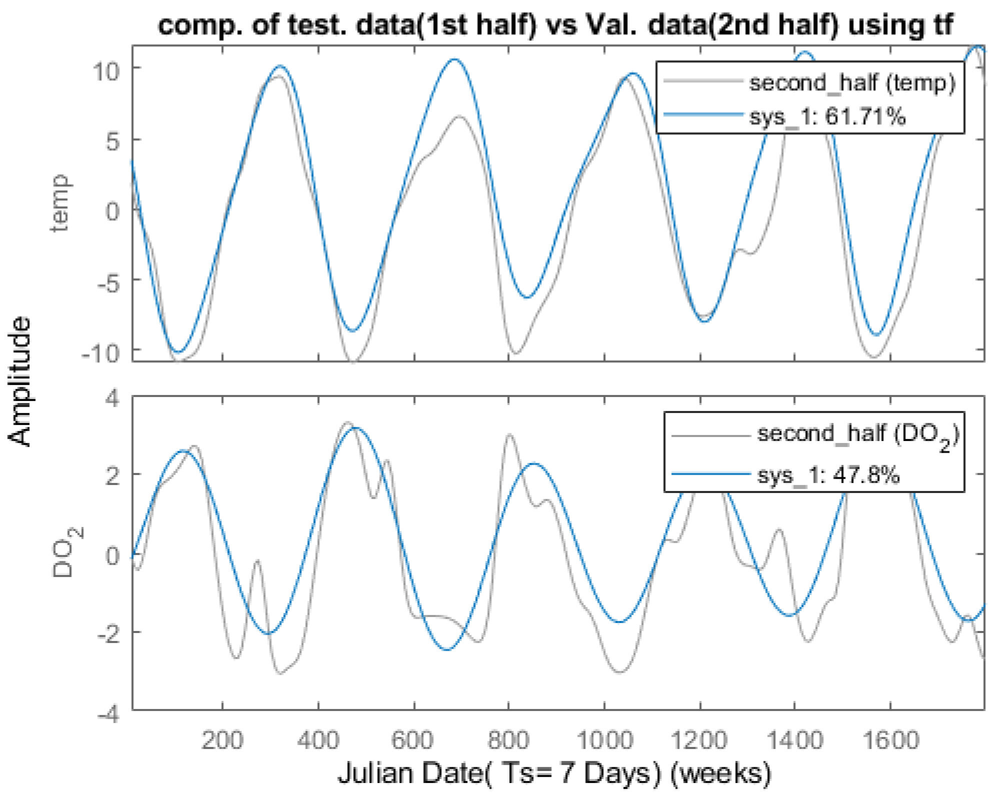
PublishedwithMATLAB®R2020b

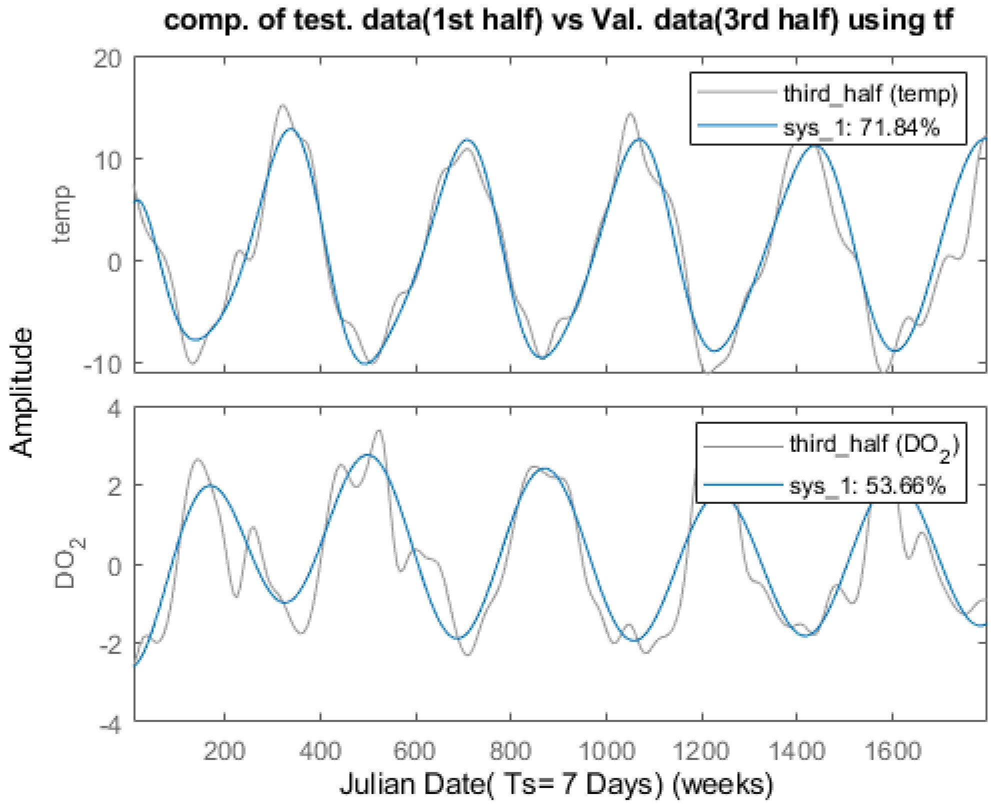
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of weekly data using n4sid for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%tf= transfer function
close all;
clc;
clear;
load fichter_weekly.mat;%load fichter dataset
% use tfest
y7=[Temp,DO_2];%load outputs of the system
u7=[Ambtemp_time,cms];%load inputs of the system
y7_avg=y7-mean(y7);%average of the output
weekly=iddata(y7_avg,u7,Ts);%converting inputs and outputs to iddata
%giving names to outputs and inputs
weekly.inputname(1)={'Ambtemp'};
weekly.inputname(2)={'cms'};
weekly.timeunit='weeks';
weekly.outputname(1)={'temp'};
weekly.outputname(2)={'DO_2'};
sys=tfest((weekly),4,0)
first_half= iddata(y7_avg(1:257,:),u7(1:257,:),Ts);%converting 1st half to iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='weeks';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y7_avg(258:514,:),u7(258:514,:),Ts);% converting 2nd half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='weeks';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y7_avg(515:771,:),u7(515:771,:),Ts);% converting 3rd half into iddata
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='weeks';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
sys_1=tfest((first_half),4,0);%using tfest to first half
sys_2=tfest((second_half),4,0);%using tfest to second half
sys_3=tfest((third_half),4,0);%using tfest to third half
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using tf')
xlabel('Julian Date( Ts= 7 Days)')
figure ()
```

```
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using tf')
xlabel('Julian Date( Ts= 7 Days)')
figure()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) using tf')
xlabel('Julian Date( Ts= 7 Days)')
```

```
sys =

  From input "Ambtemp" to output...
                                    2.168e-10
   temp:  -----------------------------------------------------------
          s^4 + 0.00149 s^3 + 0.0003889 s^2 + 3.747e-07 s + 2.789e-08

                                     6.326e-12
   DO_2:  -------------------------------------------------------------
          s^4 + 0.0006265 s^3 + 0.0002832 s^2 + 1.487e-07 s + 3.185e-09

  From input "cms" to output...
                                    -1.089e-09
   temp:  -------------------------------------------------------------
          s^4 + 0.0002638 s^3 + 0.0003237 s^2 + 7.769e-09 s + 8.666e-09

                                   -1.868e-09
   DO_2:  -----------------------------------------------------------
          s^4 + 0.003162 s^3 + 0.000594 s^2 + 1.056e-06 s + 8.671e-08

Continuous-time identified transfer function.

Parameterization:
   Number of poles: [4 4;4 4]   Number of zeros: [0 0;0 0]
   Number of free coefficients: 20
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "weekly".
Fit to estimation data: [70.58;47.07]%
FPE: 3.794, MSE: 5.147
```


comp. of test. data(1st half) vs test. data using tf
first_half (temp)
sys_1: 75.3%
first_half (DO_2)
sys_1: 46.76%
Amplitude
temp
DO_2
Julian Date(Ts= 7 Days) (weeks)


comp. of test. data(1st half) vs Val. data(2nd half) using tf
second_half (temp)
sys_1: 61.71%
second_half (DO_2)
sys_1: 47.8%
Amplitude
temp
DO_2
Julian Date(Ts= 7 Days) (weeks)

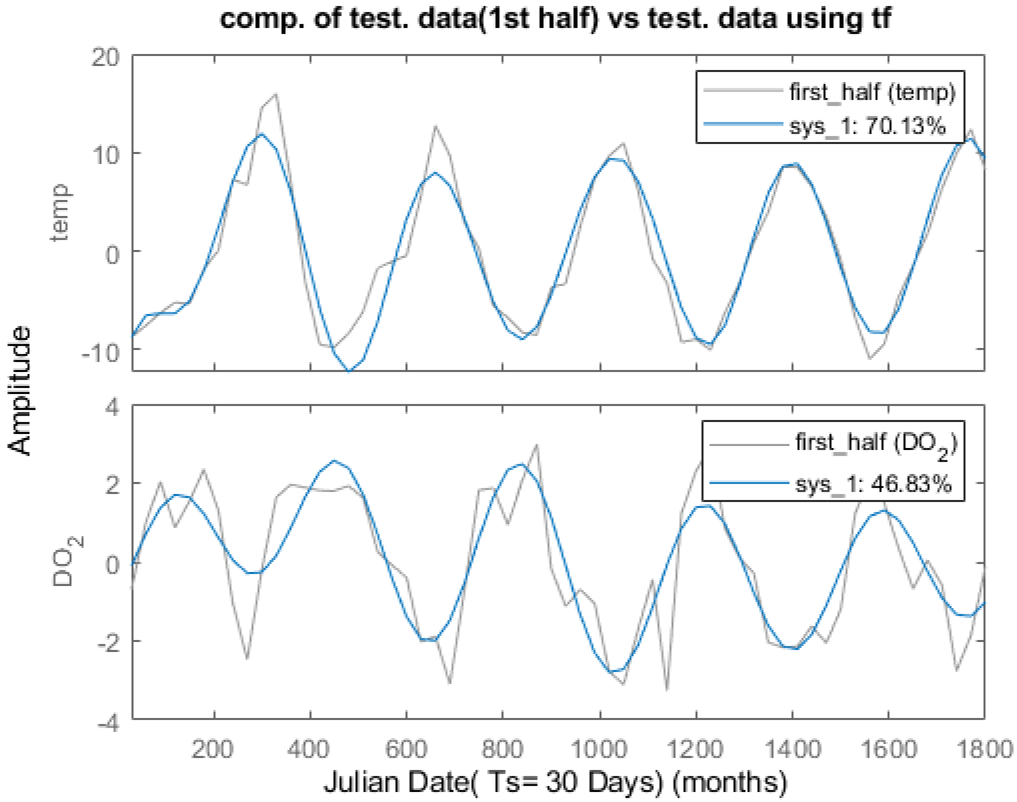


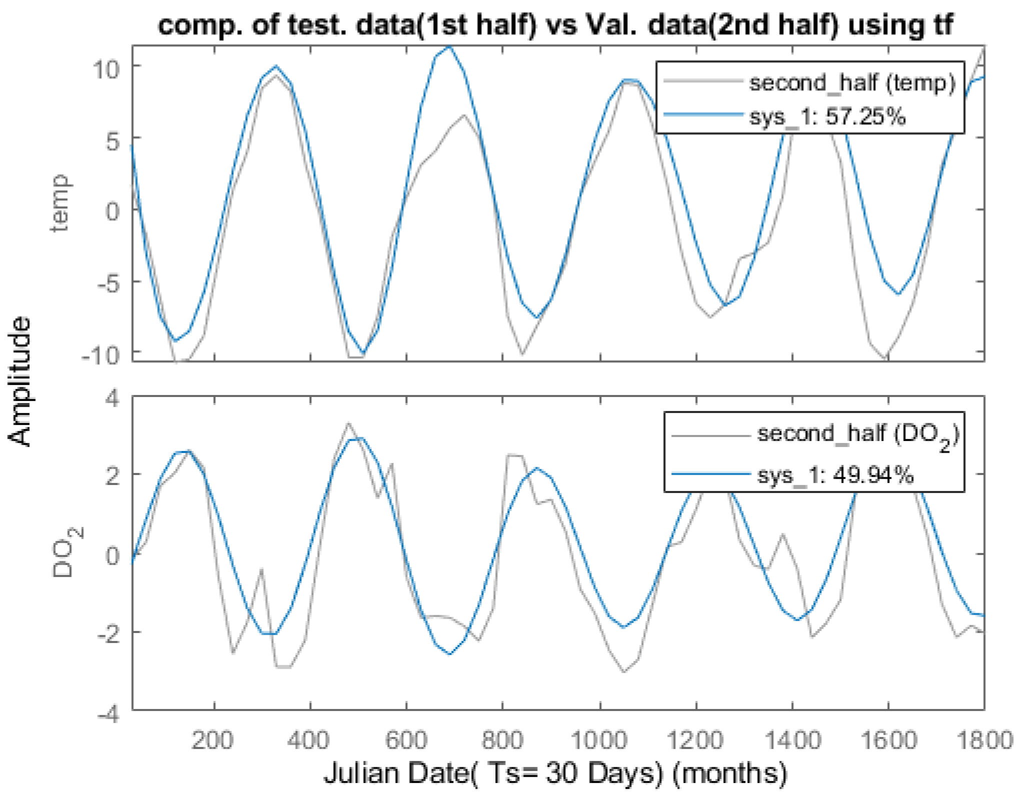
PublishedwithMATLAB®R2020b

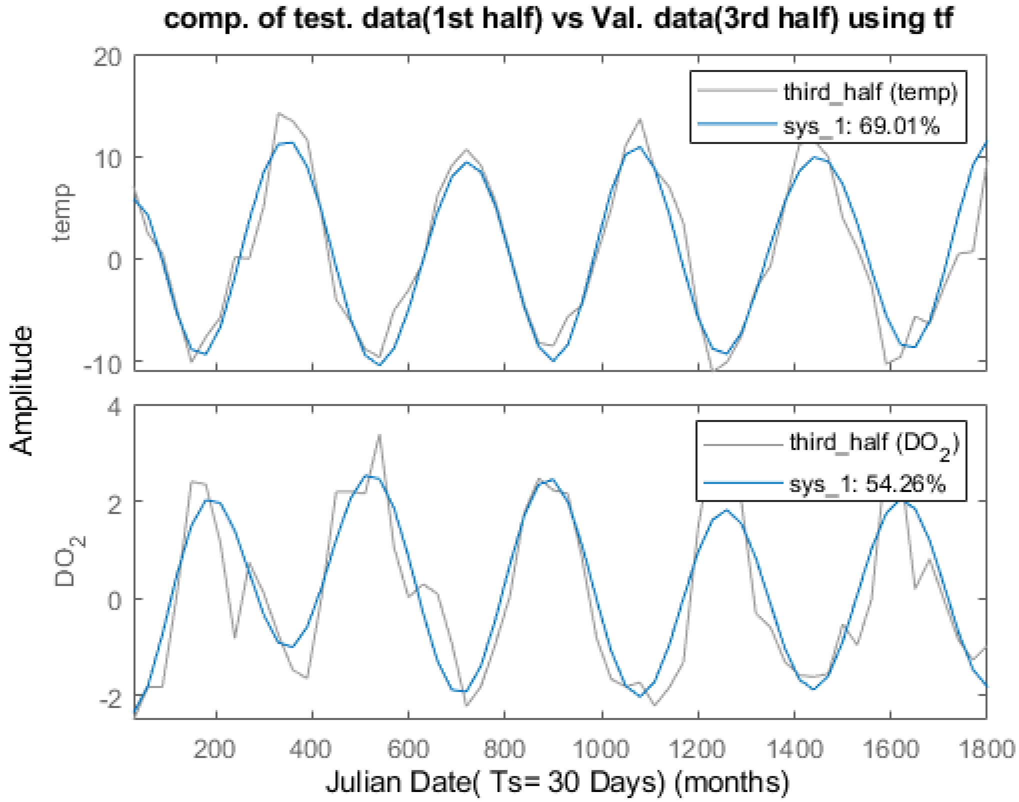
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of monthly data using n4sid for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%tf= transfer function
close all;
clc;
clear;
load fichter_monthly.mat;%load fichter dataset
y30=[Temp,DO_2];%load outputs of the system
u30=[Ambtemp_time,cms];%load inputs of the system
y30_avg=y30-mean(y30);%average of the outputs
monthly=iddata(y30_avg,u30,Ts);%converting inputs and outputs into iddata
%giving names to inputs and outputs
monthly.inputname(1)={'Ambtemp'};
monthly.inputname(2)={'cms'};
monthly.timeunit='months';
monthly.outputname(1)={'temp'};
monthly.outputname(2)={'DO_2'};
sys=tfest(monthly,4,0)
first_half= iddata(y30_avg(1:60,:),u30(1:60,:),Ts);%convering 1st half into idata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='months';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y30_avg(61:120,:),u30(61:120,:),Ts);%converting 2nd half into idata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='months';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y30_avg(121:180,:),u30(121:180,:),Ts);%converting 3rd half into iddata
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='months';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
sys_1=tfest(first_half,4,0);%using tfest to first half
sys_2=tfest(second_half,4,0);%using tfest to second half
sys_3=tfest(third_half,4,0);%converting tfest to third half
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using tf')
xlabel('Julian Date( Ts= 30 Days)')
figure ()
compare(sys_1,second_half)
```

```
title('comp. of test. data(1st half) vs Val. data(2nd half) using tf')
xlabel('Julian Date( Ts= 30 Days)')
figure ()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) using tf')
xlabel('Julian Date( Ts= 30 Days)')
```

```
sys =

  From input "Ambtemp" to output...
                              -6.243e-10
   temp:  ------------------------------------------------------------
          s^4 + 0.005828 s^3 + 0.0003877 s^2 + 1.693e-06 s + 2.422e-08

                              1.083e-13
   DO_2:  ------------------------------------------------------------
          s^4 + 0.001252 s^3 + 1.231e-05 s^2 + 1.141e-08 s + 1.683e-11

  From input "cms" to output...
                              3.878e-08
   temp:  -----------------------------------------------------------
          s^4 + 0.03811 s^3 + 0.0009106 s^2 + 1.205e-05 s + 1.749e-07

                              -1.72e-12
   DO_2:  ------------------------------------------------------------
          s^4 + 0.000228 s^3 + 1.274e-05 s^2 + 4.652e-10 s + 2.183e-11

Continuous-time identified transfer function.

Parameterization:
   Number of poles: [4 4;4 4]   Number of zeros: [0 0;0 0]
   Number of free coefficients: 20
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "monthly".
Fit to estimation data: [70.69;1.814]%
FPE: 18.06, MSE: 7.154
```


comp. of test. data(1st half) vs test. data using tf
first_half (temp)
sys_1: 70.13%
first_half (DO2)
sys_1: 46.83%
Amplitude
temp
DO2
Julian Date(Ts= 30 Days) (months)


comp. of test. data(1st half) vs Val. data(2nd half) using tf
second_half (temp)
sys_1: 57.25%
second_half (DO2)
sys_1: 49.94%
Amplitude
temp
DO2
Julian Date(Ts= 30 Days) (months)



PublishedwithMATLAB®R2020b

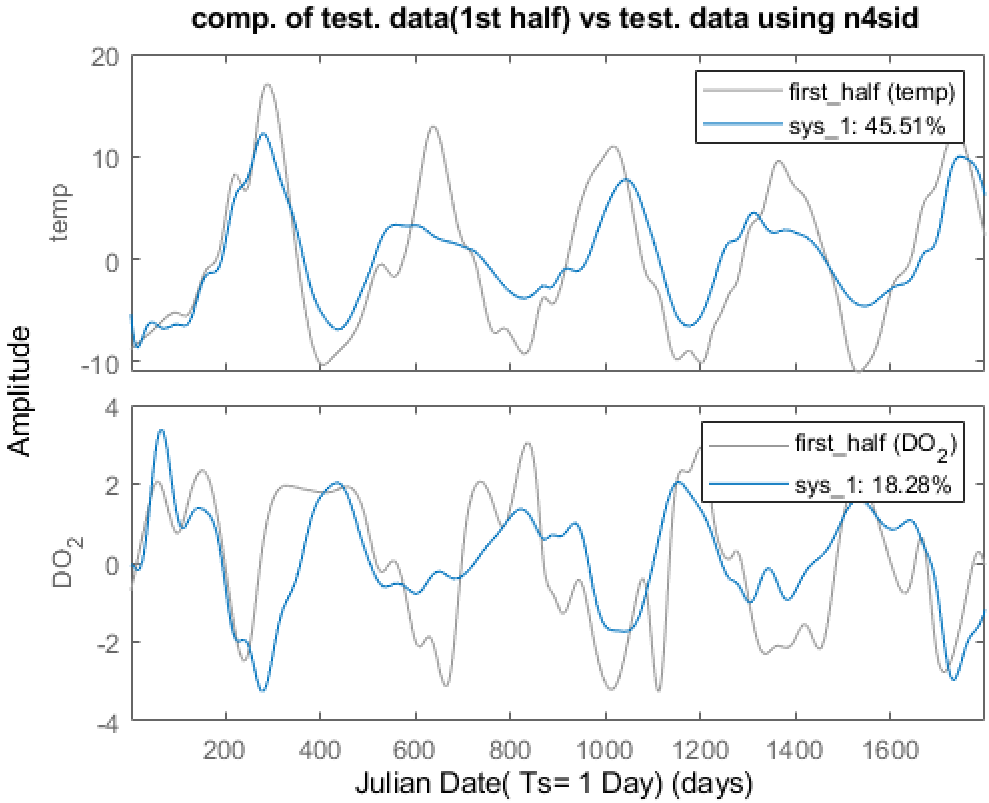
3. Data into three Portions Using TFEST

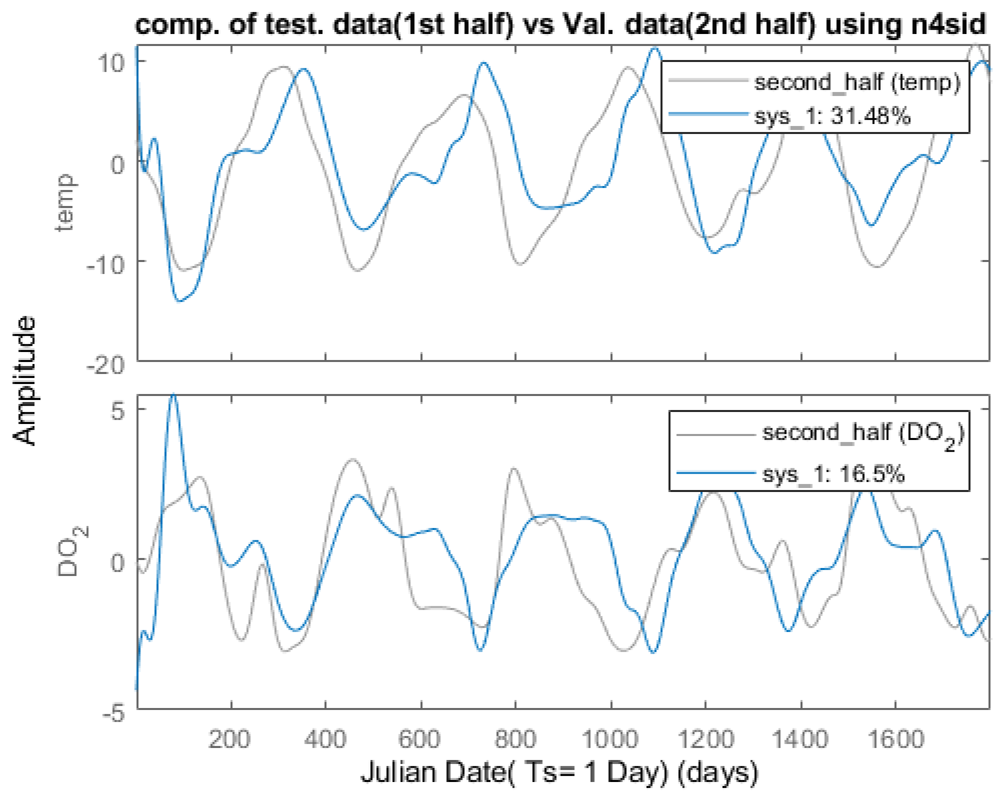
```
Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of daily data using n4sid for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%SS = State Space
close all;
clc;
clear;
load fichter_daily.mat;% load the fichter daily dataset
y=[Temp,DO_2];%load the output of the system
u=[Ambtemp_time,cms];%load the input of the system
y_avg=y-mean(y);%average of the outputs
daily=iddata(y_avg,u,Ts);%converting input and output into iddata
%giving names to inputs and outputs
daily.inputname(1)={'Ambtemp'};
daily.inputname(2)={'cms'};
daily.timeunit='days';
daily.outputname(1)={'temp'};
daily.outputname(2)={'DO_2'};
sys=n4sid(daily,4)
first_half= iddata(y_avg(1:1798,:),u(1:1798,:),Ts);%converting first half into ID data
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='days';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y_avg(1799:3597,:),u(1799:3597,:),Ts);%converting 2nd half into ID data
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='days';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y_avg(3598:5396,:),u(3598:5396,:),Ts);%converting 3rd half into id data
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='days';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
sys_1=n4sid(first_half);%using n4sid to first half
sys_2=n4sid(second_half);%using n4sid to 2nd half
sys_3=n4sid(third_half);%using n4sid to 3rd half
```

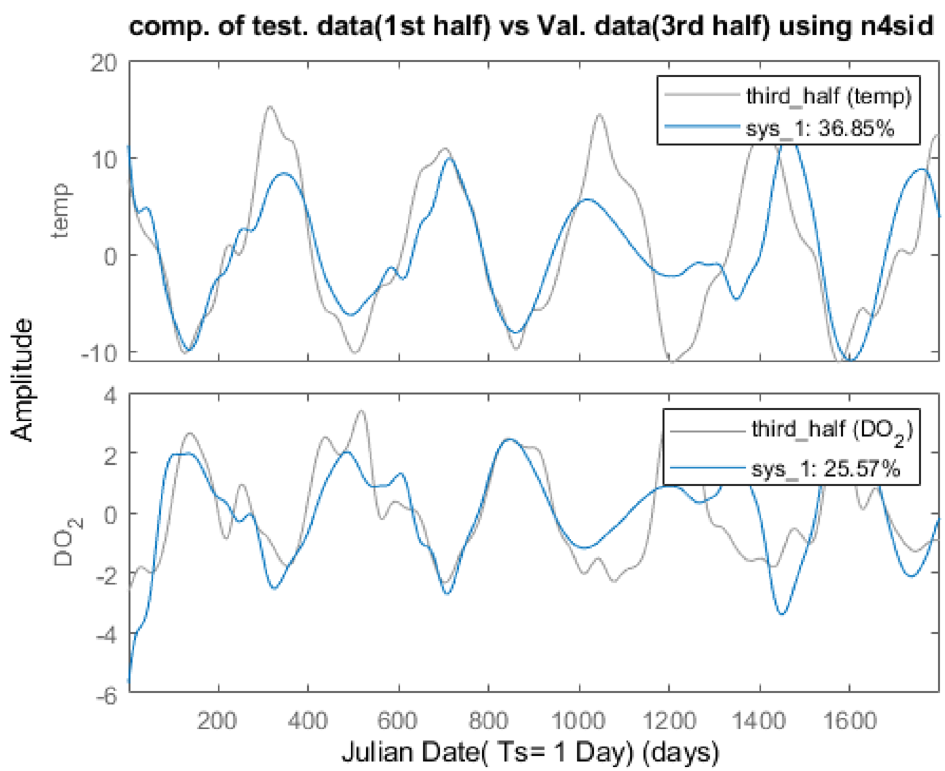
```
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using n4sid')
xlabel('Julian Date( Ts= 1 Day)')
figure ()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using n4sid')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) using n4sid')
xlabel('Julian Date( Ts= 1 Day)')
```

```
sys =
  Discrete-time identified state-space model:
    x(t+Ts) = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
               x1          x2          x3          x4
   x1       1.005  -0.0003073     0.03256   -0.001677
   x2   -0.001541      0.9992   -0.003861    -0.04135
   x3    -0.01923   0.0009281      0.9898    0.003297
   x4     0.00214     0.03995   -0.002879      0.9895

  B =
          Ambtemp         cms
   x1   2.963e-07   -2.52e-05
   x2   -1.24e-07   2.104e-05
   x3   3.203e-06  -2.361e-05
   x4   3.819e-06  -2.833e-05

  C =
             x1      x2      x3      x4
   temp   282.6   -47.7   4.665  0.7398
   DO_2  -76.99  -60.08  -1.129     1.3

  D =
         Ambtemp      cms
   temp        0        0
   DO_2        0        0

  K =
             temp       DO_2
   x1    0.001871  -0.001537
   x2   -0.002386  -0.008774
   x3     0.01589  -0.009345
   x4     0.01364    0.05735

Sample time: 1 days

Parameterization:
```

```
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 40
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using N4SID on time domain data "daily".
Fit to estimation data: [99.55;99.24]% (prediction focus)
FPE: 1.589e-07, MSE: 0.001171
```







PublishedwithMATLAB®R2020b

```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of weekly data using n4sid for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%SS = State Space
close all;
clc;
clear;
load fichter_weekly.mat;% load the fichter daily dataset
% use n4sid
y7=[Temp,DO_2];%load the output of the system
u7=[Ambtemp_time,cms];%load the input of the system
y7_avg=y7-mean(y7);%average of the output
weekly=iddata(y7_avg,u7,Ts);%converting input and output to iddata
%giving names to input and output
weekly.inputname(1)={'Ambtemp'};
weekly.inputname(2)={'cms'};
weekly.timeunit='weeks';
weekly.outputname(1)={'temp'};
weekly.outputname(2)={'DO_2'};
sys=n4sid((weekly),4)
first_half= iddata(y7_avg(1:257,:),u7(1:257,:),Ts);%converting 1st into id data
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='weeks';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y7_avg(258:514,:),u7(258:514,:),Ts);%converting 2nd into id data
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='weeks';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y7_avg(515:771,:),u7(515:771,:),Ts)%converting 3rd half into id data
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='weeks';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
sys_1=n4sid(first_half);
sys_2=n4sid(second_half);
sys_3=n4sid(third_half);
sys_1=n4sid((first_half));
sys_2=n4sid((second_half));
sys_3=n4sid((third_half));
figure()
compare(sys_1,first_half)
```

```
title('comp. of test. data(1st half) vs test. data using n4sid')
xlabel('Julian Date( Ts= 7 Days)')
figure ()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using n4sid')
xlabel('Julian Date( Ts= 7 Days)')
figure()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half)using n4sid')
xlabel('Julian Date( Ts= 7 Days)')
```

```
Error using evalin
Unrecognized function or variable 'n4sid_weekly'.
```

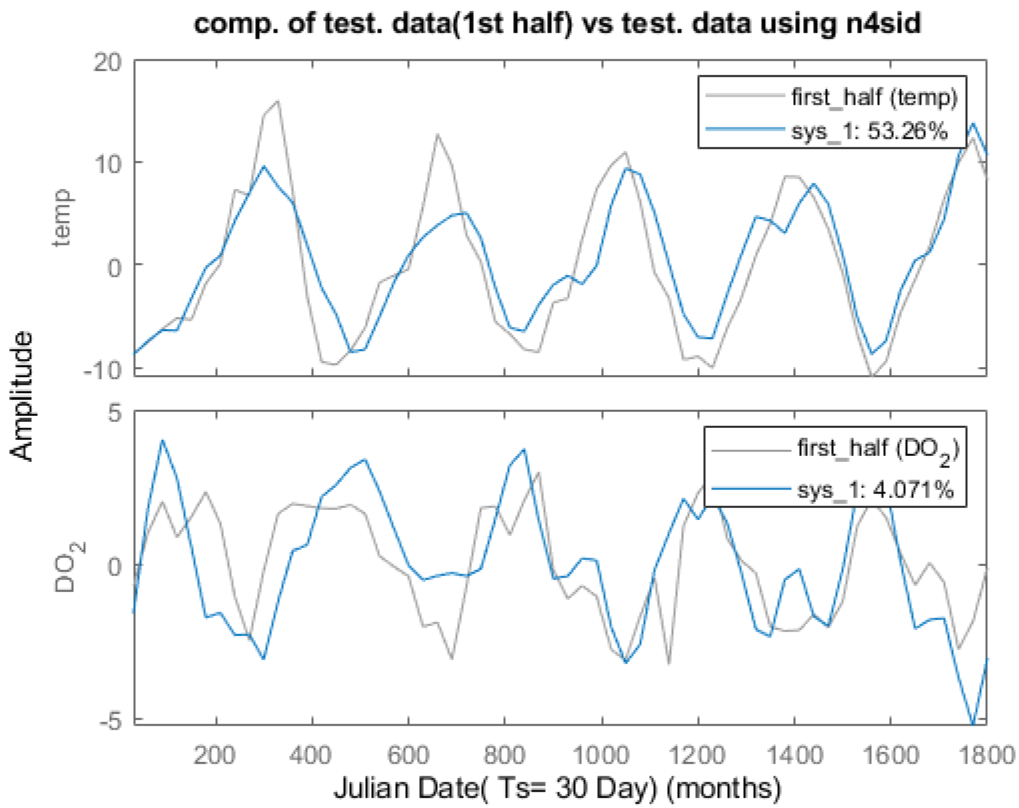
PublishedwithMATLAB®R2020b

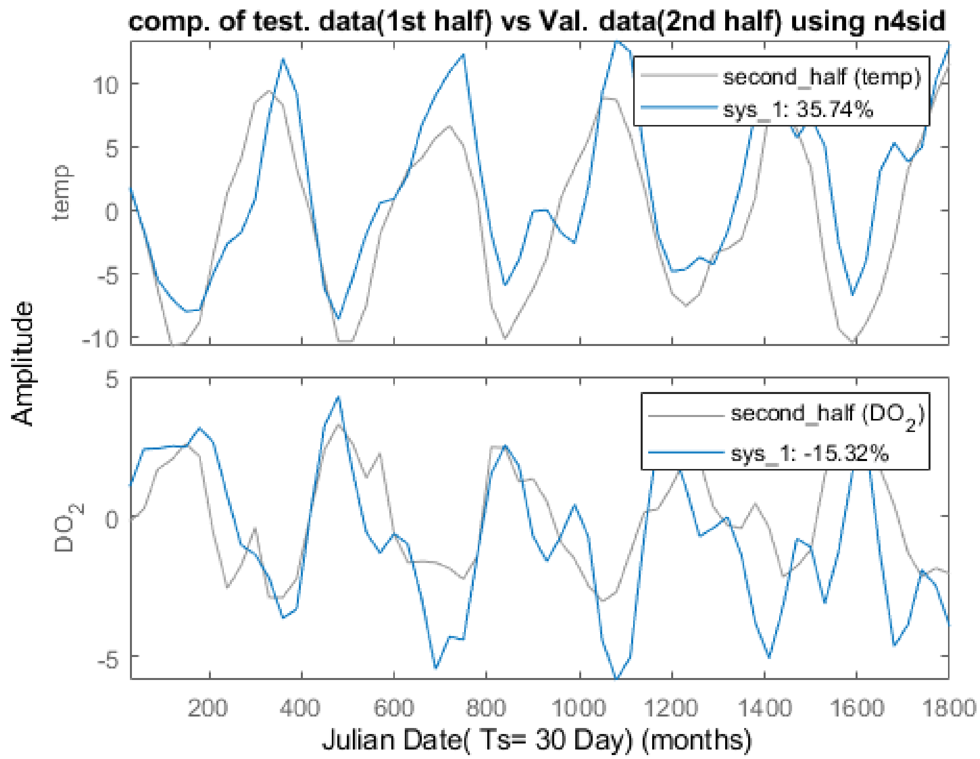
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset. It
%will compare the result on the basis of Monthly data using n4sid for
%the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%SS = State Space
close all;
clc;
clear;
load fichter_monthly.mat;%load the fichter dataset
y30=[Temp,DO_2];% load the output of the system
u30=[Ambtemp_time,cms];%load the input of the system
y30_avg=y30-mean(y30);%average of the output
monthly=iddata(y30_avg,u30,Ts);%converting input and output into id data
%giving names to inputs and outputs
monthly.inputname(1)={'Ambtemp'};
monthly.inputname(2)={'cms'};
monthly.timeunit='months';
monthly.outputname(1)={'temp'};
monthly.outputname(2)={'DO_2'};
sys=n4sid(monthly,4)
first_half= iddata(y30_avg(1:60,:),u30(1:60,:),Ts);% converting 1st half into id data
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='months';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y30_avg(61:120,:),u30(61:120,:),Ts);%converting 2nd half into id data
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='months';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y30_avg(121:180,:),u30(121:180,:),Ts);%converting 3rd half into id data
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='months';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
sys_1=n4sid(first_half);
sys_2=n4sid(second_half);
sys_3=n4sid(third_half);
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using n4sid')
xlabel('Julian Date( Ts= 30 Day)')
figure ()
compare(sys_1,second_half)
```

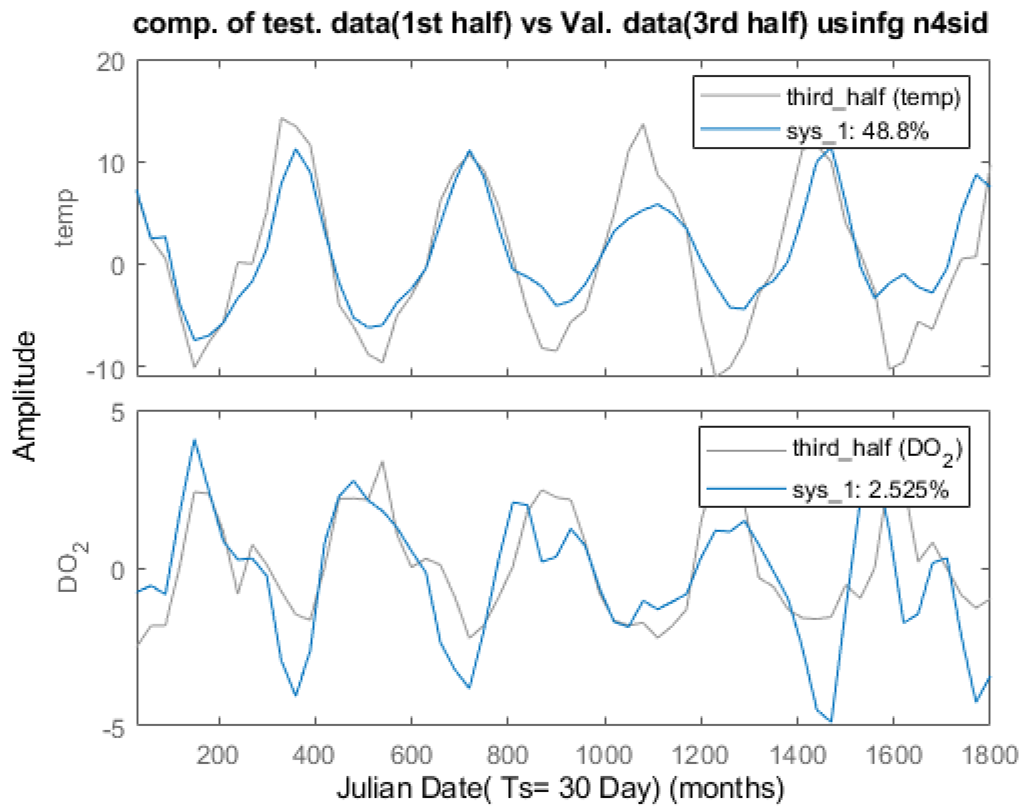
```
title('comp. of test. data(1st half) vs Val. data(2nd half) using n4sid')
xlabel('Julian Date( Ts= 30 Day)')
figure ()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) usinfg n4sid')
xlabel('Julian Date( Ts= 30 Day)')
```

```
sys =
  Discrete-time identified state-space model:
    x(t+Ts) = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
             x1         x2         x3         x4
   x1    0.8162    -0.6177    -0.2593    0.05737
   x2    0.3082     0.7465    -0.4136   -0.05596
   x3   -0.2027    0.06001     0.3336    -0.6014
   x4  -0.06819    0.05061     0.1426     0.8351

  B =
          Ambtemp          cms
   x1  -0.0004729     0.007296
   x2    0.000317     0.005524
   x3   0.0002953     0.008062
   x4  -0.0002455    -0.000906

  C =
              x1        x2        x3        x4
   temp    -32.52     -1.37    -6.882     3.832
   DO_2     7.363   -0.1929     1.761    -7.467

  D =
         Ambtemp       cms
   temp        0         0
   DO_2        0         0

  K =
             temp        DO_2
   x1    -0.02293   -0.006857
   x2   -0.007012   -0.004436
   x3    -0.00534      0.0084
   x4   -0.009726    -0.03891

Sample time: 30 months

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 40
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.
```

```
Status:
Estimated using N4SID on time domain data "monthly".
Fit to estimation data: [68.75;41.04]% (prediction focus)
FPE: 6.367, MSE: 5.895
```







PublishedwithMATLAB®R2020b

Appendix 3. System Identification Excluding Three Quarters of Data

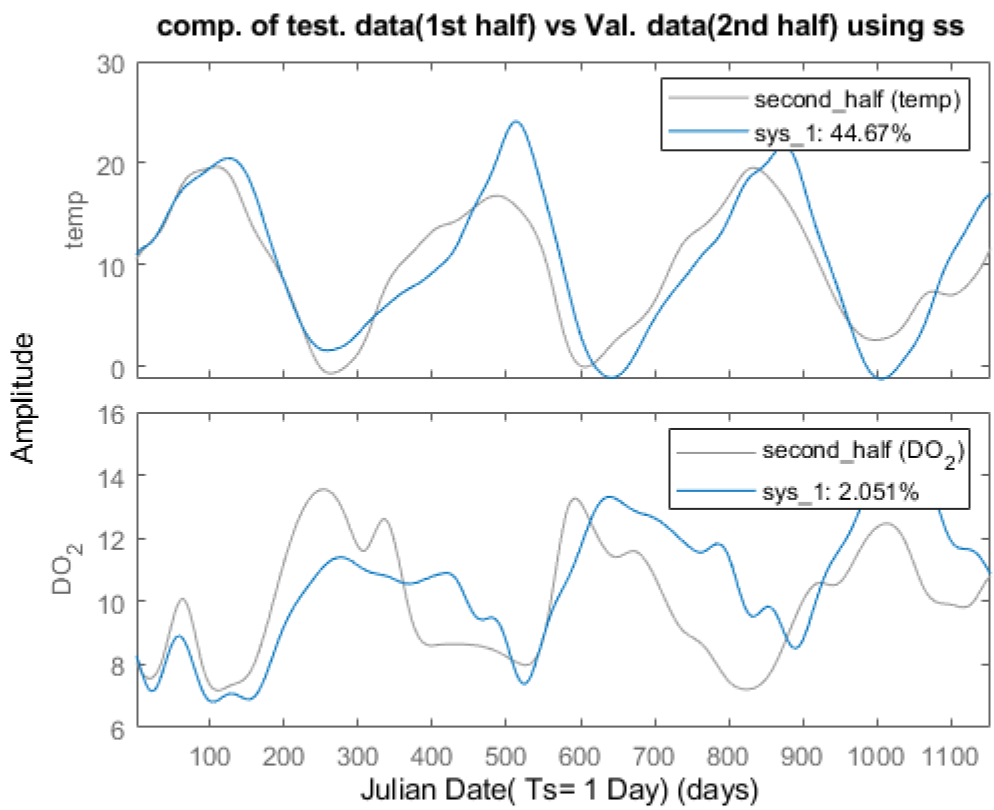
1. *Data Into Four Quarters Using N4SID*

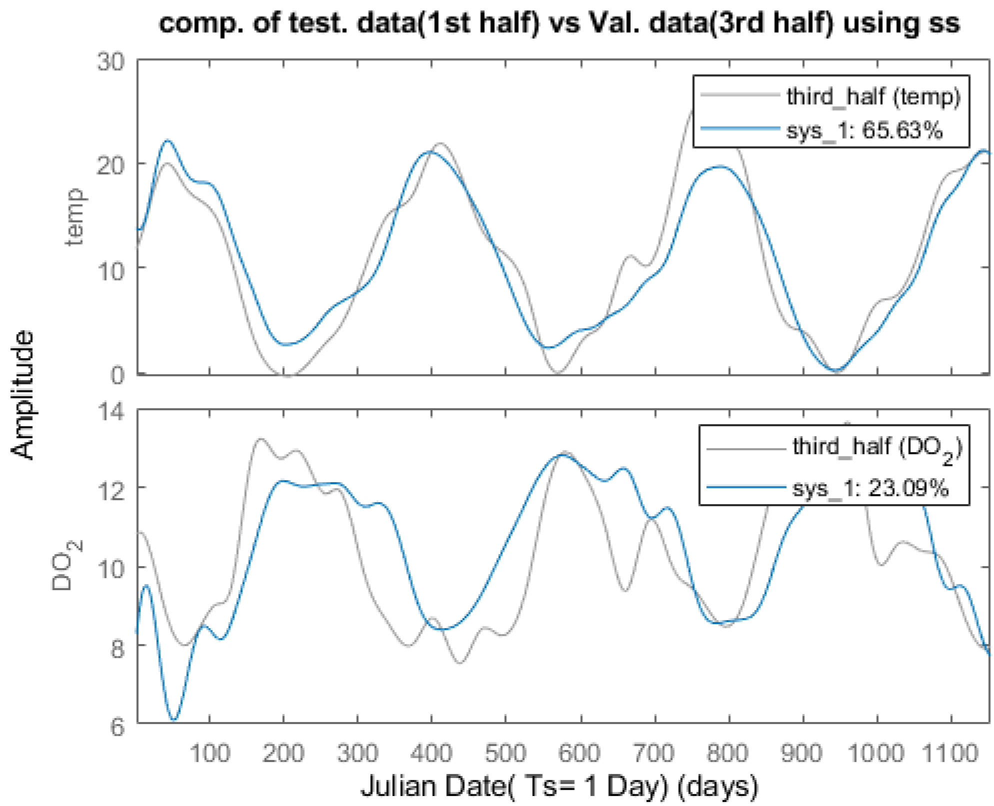
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset.
%It will compare the result on the basis of daily data excluding three years using state space
%for the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%ss. = state space
close all;
clc;
clear;
load fichter_excluding_three_years_daily.mat;%load fichter excluding three years
y=[Temp,DO_2];%load outputs of the system
u=[Ambtemp_time,cms];%load input of the system
daily=iddata(y,u,Ts);% converting inputs and outputs into iddata
%giving names to the inputs and outputs
daily.inputname(1)={'Ambtemp'};
daily.inputname(2)={'cms'};
daily.timeunit='days';
daily.outputname(1)={'temp'};
daily.outputname(2)={'DO_2'};
sys=n4sid(daily,4)
first_half= iddata(y(1:1153,:),u(1:1153,:),Ts);%converting 1st half into iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='days';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y(1154:2306,:),u(1154:2306,:),Ts);%converting 2nd half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='days';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y(2307:3459,:),u(2307:3459,:),Ts);%converting 3rd half into iddata
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='days';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
fourth_half=iddata(y(3459:4548,:),u(3459:4548,:),Ts);%converting 4th half into iddata
fourth_half.inputname(1)={'Ambtemp'};
```

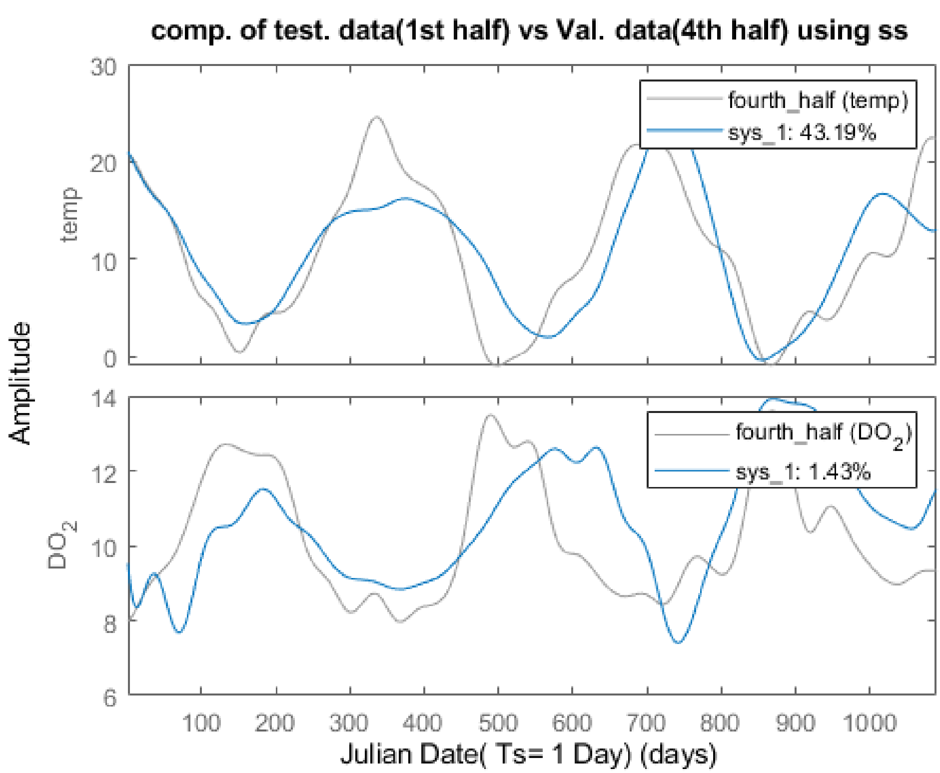
```
fourth_half.inputname(2)={'cms'};
fourth_half.timeunit='days';
fourth_half.outputname(1)={'temp'};
fourth_half.outputname(2)={'DO_2'};
sys_1=n4sid(first_half);%using n4sid to 1st half
sys_2=n4sid(second_half);%using n4sid to 2nd half
sys_3=n4sid(third_half);%using n4sid to 3rd half
sys_4=n4sid(fourth_half);%using n4sid to 4th half
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using ss')
xlabel('Julian Date( Ts= 1 Day)')
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,fourth_half)
title('comp. of test. data(1st half) vs Val. data(4th half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
```

```
sys =
  Discrete-time identified state-space model:
    x(t+Ts) = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
               x1          x2          x3          x4
   x1           1   3.137e-05    0.001944     0.01542
   x2     0.01265           1     0.04603   -0.003061
   x3   -0.009793     -0.0134      0.9936    -0.00557
   x4    -0.01461    0.005665    0.005608      0.9837

  B =
           Ambtemp          cms
   x1   -6.614e-09      9.4e-06
   x2   -5.077e-07    4.401e-05
   x3   -1.091e-05   -1.603e-06
   x4     2.49e-05    0.0001088

  C =
               x1        x2        x3        x4
   temp    -50.48    -176.1    -4.087   -0.1168
   DO_2     173.4     32.16     0.906     1.275

  D =
          Ambtemp       cms
   temp         0         0
   DO_2         0         0
```

```
  K =
            temp       DO_2
   x1  0.0006697   0.003811
   x2  -0.003797  -0.000939
   x3   -0.02102  -0.001399
   x4    0.01275    0.06282

Sample time: 1 days

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 40
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using N4SID on time domain data "daily".
Fit to estimation data: [99.56;99.1]% (prediction focus)
FPE: 2.218e-07, MSE: 0.001182
```







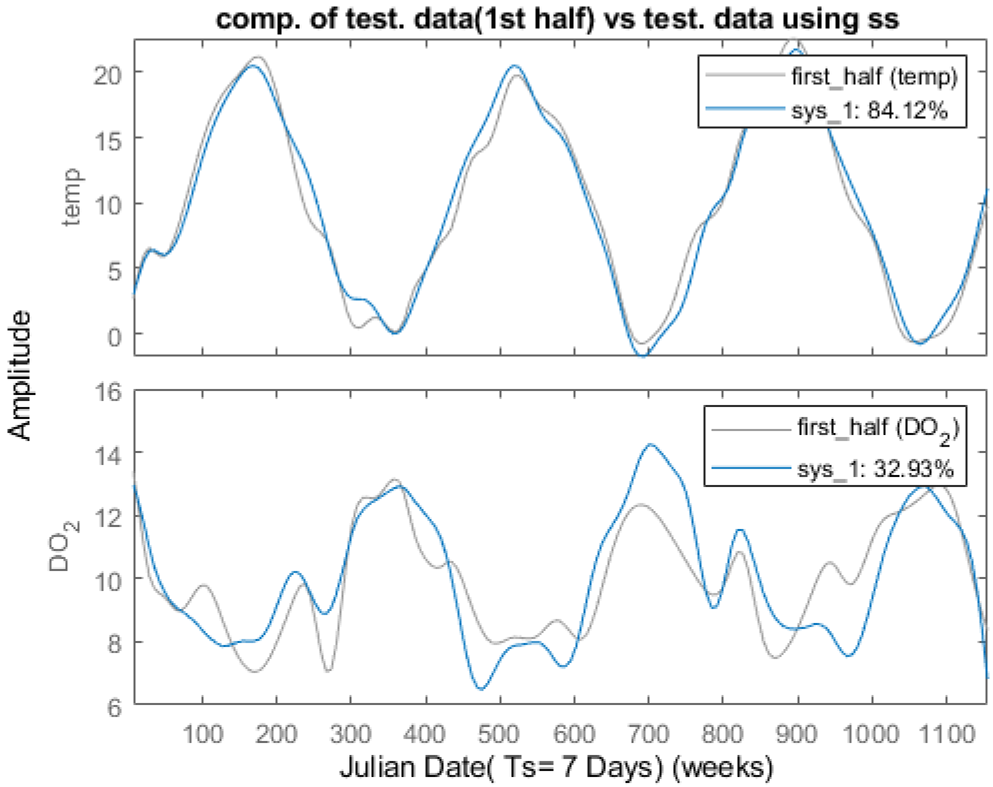
PublishedwithMATLAB®R2020b

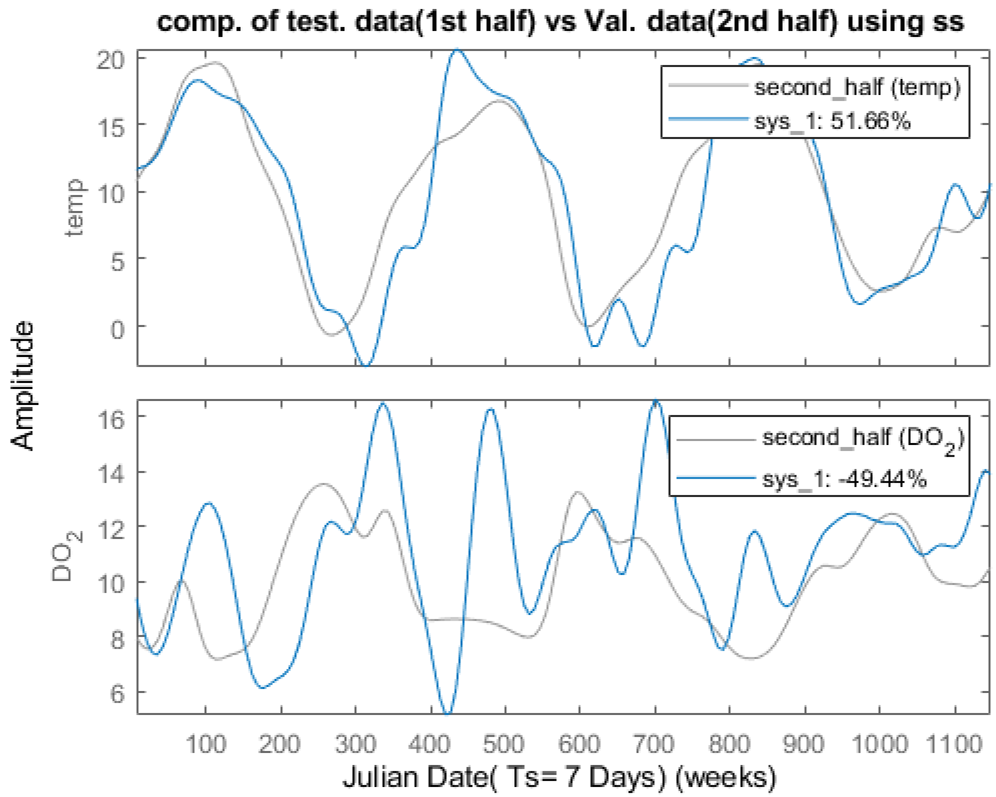
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset.
%It will compare the result on the basis of weekly data excluding three years using state space
%for the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%ss. = state space
close all;
clc;
clear;
load fichter_excluding_three_years_weekly.mat;%load fichter data excluding three years
y7=[Temp,DO_2];%load outputs of the system
u7=[Ambtemp_time,cms];%load inputs of the system
weekly=iddata(y7,u7,Ts);%converting inputs and outputs into iddata
%giving names to inputs and outputs
weekly.inputname(1)={'Ambtemp'};
weekly.inputname(2)={'cms'};
weekly.timeunit='weeks';
weekly.outputname(1)={'temp'};
weekly.outputname(2)={'DO_2'};
sys=n4sid((weekly),4)
first_half= iddata(y7(1:165,:),u7(1:165,:),Ts);%converting 1st half to iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='weeks';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y7(166:329,:),u7(166:329,:),Ts);%converting 2nd half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='weeks';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y7(329:494,:),u7(329:494,:),Ts);%converting 3rd half into iddata
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='weeks';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
fourth_half=iddata(y7(495:650,:),u7(495:650,:),Ts);%converting 4th half into iddata
fourth_half.inputname(1)={'Ambtemp'};
fourth_half.inputname(2)={'cms'};
fourth_half.timeunit='weeks';
fourth_half.outputname(1)={'temp'};
fourth_half.outputname(2)={'DO_2'};
sys_1=n4sid((first_half));%using n4sid to 1st half
sys_2=n4sid((second_half));%using n4sid to 2nd half
sys_3=n4sid((third_half));%using n4sid to 3rd half
sys_4=n4sid((fourth_half));%using n4sid to 4th half
```

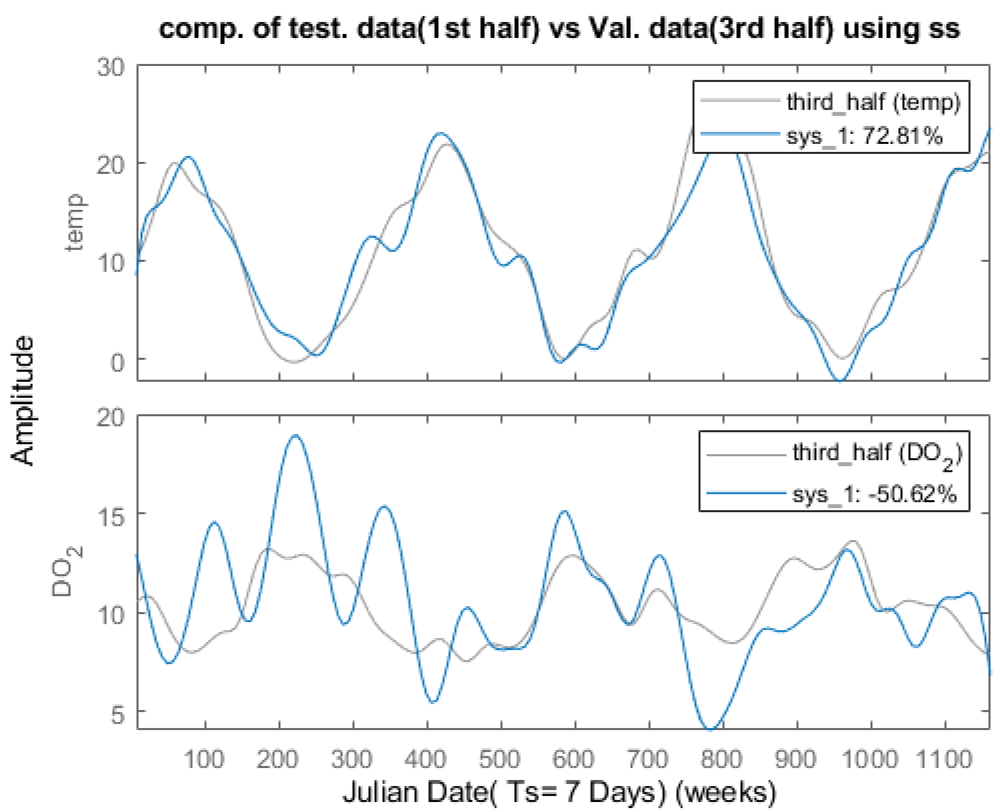
```
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure ()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure()
compare(sys_1,fourth_half)
title('comp. of test. data(1st half) vs Val. data(4th half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
```

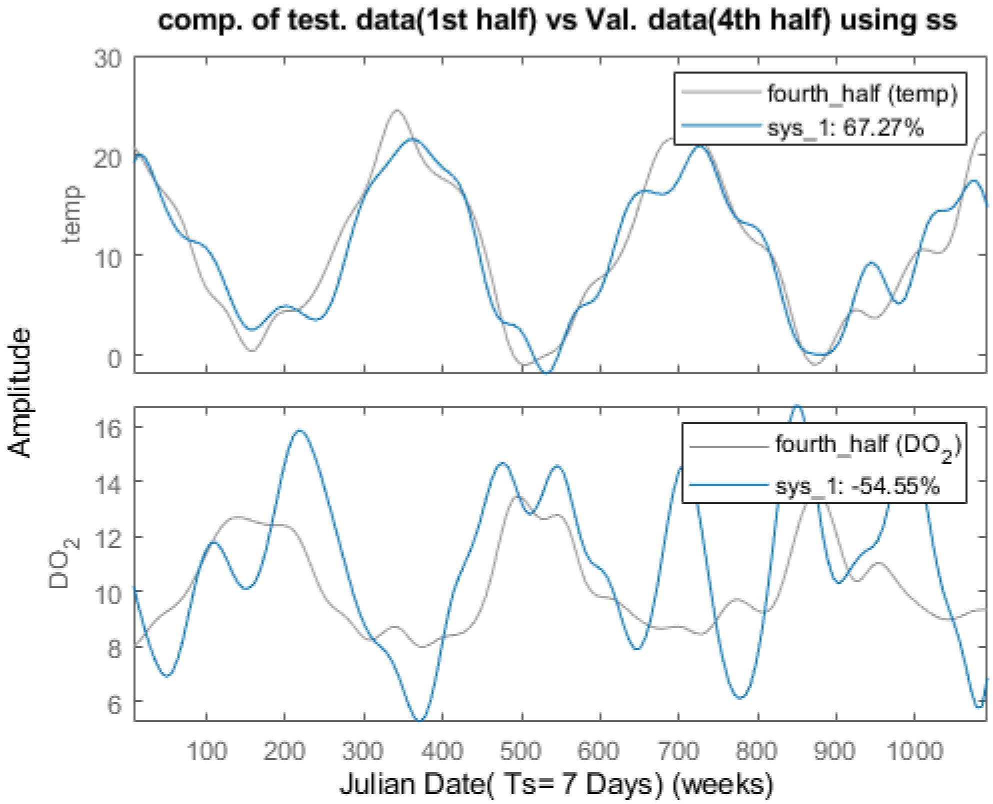
```
sys =
  Discrete-time identified state-space model:
    x(t+Ts) = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
             x1         x2         x3         x4
   x1     1.008     0.0207    0.09254      0.134
   x2   -0.0609     0.8615    -0.2484     0.1292
   x3  -0.06836     0.1421     0.8713    -0.1812
   x4  -0.09684    -0.1082   -0.04761     0.3351

  B =
          Ambtemp          cms
   x1   4.461e-06    -0.001402
   x2   0.0001649    -0.003551
   x3  -0.0007218     0.002016
   x4    0.001144      0.01303

  C =
             x1       x2       x3       x4
   temp   -23.84    54.91   -8.118    2.112
   DO_2    55.89   -3.473      2.8    2.436

  D =
         Ambtemp       cms
   temp        0         0
   DO_2        0         0

  K =
             temp        DO_2
   x1    0.001725      0.0188
   x2     0.01585    0.007255
   x3    -0.02434   -0.004789
   x4    0.006036    0.007623
```

```
Sample time: 7 weeks

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 40
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using N4SID on time domain data "weekly".
Fit to estimation data: [92.07;85.65]% (prediction focus)
FPE: 0.01831, MSE: 0.3666
```




comp. of test. data(1st half) vs Val. data(2nd half) using ss
Amplitude
temp
second_half (temp)
sys_1: 51.66%
DO_2
second_half (DO_2)
sys_1: -49.44%
Julian Date(Ts= 7 Days) (weeks)


comp. of test. data(1st half) vs Val. data(3rd half) using ss
Amplitude
temp
third_half (temp)
sys_1: 72.81%
DO_2
third_half (DO_2)
sys_1: -50.62%
Julian Date(Ts= 7 Days) (weeks)



PublishedwithMATLAB®R2020b

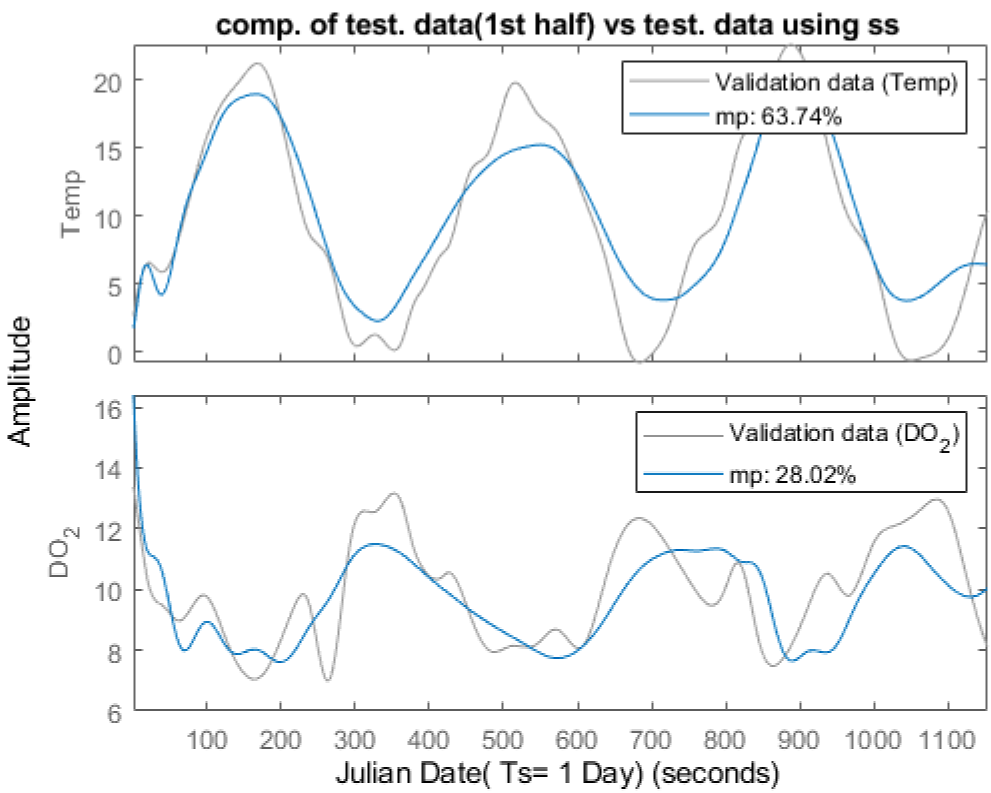
2. *Data Into Four Quarters Using SSEST*

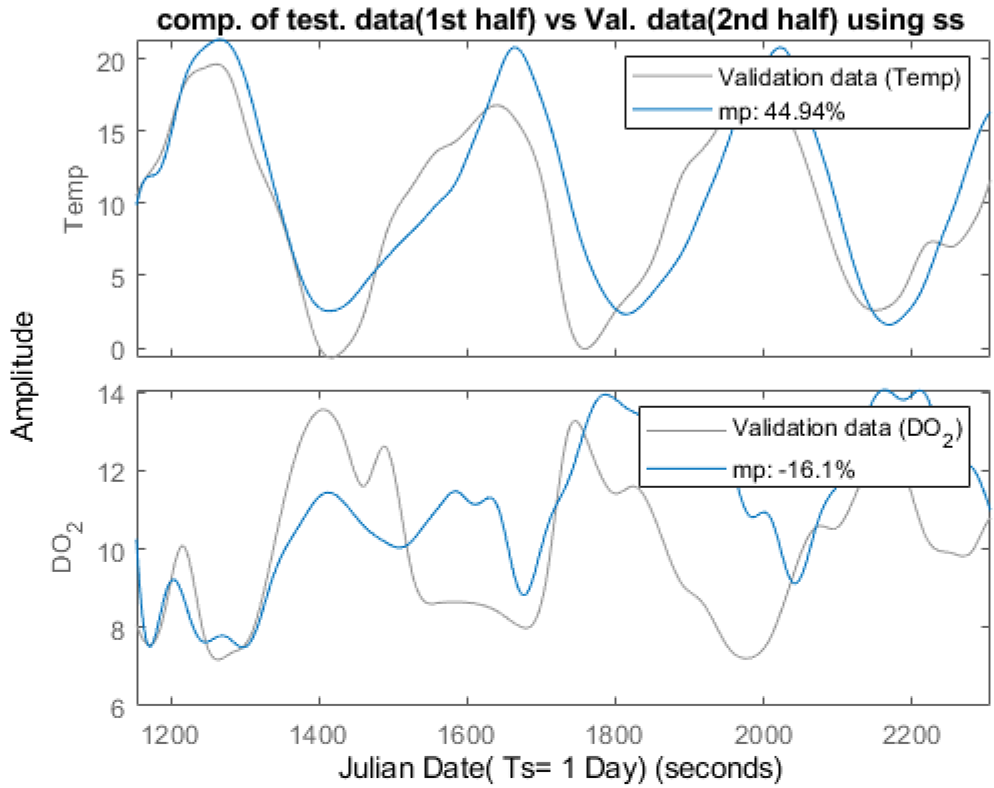
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset.
%It will compare the result on the basis of daily data excluding three years using state space
%for the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%ss. = state space
close all;
clc;
clear;
load fichter_excluding_three_years_daily.mat;%load fichter data excluding three years
Daily=iddata([Temp,DO_2],[Ambtemp_time,cms],Ts);%converting inputs and outputs into iddata
Daily.InputName={'Ambtemp_time','cms'};%giving names to inputs of the system
Daily.Outputname={'Temp';'DO_2'};%giving names to output of the system
daily.timeunit='days';%time unit
mp=ssest(Daily(1:1153))%using ssest to 1st half
figure(7)
compare(Daily(1:1153),mp);
title('comp. of test. data(1st half) vs test. data using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure(8)
compare(Daily(1154:2306),mp);
title('comp. of test. data(1st half) vs Val. data(2nd half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure(9)
compare(Daily(2307:3459),mp);
title('comp. of test. data(1st half) vs Val. data(3rd half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure(10)
compare(Daily(3460:4548),mp);
title('comp. of test. data(1st half) vs Val. data(4th half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
```

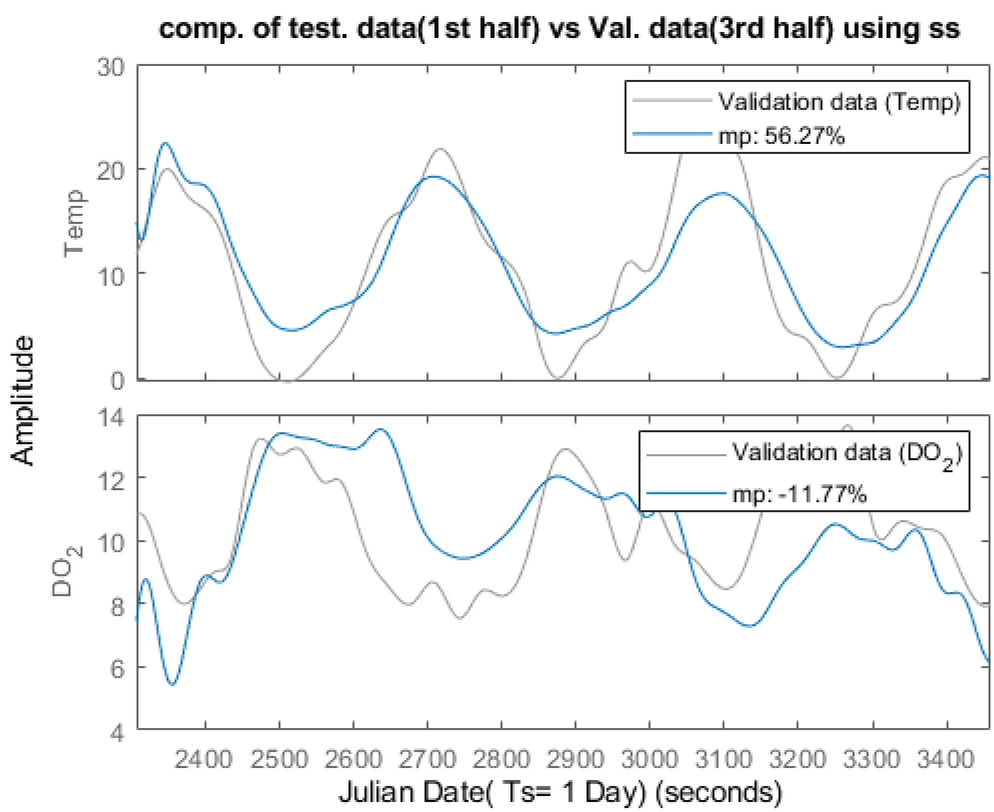
```
mp =
  Continuous-time identified state-space model:
      dx/dt = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
              x1         x2         x3         x4         x5         x6
   x1    0.04591  -0.003035    -0.1248   -0.02865   -0.01099  -0.003523
   x2      0.127     -0.024      0.078    -0.1149    0.03694  -0.009262
   x3    0.01984   0.006262      0.004   0.007158    -0.1412   -0.03958
   x4    0.03512  0.0001703   -0.02345    0.05122    0.01121     0.1723
   x5    0.02138   -0.00037    0.04958     0.0311   -0.01012   -0.02095
```

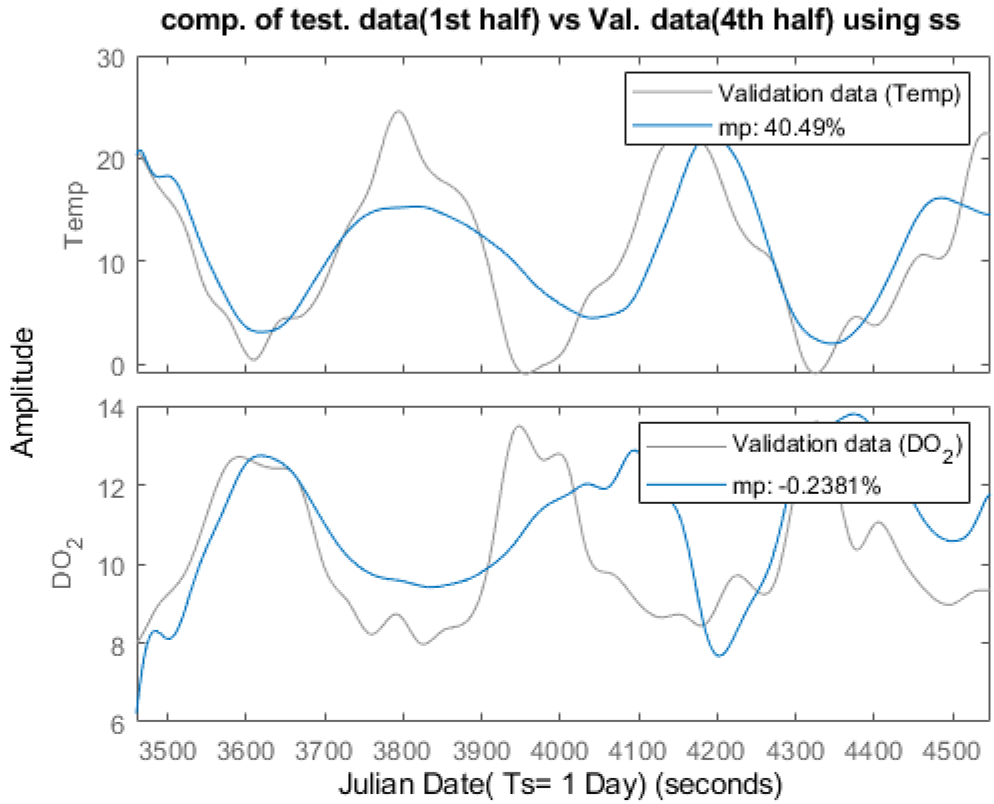
```
  x6    0.03256   -0.02593   0.001326   0.008405    0.05581   -0.01799
  x7    0.03413  -0.003235  -0.003684    0.05992    0.04263    0.04501
  x8   -0.01766    0.01763     0.0179   -0.03768    0.01267    -0.1215

             x7         x8
  x1  -0.004941   0.003374
  x2   0.003082    0.01959
  x3   0.001081    0.02258
  x4    0.00225    -0.0543
  x5   -0.08328   -0.04616
  x6    -0.1011     0.1412
  x7    -0.1156   -0.07453
  x8   -0.00376    -0.1394

 B =
      Ambtemp_time           cms
  x1    -1.329e-09    -0.0002656
  x2    -3.186e-08    -0.0007678
  x3     1.075e-07     3.816e-05
  x4     -1.19e-06    -0.0008721
  x5    -4.257e-07    -0.0001303
  x6     1.581e-06     0.0003113
  x7     2.284e-06    -0.0003611
  x8    -4.314e-06     0.0001376

 C =
               x1         x2         x3         x4         x5         x6
  Temp      14.69      63.78      0.452     -1.412   -0.01179   -0.04446
  DO_2     -67.51     -18.67     0.5342     0.5705   -0.02496    0.01608

               x7         x8
  Temp   0.003965  -0.001596
  DO_2   0.002108   0.002938

 D =
        Ambtemp_time           cms
  Temp             0             0
  DO_2             0             0

 K =
           Temp       DO_2
  x1  -0.007802   -0.02819
  x2    0.03054   0.005301
  x3     0.1853     0.4619
  x4    -0.1982     0.1382
  x5      -1.04     -3.088
  x6     -2.106     0.8985
  x7      3.327      5.613
  x8     0.6618      4.775

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
```

```
   Number of free coefficients: 112
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using SSEST on time domain data.
Fit to estimation data: [100;99.99]% (prediction focus)
FPE: 5.217e-16, MSE: 7.357e-08
```




comp. of test. data(1st half) vs Val. data(2nd half) using ss
Validation data (Temp)
mp: 44.94%
Validation data (DO2)
mp: -16.1%
Temp
DO2
Amplitude
Julian Date(Ts= 1 Day) (seconds)


comp. of test. data(1st half) vs Val. data(3rd half) using ss
Validation data (Temp)
mp: 56.27%
Validation data (DO2)
mp: -11.77%
Temp
DO2
Amplitude
Julian Date(Ts= 1 Day) (seconds)



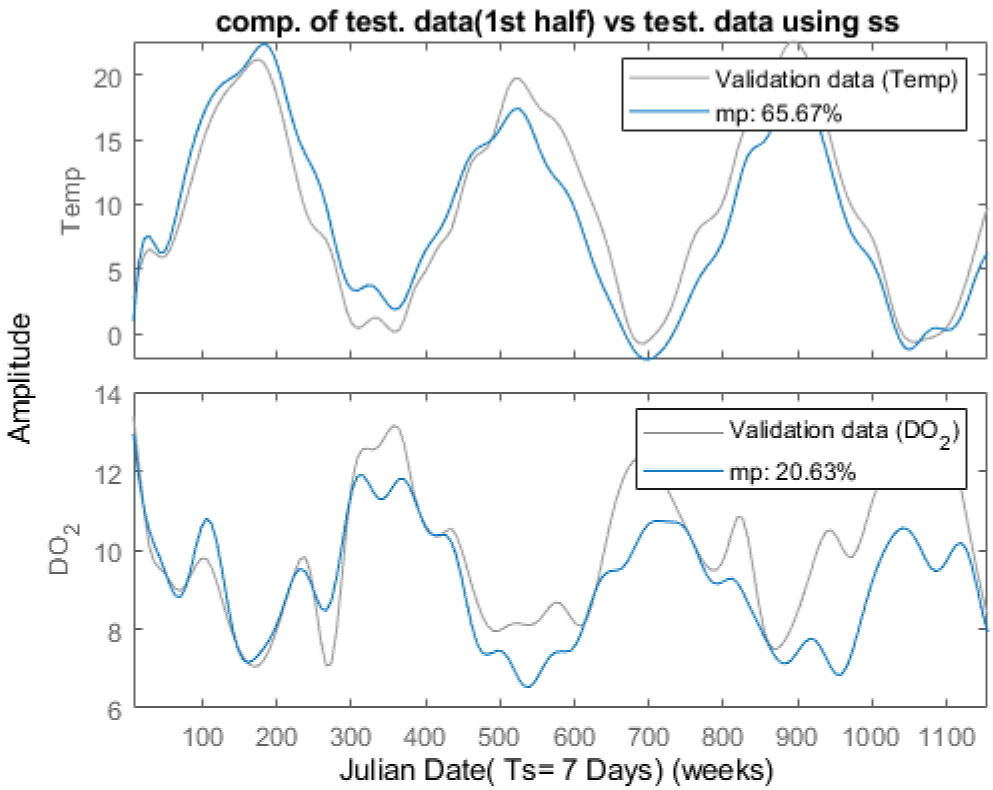
PublishedwithMATLAB®R2020b

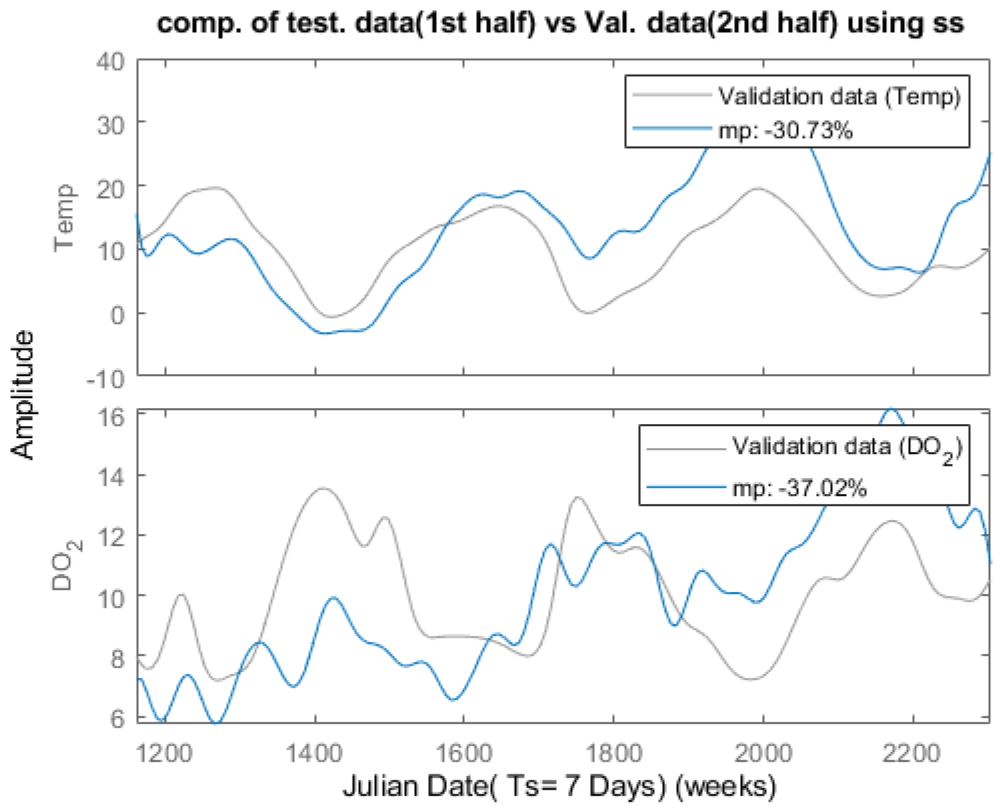
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset.
%It will compare the result on the basis of daily data excluding three years using state space
%for the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%ss. = state space
close all;
clc;
clear;
load fichter_excluding_three_years_weekly;%load fichter data excluding three years
weekly=iddata([Temp,DO_2],[Ambtemp_time,cms],Ts);%converting inputs and outputs to iddata
weekly.InputName={'Ambtemp_time';'cms'};%giving names to inputs
weekly.Outputname={'Temp';'DO_2'};%giving names to output
weekly.timeunit='weeks';%time unit
mp=ssest(weekly(1:165))%using ssest to first half
figure(8)
compare(weekly(1:165),mp);
title('comp. of test. data(1st half) vs test. data using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure(9)
compare(weekly(166:329),mp);
title('comp. of test. data(1st half) vs Val. data(2nd half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure(10)
compare(weekly(329:494),mp);
title('comp. of test. data(1st half) vs Val. data(3rd half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure(11)
compare(weekly(495:650),mp);
title('comp. of test. data(1st half) vs Val. data(4th half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
```

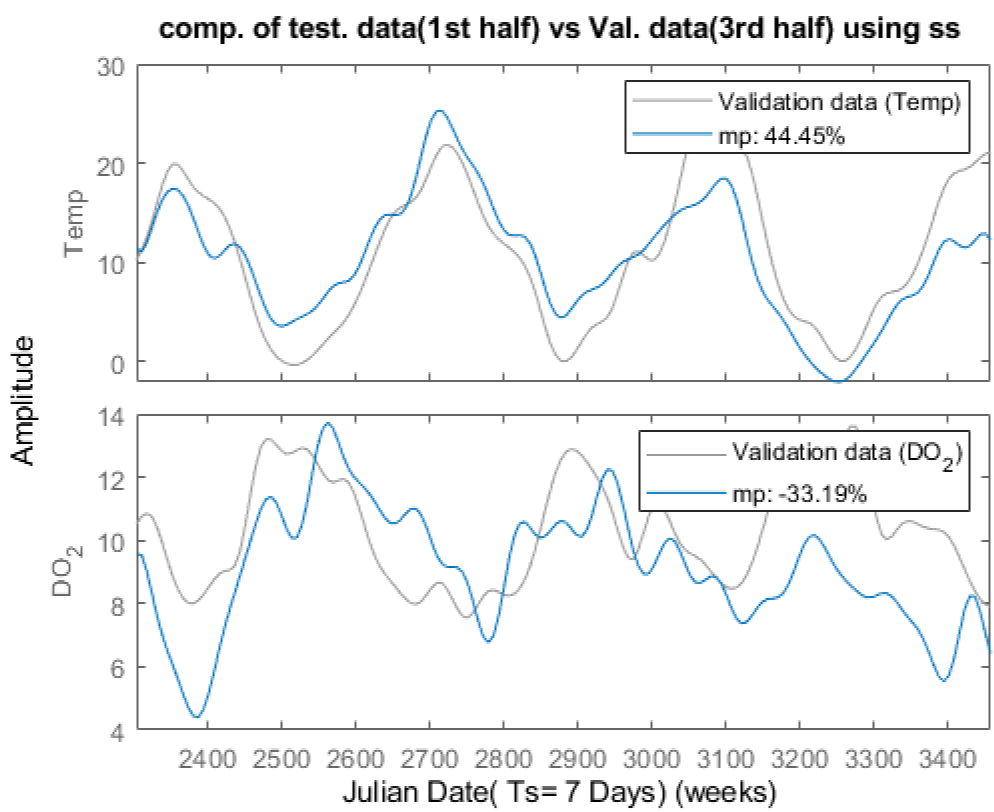
```
mp =
  Continuous-time identified state-space model:
      dx/dt = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
                x1          x2          x3          x4          x5          x6
   x1     -0.01331    -0.01434    0.006236    -0.03182    0.008654    -0.05538
   x2      0.02897    0.002828    -0.03202    -0.01431    0.003637    -0.02007
   x3     0.002712     0.01858    0.001343     0.03494     0.03958   -0.003891
   x4      0.03306    -0.01507    -0.03315    0.001748   -0.007524     0.02344
   x5     0.009429    -0.01086    -0.03246     0.01478   -0.003116    -0.02979
   x6   -0.0009044     0.03122     -0.0016    -0.03134     0.04086    -0.02228
   x7    -0.008036    0.006833     0.02086    -0.04009      0.0151     0.02171
   x8      0.02321    -0.01587    -0.01378   0.0003247     0.01564    0.001328
```

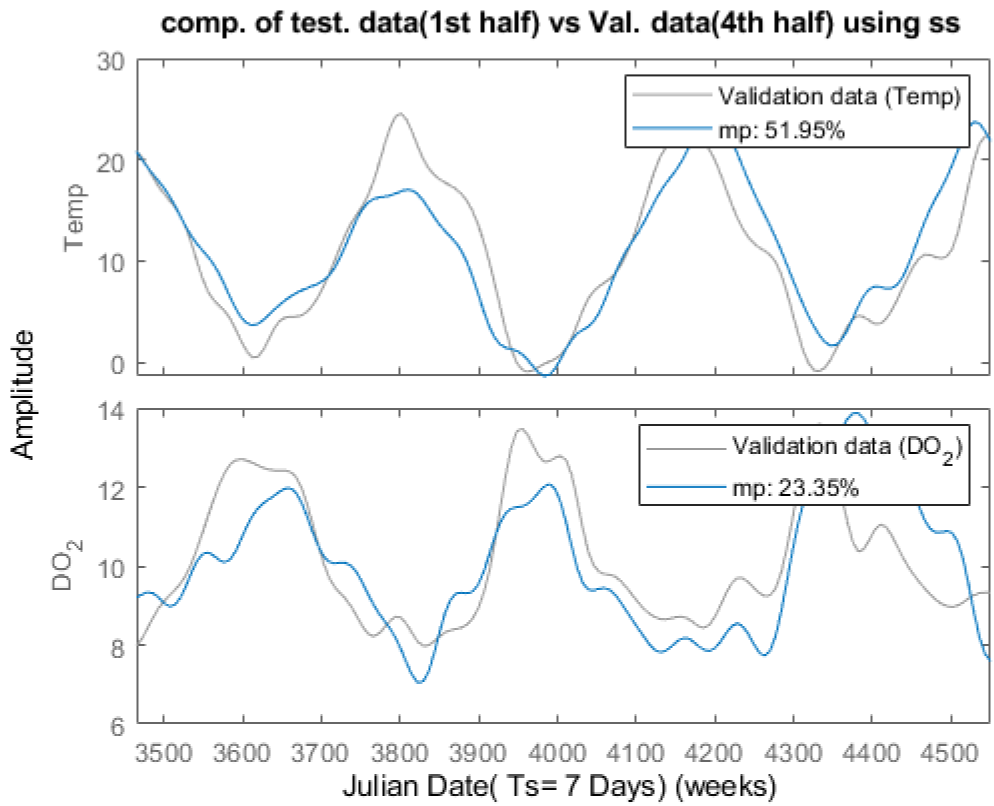
```
 x9     -0.01995   -0.008212      0.0176    -0.01544     0.01514    -0.02778
 x10    -0.03363     0.03356    -0.03307     0.02301   -0.006679     0.01949

              x7          x8          x9         x10
 x1      0.05076    -0.01355    -0.04382     0.01877
 x2     -0.02193     0.02384     0.03043    -0.02503
 x3     -0.05242     0.01421   0.0007699    0.002685
 x4       0.0237   -0.003021     0.02994     -0.0107
 x5     -0.01308   1.477e-05    -0.02525   0.0004728
 x6      0.03168    0.009036     0.00675      0.0144
 x7     -0.02641     0.03574     0.06248   -0.009968
 x8      0.03447   -0.006824   -0.005412      0.0514
 x9     -0.01604    -0.02586    -0.02126     0.01981
 x10    -0.05028    -0.08594    -0.01955    -0.01101

B =
      Ambtemp_time           cms
 x1     -2.138e-05      0.003806
 x2      0.0001018      -0.00341
 x3     -0.0003446     -0.007852
 x4     -0.0001393      0.004301
 x5      4.814e-05    -0.0008482
 x6      0.0004235    -0.0001682
 x7      0.0002477     -0.004271
 x8      0.0002148      0.003125
 x9     -0.0008356      -0.00098
 x10     -0.000986      -0.00228

C =
            x1       x2       x3       x4       x5       x6       x7
 Temp   -10.17    15.29   0.2082    2.246    -1.22   -1.199  -0.9041
 DO_2   -7.855   -13.01   -1.231    1.556    1.345    1.392  -0.2285

            x8       x9      x10
 Temp   0.4788    1.271  -0.2408
 DO_2   -0.463  -0.3343  -0.2053

D =
       Ambtemp_time           cms
 Temp             0             0
 DO_2             0             0

K =
           Temp       DO_2
 x1    0.005255   0.003899
 x2   -0.001179   -0.01133
 x3    0.005051  -0.009495
 x4     0.02979    0.03845
 x5    -0.01867   -0.01334
 x6     0.04043    0.08398
 x7    -0.09046    -0.0229
 x8     0.02142   -0.01567
 x9     0.07973     0.0576
 x10   -0.01061   -0.01439
```

```
Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 160
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using SSEST on time domain data.
Fit to estimation data: [98.58;96.83]% (prediction focus)
FPE: 6.352e-05, MSE: 0.01284
```




comp. of test. data(1st half) vs Val. data(2nd half) using ss
Validation data (Temp)
mp: -30.73%
Validation data (DO_2)
mp: -37.02%
Temp
DO_2
Amplitude
Julian Date(Ts= 7 Days) (weeks)


comp. of test. data(1st half) vs Val. data(3rd half) using ss
Validation data (Temp)
mp: 44.45%
Validation data (DO_2)
mp: -33.19%
Temp
DO_2
Amplitude
Julian Date(Ts= 7 Days) (weeks)



PublishedwithMATLAB®R2020b

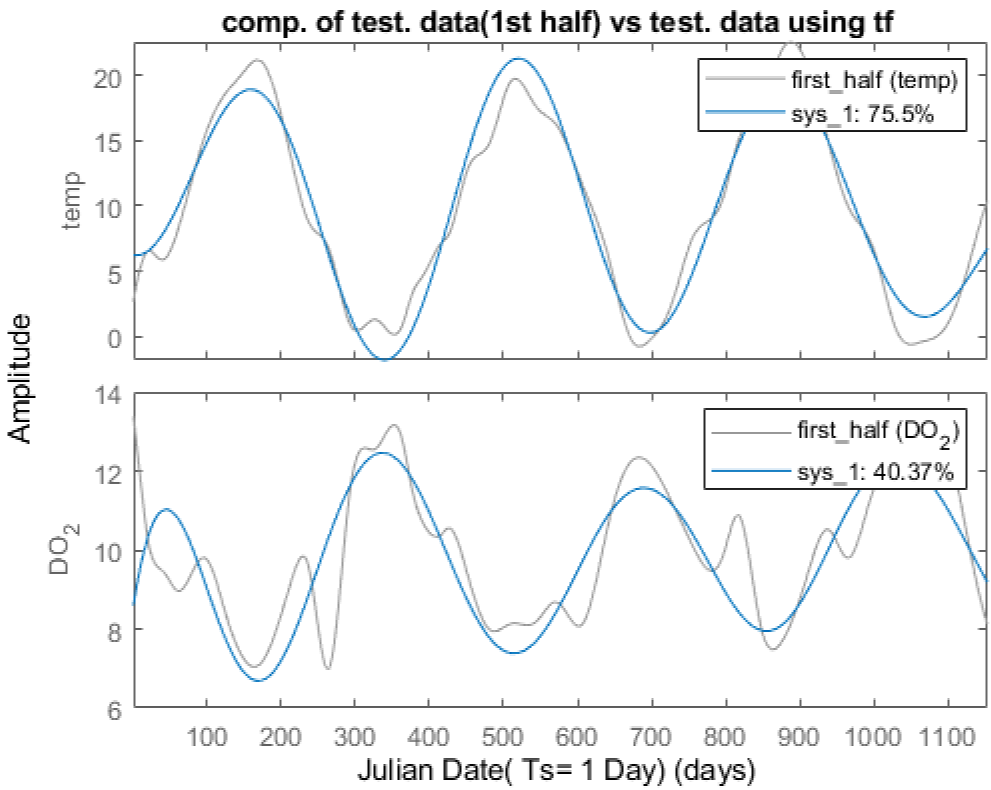
3. Data Into Four Quarters Using TFEST

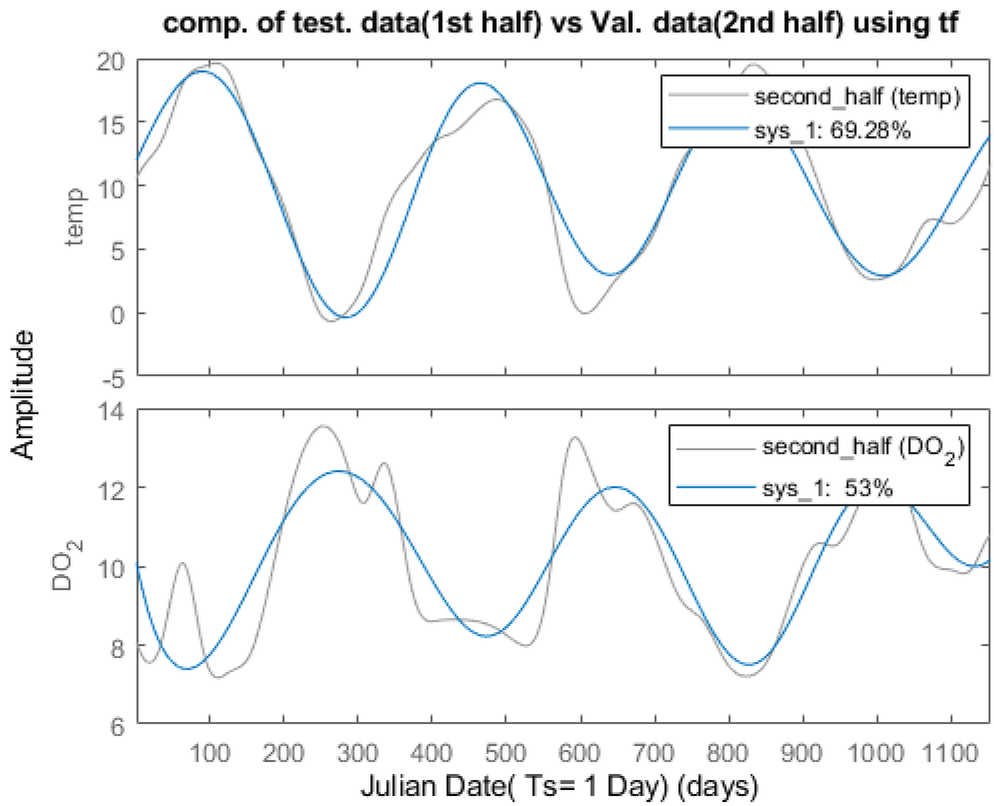
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset.
%It will compare the result on the basis of daily data excluding three years using state space
%for the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%tf. = transfer function
close all;
clc;
clear;
load fichter_excluding_three_years_daily.mat;%load fichter data excluding three years
y=[Temp,DO_2];%load output of the system
u=[Ambtemp_time,cms];%load inputs of the system
daily=iddata(y,u,Ts);%converting inputs and outputs into iddata
%giving names to inputs and outputs of the system
daily.inputname(1)={'Ambtemp'};
daily.inputname(2)={'cms'};
daily.timeunit='days';
daily.outputname(1)={'temp'};
daily.outputname(2)={'DO_2'};
sys=tfest(daily,4,0)
first_half= iddata(y(1:1153,:),u(1:1153,:),Ts);%converting 1st half into iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='days';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y(1154:2306,:),u(1154:2306,:),Ts);%converting 2nd half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='days';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'}
third_half=iddata(y(2307:3459,:),u(2307:3459,:),Ts);%converting 3rd half into iddata
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='days';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
fourth_half=iddata(y(3459:4548,:),u(3459:4548,:),Ts);%converting 4th half into iddata
fourth_half.inputname(1)={'Ambtemp'};
fourth_half.inputname(2)={'cms'};
fourth_half.timeunit='days';
fourth_half.outputname(1)={'temp'};
```

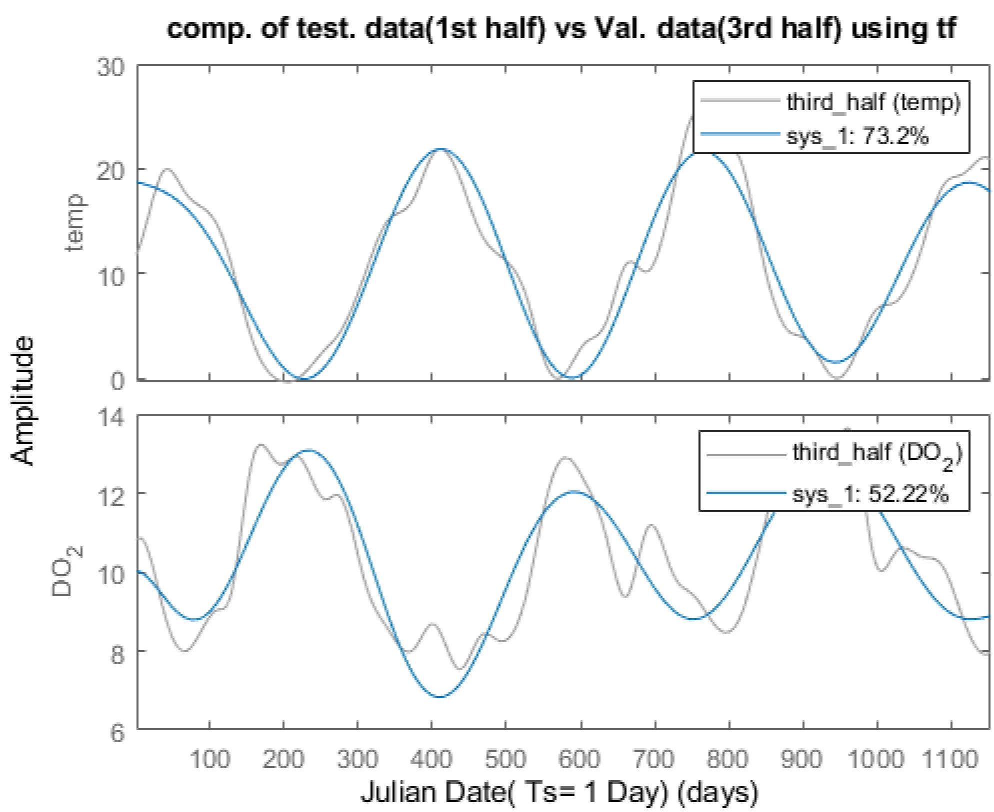
```
fourth_half.outputname(2)={'DO_2'};
sys_1=tfest(first_half,4,0);%using tfest to 1st half
sys_2=tfest(second_half,4,0);%using tfest to 2nd half
sys_3=tfest(third_half,4,0);%using tfest to 3rd half
sys_4=tfest(fourth_half,4,0);%using tfest to 4th half
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using tf')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using tf')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) using tf')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,fourth_half)
title('comp. of test. data(1st half) vs Val. data(4th half) using tf')
xlabel('Julian Date( Ts= 1 Day)')
```

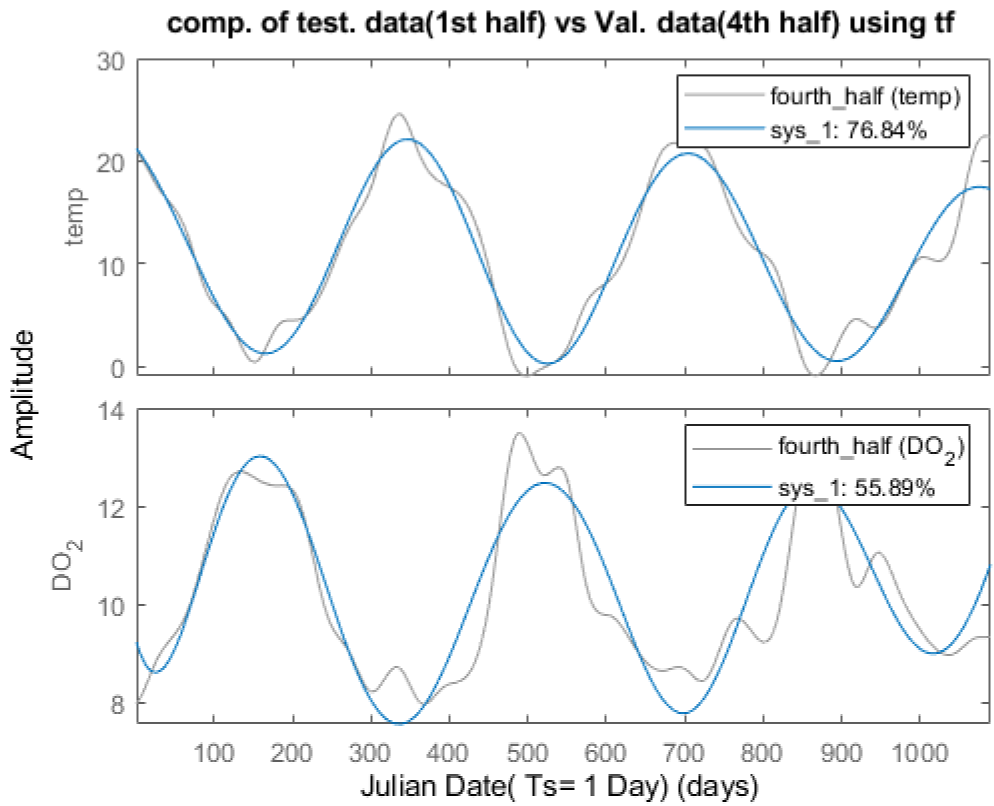
```
sys =

  From input "Ambtemp" to output...
                                    7.942e-09
   temp:  -----------------------------------------------------------
          s^4 + 0.05495 s^3 + 0.0003844 s^2 + 1.581e-05 s + 2.231e-08

                                   -2.603e-11
   DO_2:  -----------------------------------------------------------
          s^4 + 0.001599 s^3 + 2.08e-05 s^2 + 9.502e-09 s + 4.246e-12

  From input "cms" to output...
                                    -1.019e-09
   temp:  ------------------------------------------------------------
          s^4 + 0.004269 s^3 + 0.0002942 s^2 + 1.233e-06 s + 8.243e-10

                                    4.678e-09
   DO_2:  ------------------------------------------------------------
          s^4 + 0.001729 s^3 + 0.0002826 s^2 + 3.413e-07 s + 1.491e-18

Continuous-time identified transfer function.

Parameterization:
   Number of poles: [4 4;4 4]   Number of zeros: [0 0;0 0]
   Number of free coefficients: 20
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "daily".
Fit to estimation data: [71.78;31.84]%
```

```
FPE: 4.757, MSE: 5.248

second_half =

Time domain dataset with 1153 samples.
Sample time: 1 days

Outputs        Unit (if specified)
   temp
   DO_2

Inputs         Unit (if specified)
   Ambtemp
   cms
```




comp. of test. data(1st half) vs Val. data(2nd half) using tf
second_half (temp)
sys_1: 69.28%
second_half (DO2)
sys_1: 53%
Amplitude
temp
DO2
Julian Date(Ts= 1 Day) (days)


comp. of test. data(1st half) vs Val. data(3rd half) using tf
third_half (temp)
sys_1: 73.2%
third_half (DO2)
sys_1: 52.22%
Amplitude
temp
DO2
Julian Date(Ts= 1 Day) (days)

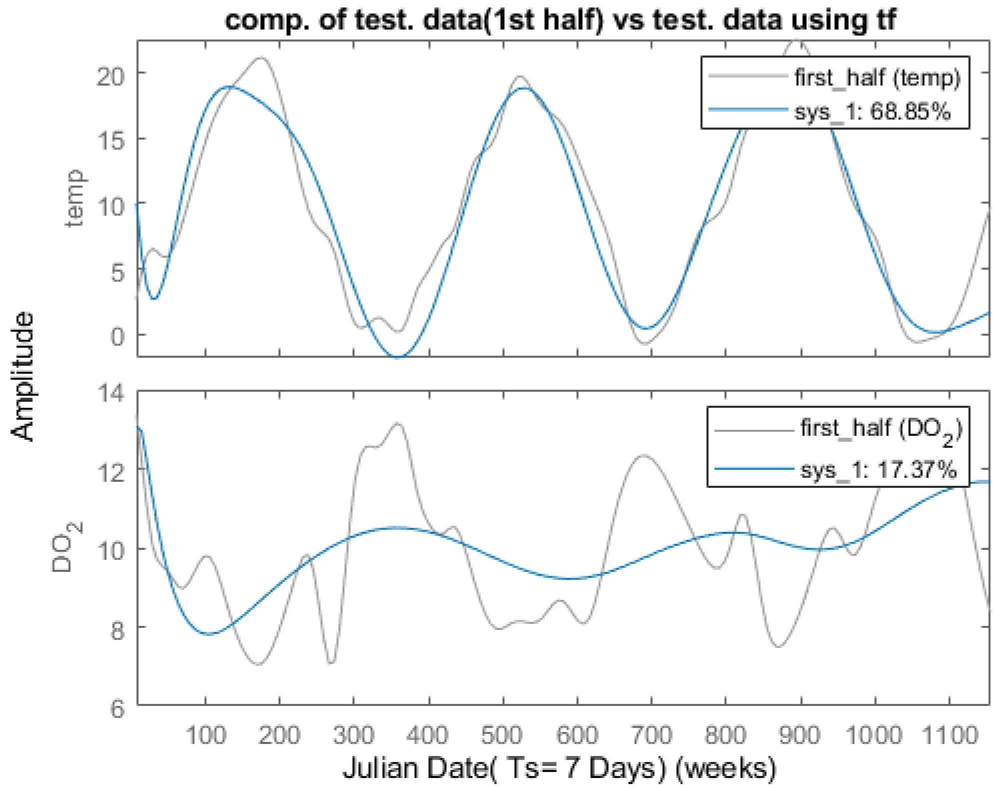


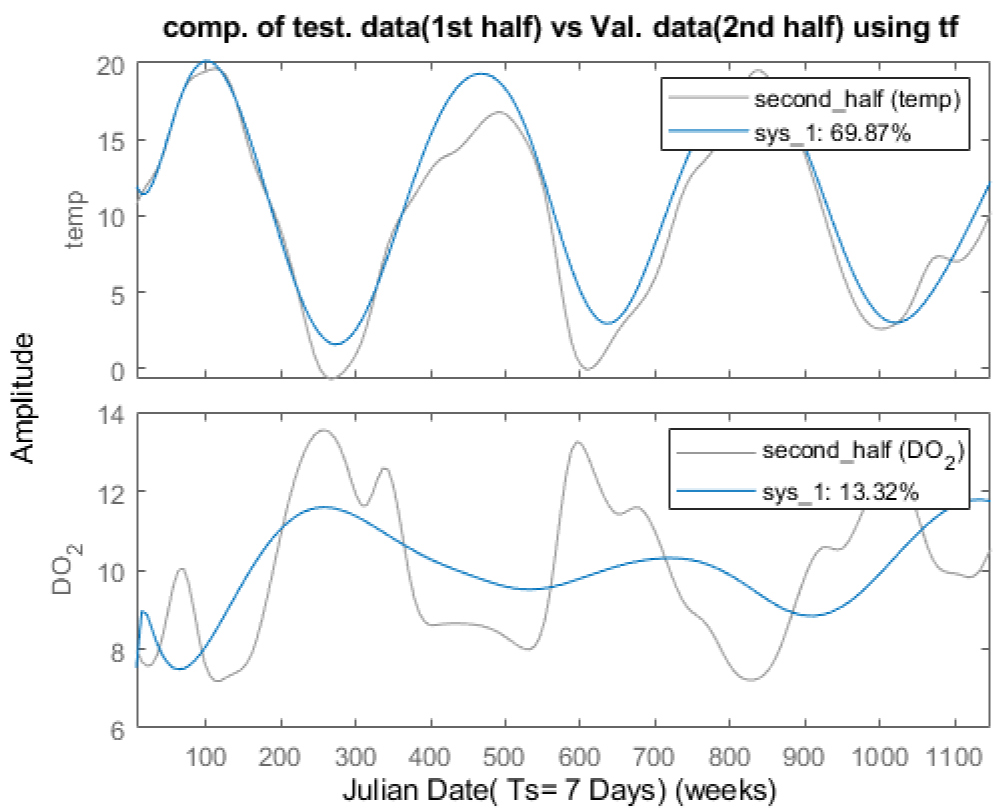
PublishedwithMATLAB®R2020b

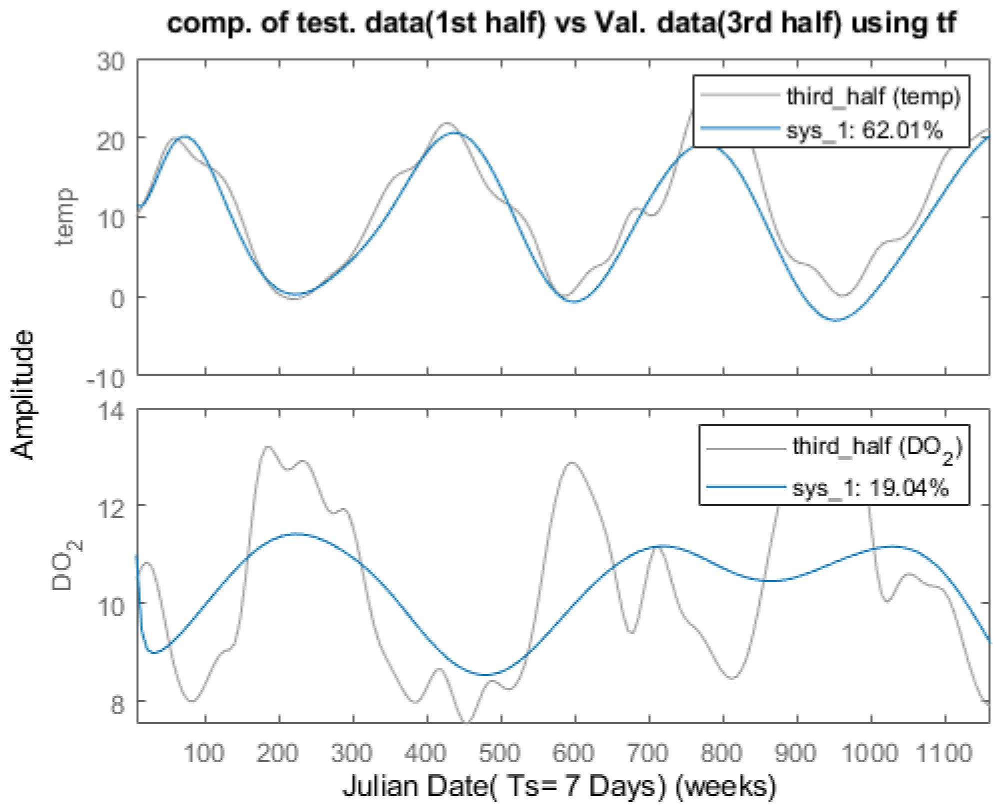
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset.
%It will compare the result on the basis of daily data excluding three years using state space
%for the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%tf. = transfer function
close all;
clc;
clear;
load fichter_excluding_three_years_weekly.mat;%load fichter data excluding three years
y7=[Temp,DO_2];%load output of the system
u7=[Ambtemp_time,cms];%load inputs of the system
weekly=iddata(y7,u7,Ts);%converting inputs and outputs into iddata
%giving names to inputs and outputs of the system
weekly.inputname(1)={'Ambtemp'};
weekly.inputname(2)={'cms'};
weekly.timeunit='weeks';
weekly.outputname(1)={'temp'};
weekly.outputname(2)={'DO_2'};
sys=tfest((weekly),4,0)
first_half= iddata(y7(1:165,:),u7(1:165,:),Ts);%converting 1st half into iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='weeks';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y7(166:329,:),u7(166:329,:),Ts);%converting 2nd half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='weeks';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y7(329:494,:),u7(329:494,:),Ts);%converting 3rd half into iddata
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='weeks';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
fourth_half=iddata(y7(495:650,:),u7(495:650,:),Ts);%converting 4th half into iddata
fourth_half.inputname(1)={'Ambtemp'};
fourth_half.inputname(2)={'cms'};
fourth_half.timeunit='weeks';
fourth_half.outputname(1)={'temp'};
fourth_half.outputname(2)={'DO_2'};
sys_1=tfest((first_half),4,0);%using tfest to 1st half
sys_2=tfest((second_half),4,0);%using tfest to 2nd half
sys_3=tfest((third_half),4,0);%using tfest to 3rd half
sys_4=tfest((fourth_half),4,0);%using tfest to 4th half
```

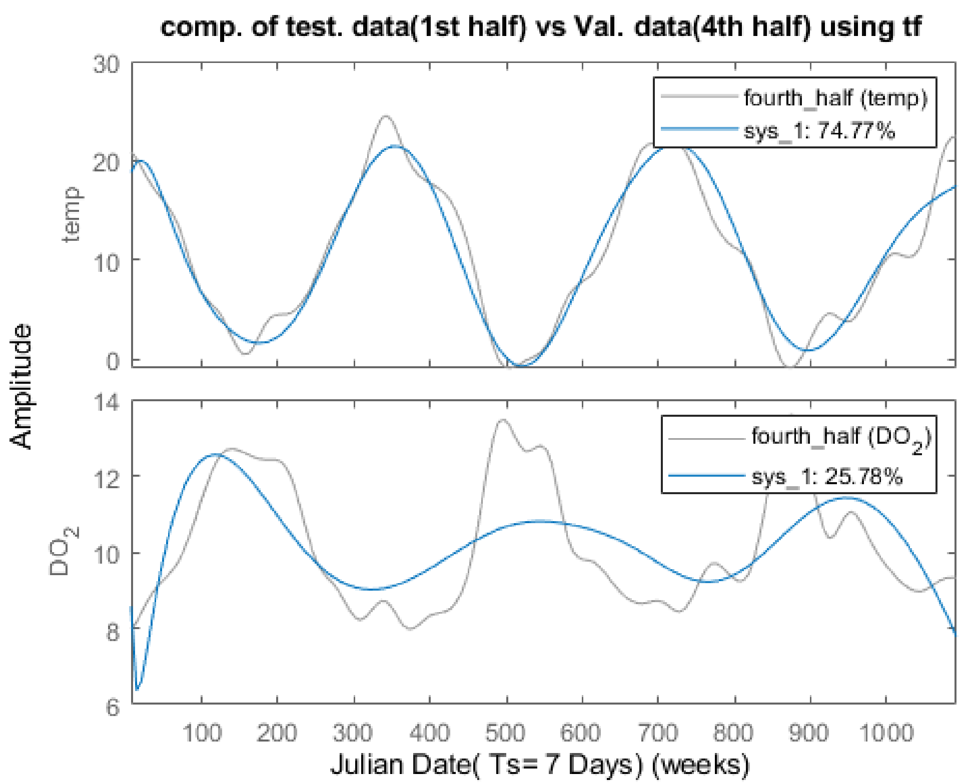
```
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using tf')
xlabel('Julian Date( Ts= 7 Days)')
figure ()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using tf')
xlabel('Julian Date( Ts= 7 Days)')
figure()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) using tf')
xlabel('Julian Date( Ts= 7 Days)')
figure()
compare(sys_1,fourth_half)
title('comp. of test. data(1st half) vs Val. data(4th half) using tf')
xlabel('Julian Date( Ts= 7 Days)')
```

```
sys =

  From input "Ambtemp" to output...
                                 2.387e-10
   temp:  -----------------------------------------------------------
          s^4 + 0.001212 s^3 + 0.0003002 s^2 + 2.594e-07 s + 1.99e-09

                                 1.296e-10
   DO_2:  ------------------------------------------------------------
          s^4 + 0.0006132 s^3 + 0.0003003 s^2 + 1.43e-07 s + 1.426e-10

  From input "cms" to output...
                                 8.045e-10
   temp:  ------------------------------------------------------------
          s^4 + 0.002049 s^3 + 0.0002984 s^2 + 4.975e-07 s + 7.767e-10

                                -1.249e-11
   DO_2:  -----------------------------------------------------------
          s^4 + 0.0009116 s^3 + 5.081e-06 s^2 + 2.816e-09 s + 1.6e-12

Continuous-time identified transfer function.

Parameterization:
   Number of poles: [4 4;4 4]   Number of zeros: [0 0;0 0]
   Number of free coefficients: 20
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "weekly".
Fit to estimation data: [73.63;47.98]%
FPE: 2.656, MSE: 4.182
```


comp. of test. data(1st half) vs test. data using tf
first_half (temp)
sys_1: 68.85%
first_half (DO2)
sys_1: 17.37%
Amplitude
temp
DO2
Julian Date(Ts= 7 Days) (weeks)


comp. of test. data(1st half) vs Val. data(2nd half) using tf
second_half (temp)
sys_1: 69.87%
second_half (DO2)
sys_1: 13.32%
Amplitude
temp
DO2
Julian Date(Ts= 7 Days) (weeks)





Published with MATLAB® R2020b

Appendix 4. System Identification of Average of Excluding Three Quarters

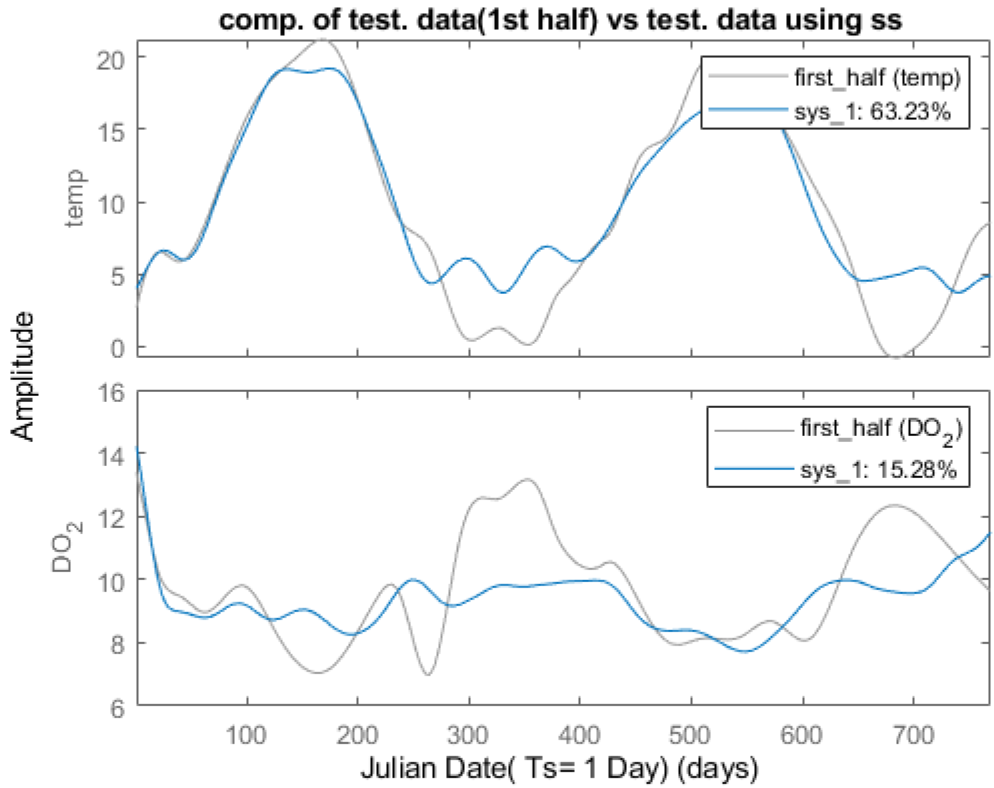
1. *Data Into Six Portions Using N4SID*

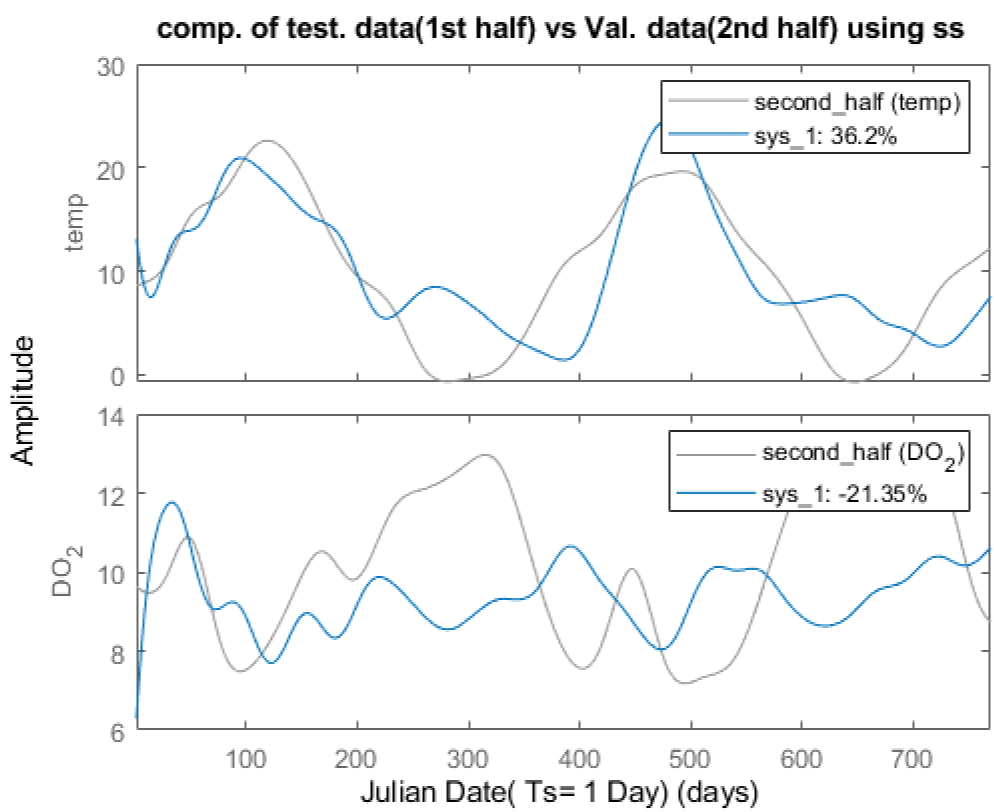
```
Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset.
%It will compare the result on the basis of daily data excluding three years using state space
%for the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%ss. = state space
close all;
clc;
clear;
load fichter_excluding_three_years_daily.mat;%loading fichter data excluding three years
y=[Temp,DO_2];%load outputs of the system
u=[Ambtemp_time,cms];%load inputs of the system
daily=iddata(y,u,Ts);%converting inputs and outputs into iddata
%giving names to inputs and outputs of the system
daily.inputname(1)={'Ambtemp'};
daily.inputname(2)={'cms'};
daily.timeunit='days';
daily.outputname(1)={'temp'};
daily.outputname(2)={'DO_2'};
sys=n4sid(daily,4)
first_half= iddata(y(1:769,:),u(1:769,:),Ts);%converting 1st half into iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='days';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y(770:1539,:),u(770:1539,:),Ts);%converting 2nd half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='days';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y(1540:2308,:),u(1540:2308,:),Ts);%converting 3rd half into iddata
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='days';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
fourth_half=iddata(y(2309:3078,:),u(2309:3078,:),Ts);%converting 4th half into idata
fourth_half.inputname(1)={'Ambtemp'};
```

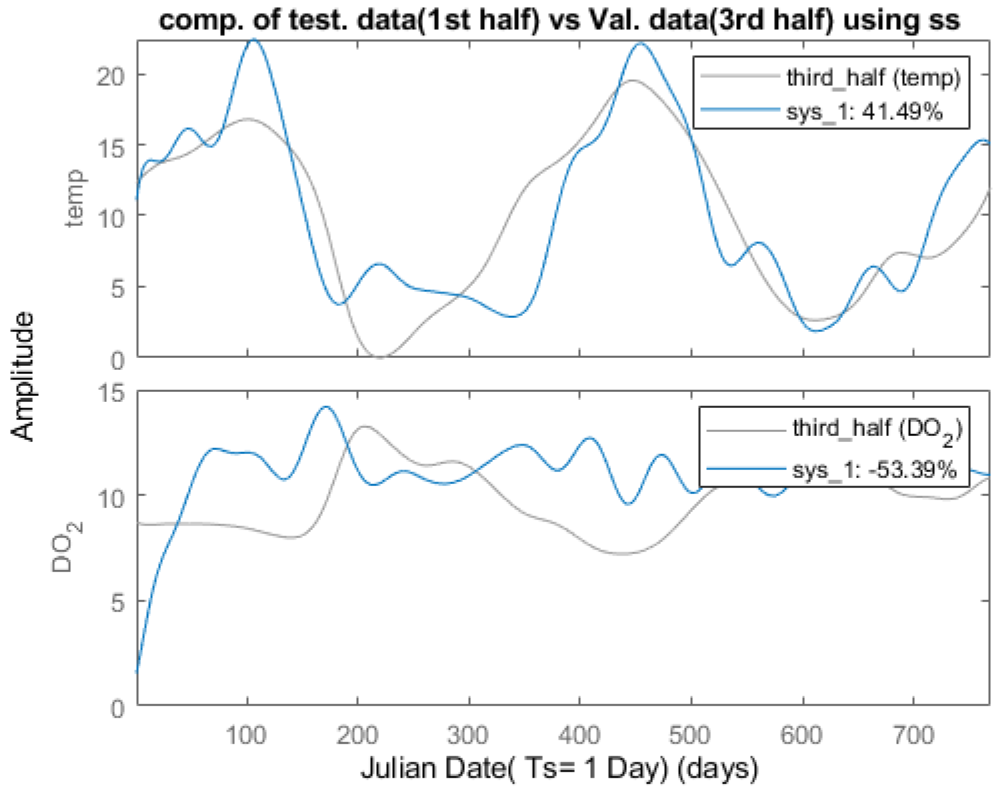
```
fourth_half.inputname(2)={'cms'};
fourth_half.timeunit='days';
fourth_half.outputname(1)={'temp'};
fourth_half.outputname(2)={'DO_2'};
fifth_half=iddata(y(3079:3848,:),u(3079:3848,:),Ts);%converting 5th half into iddata
fifth_half.inputname(1)={'Ambtemp'};
fifth_half.inputname(2)={'cms'};
fifth_half.timeunit='days';
fifth_half.outputname(1)={'temp'};
fifth_half.outputname(2)={'DO_2'};
sixth_half=iddata(y(3849:4548,:),u(3849:4548,:),Ts);%converting 6th half into iddata
sixth_half.inputname(1)={'Ambtemp'};
sixth_half.inputname(2)={'cms'};
sixth_half.timeunit='days';
sixth_half.outputname(1)={'temp'};
sixth_half.outputname(2)={'DO_2'};
sys_1=n4sid(first_half);%using n4sid to 1st half
sys_2=n4sid(second_half);%using n4sid to 2nd half
sys_3=n4sid(third_half);%using n4sid to 3rd half
sys_4=n4sid(fourth_half);%using n4sid to 4th half
sys_5=n4sid(fifth_half);%using n4sid to 5th half
sys_6=n4sid(sixth_half);%using n4sid to 6th half
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,fourth_half)
title('comp. of test. data(1st half) vs Val. data(4th half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure ()
compare(sys_1,fifth_half)
title('comp. of test. data(1st half) vs Val. data(5th half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure ()
compare(sys_1,sixth_half)
title('comp. of test. data(1st half) vs Val. data(6th half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
```

```
sys =
  Discrete-time identified state-space model:
    x(t+Ts) = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)
```

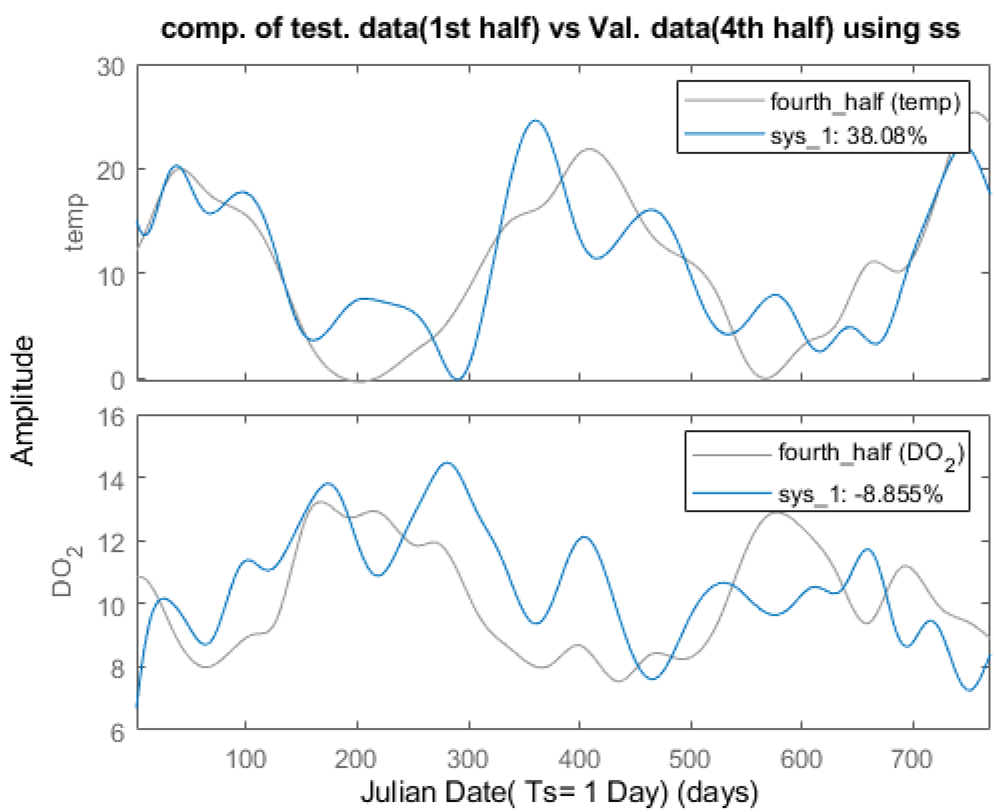
```
  A = 
            x1         x2         x3         x4
   x1          1  3.137e-05   0.001944    0.01542
   x2    0.01265          1    0.04603  -0.003061
   x3  -0.009793    -0.0134     0.9936   -0.00557
   x4   -0.01461   0.005665   0.005608     0.9837

  B = 
        Ambtemp         cms
   x1  -6.614e-09     9.4e-06
   x2  -5.077e-07   4.401e-05
   x3  -1.091e-05  -1.603e-06
   x4    2.49e-05   0.0001088

  C = 
            x1       x2       x3       x4
   temp   -50.48   -176.1   -4.087  -0.1168
   DO_2    173.4    32.16    0.906    1.275

  D = 
       Ambtemp      cms
   temp        0        0
   DO_2        0        0

  K = 
          temp       DO_2
   x1  0.0006697   0.003811
   x2  -0.003797  -0.000939
   x3   -0.02102  -0.001399
   x4    0.01275    0.06282

Sample time: 1 days

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 40
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using N4SID on time domain data "daily".
Fit to estimation data: [99.56;99.1]% (prediction focus)
FPE: 2.218e-07, MSE: 0.001182
```


comp. of test. data(1st half) vs test. data using ss
first_half (temp)
sys_1: 63.23%
first_half (DO_2)
sys_1: 15.28%
Amplitude
temp
DO_2
Julian Date(Ts= 1 Day) (days)

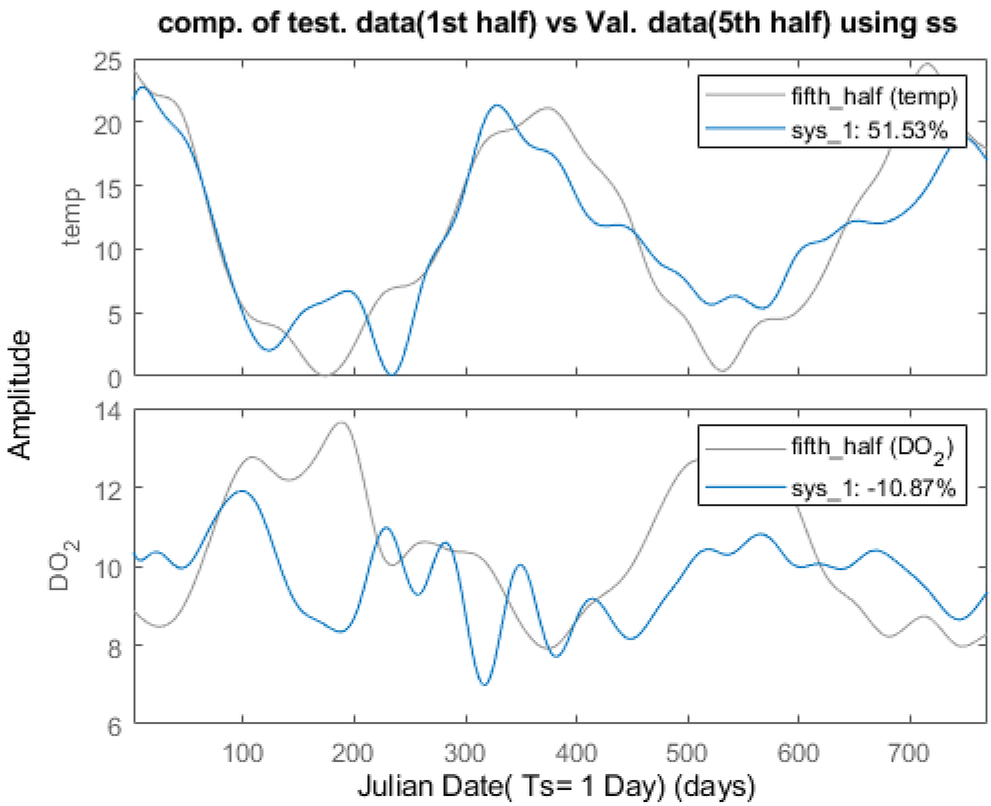

comp. of test. data(1st half) vs Val. data(2nd half) using ss
second_half (temp)
sys_1: 36.2%
second_half (DO_2)
sys_1: -21.35%
Amplitude
temp
DO_2
Julian Date(Ts= 1 Day) (days)

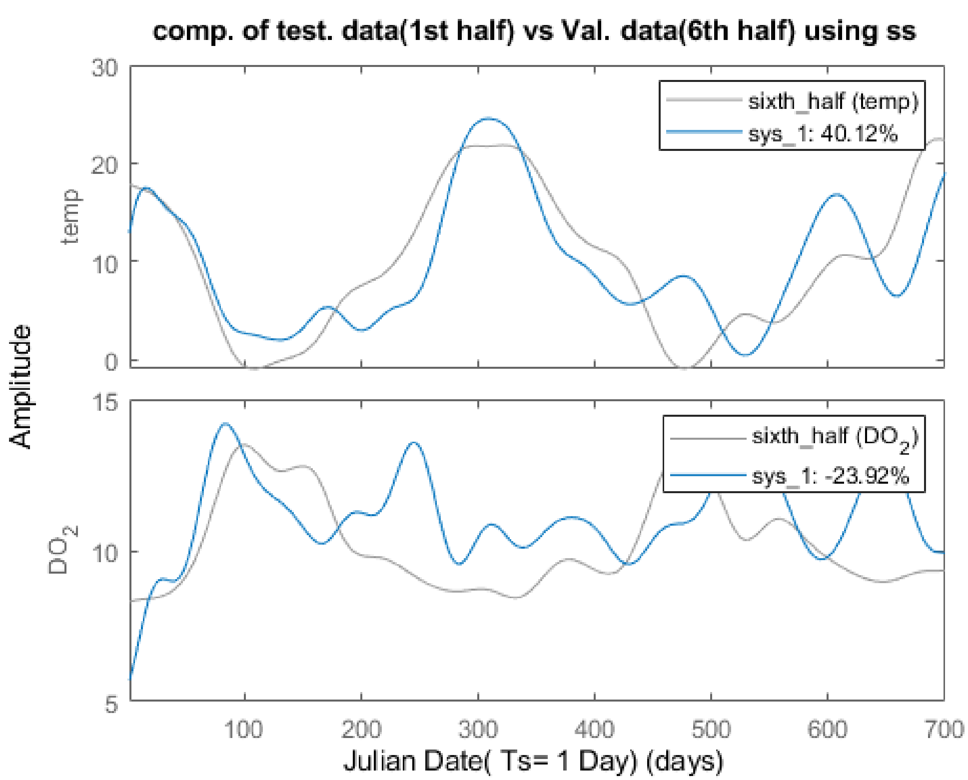


comp. of test. data(1st half) vs Val. data(3rd half) using ss
third_half (temp)
sys_1: 41.49%
third_half (DO2)
sys_1: -53.39%
Amplitude
temp
DO2
Julian Date(Ts= 1 Day) (days)



comp. of test. data(1st half) vs Val. data(4th half) using ss
fourth_half (temp)
sys_1: 38.08%
fourth_half (DO2)
sys_1: -8.855%
Amplitude
temp
DO2
Julian Date(Ts= 1 Day) (days)





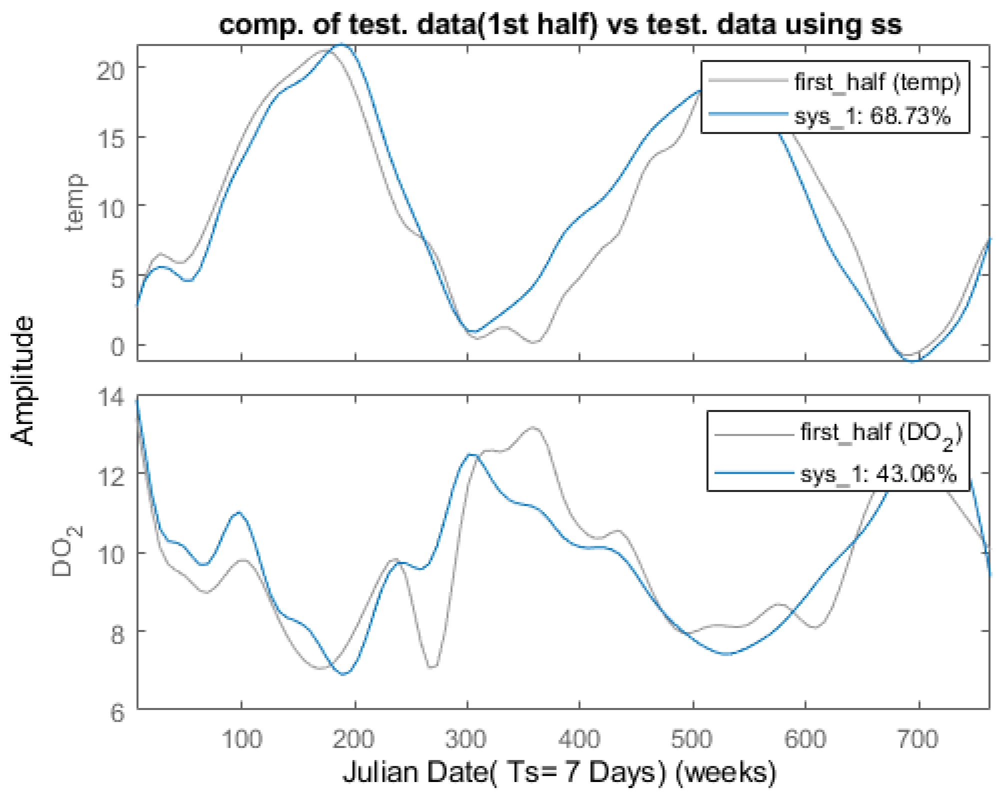
Published with MATLAB® R2020b

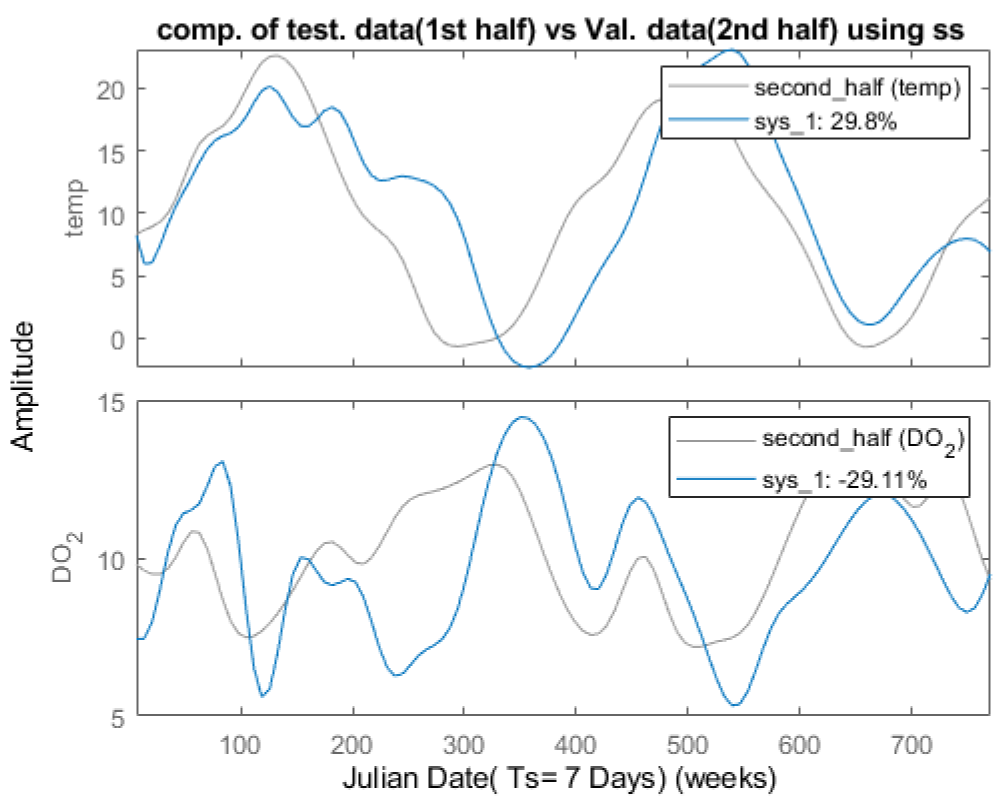
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset.
%It will compare the result on the basis of weekly data excluding three years using state space
%for the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%ss. = state space
close all;
clc;
clear;
load fichter_excluding_three_years_weekly.mat;%load fichter data excluding three years
y7=[Temp,DO_2];%load output of the system
u7=[Ambtemp_time,cms];%load inputs of the system
weekly=iddata(y7,u7,Ts);%converting inputs and outputs of the system
%giving names to inputs and outputs of the system
weekly.inputname(1)={'Ambtemp'};
weekly.inputname(2)={'cms'};
weekly.timeunit='weeks';
weekly.outputname(1)={'temp'};
weekly.outputname(2)={'DO_2'};
sys=n4sid((weekly),4)
first_half= iddata(y7(1:109,:),u7(1:109,:),Ts);%converting 1st half into iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='weeks';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y7(110:219,:),u7(110:219,:),Ts);%converting 2nd half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='weeks';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y7(220:329,:),u7(220:329,:),Ts);%converting 3rd half into iddata
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='weeks';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
fourth_half=iddata(y7(330:449,:),u7(330:449,:),Ts);%converting 4th half into iddata
fourth_half.inputname(1)={'Ambtemp'};
fourth_half.inputname(2)={'cms'};
fourth_half.timeunit='weeks';
fourth_half.outputname(1)={'temp'};
fourth_half.outputname(2)={'DO_2'};
fifth_half=iddata(y7(450:559,:),u7(450:559,:),Ts);%converting 5th half into iddata
fifth_half.inputname(1)={'Ambtemp'};
fifth_half.inputname(2)={'cms'};
fifth_half.timeunit='weeks';
```

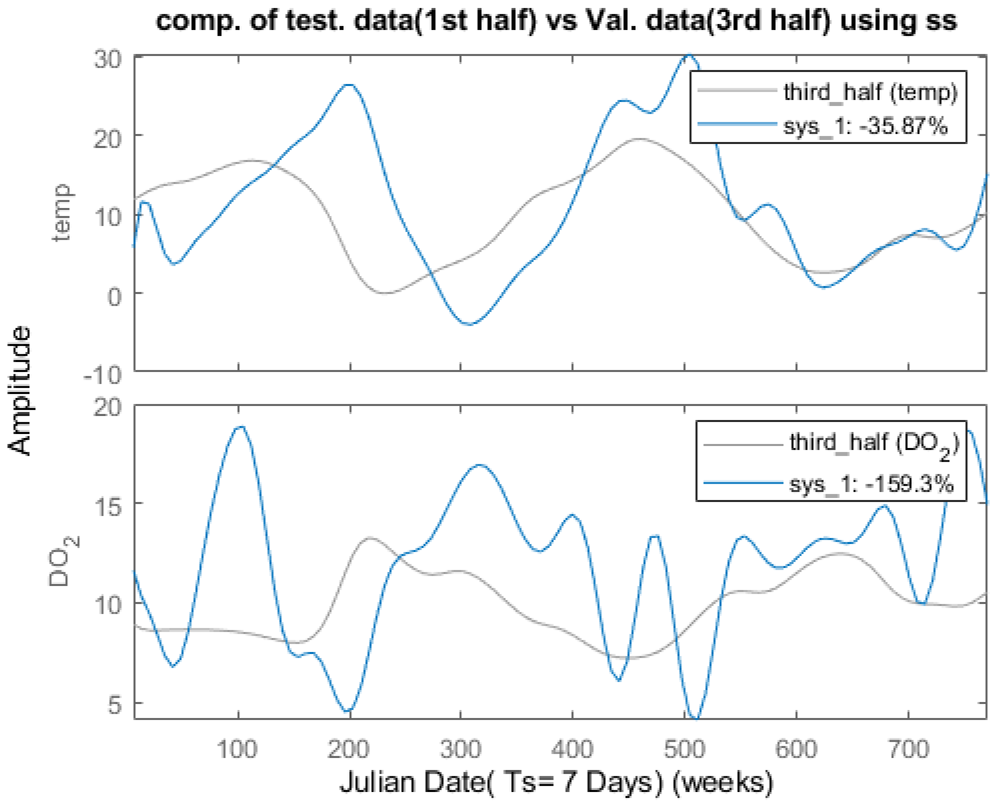
```
fifth_half.outputname(1)={'temp'};
fifth_half.outputname(2)={'DO_2'};
sixth_half=iddata(y7(560:650,:),u7(560:650,:),Ts);%converting 6th half into iddata
sixth_half.inputname(1)={'Ambtemp'};
sixth_half.inputname(2)={'cms'};
sixth_half.timeunit='weeks';
sixth_half.outputname(1)={'temp'};
sixth_half.outputname(2)={'DO_2'};
sys_1=n4sid((first_half));%using n4sid to 1st half
sys_2=n4sid((second_half));%using n4sid to 2nd half
sys_3=n4sid((third_half));%using n4sid to 3rd half
sys_4=n4sid((fourth_half));%using n4sid to 4th half
sys_5=n4sid((fifth_half));%using n4sid to 5th half
sys_6=n4sid((sixth_half));%using n4sid to 6th half
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure()
compare(sys_1,fourth_half)
title('comp. of test. data(1st half) vs Val. data(4th half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure ()
compare(sys_1,fifth_half)
title('comp. of test. data(1st half) vs Val. data(5th half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure ()
compare(sys_1,sixth_half)
title('comp. of test. data(1st half) vs Val. data(6th half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
```

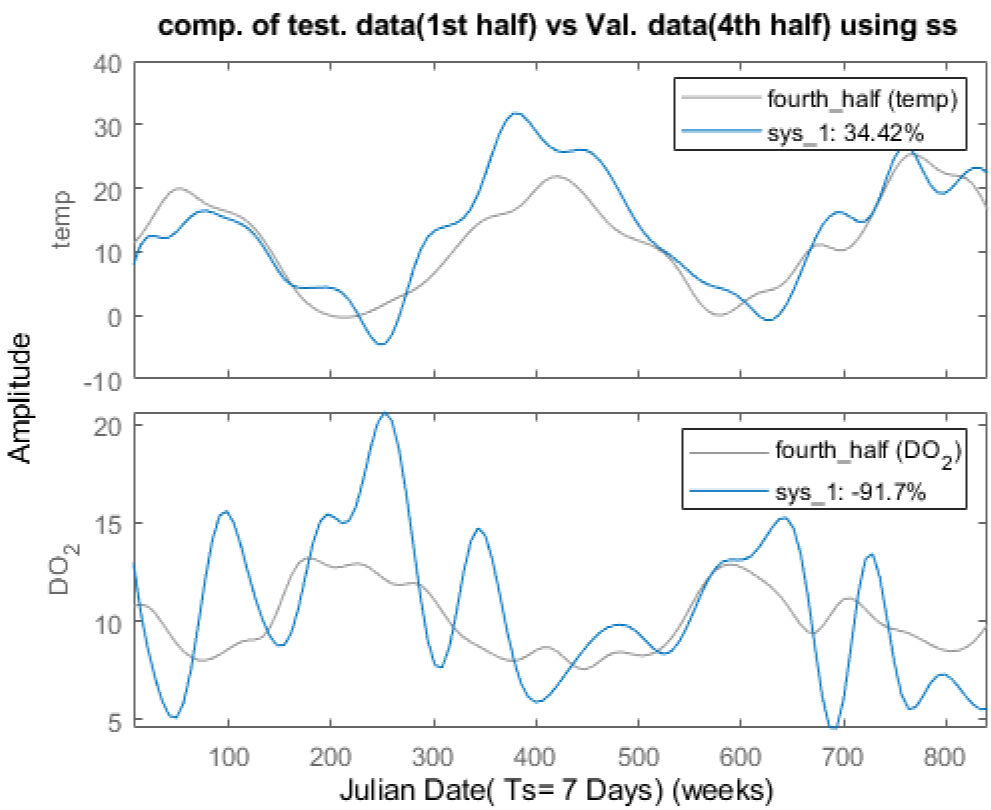
```
sys =
  Discrete-time identified state-space model:
    x(t+Ts) = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
              x1         x2         x3         x4
   x1      1.008     0.0207    0.09254      0.134
   x2    -0.0609     0.8615    -0.2484     0.1292
   x3   -0.06836     0.1421     0.8713    -0.1812
   x4   -0.09684    -0.1082   -0.04761     0.3351

  B =
```

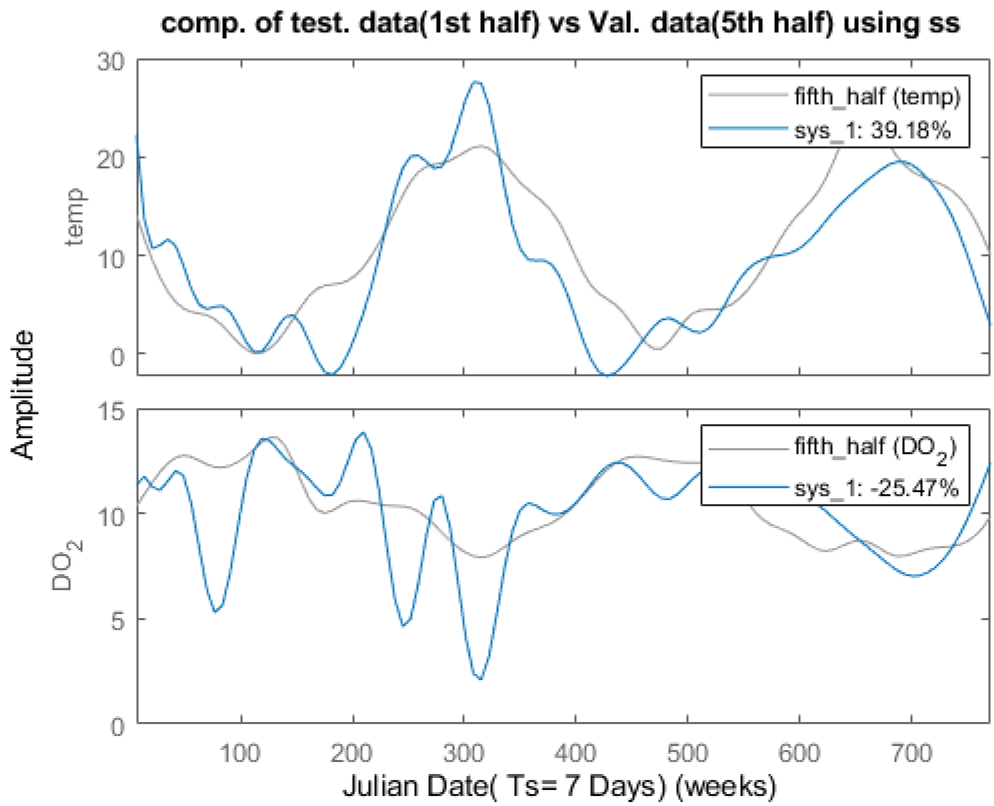
```
         Ambtemp          cms
   x1   4.461e-06   -0.001402
   x2   0.0001649   -0.003551
   x3  -0.0007218    0.002016
   x4    0.001144     0.01303

  C =
            x1      x2      x3      x4
   temp  -23.84   54.91  -8.118   2.112
   DO_2   55.89  -3.473     2.8   2.436

  D =
        Ambtemp      cms
   temp        0        0
   DO_2        0        0

  K =
            temp       DO_2
   x1   0.001725     0.0188
   x2    0.01585   0.007255
   x3   -0.02434  -0.004789
   x4   0.006036   0.007623

Sample time: 7 weeks

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 40
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using N4SID on time domain data "weekly".
Fit to estimation data: [92.07;85.65]% (prediction focus)
FPE: 0.01831, MSE: 0.3666
```

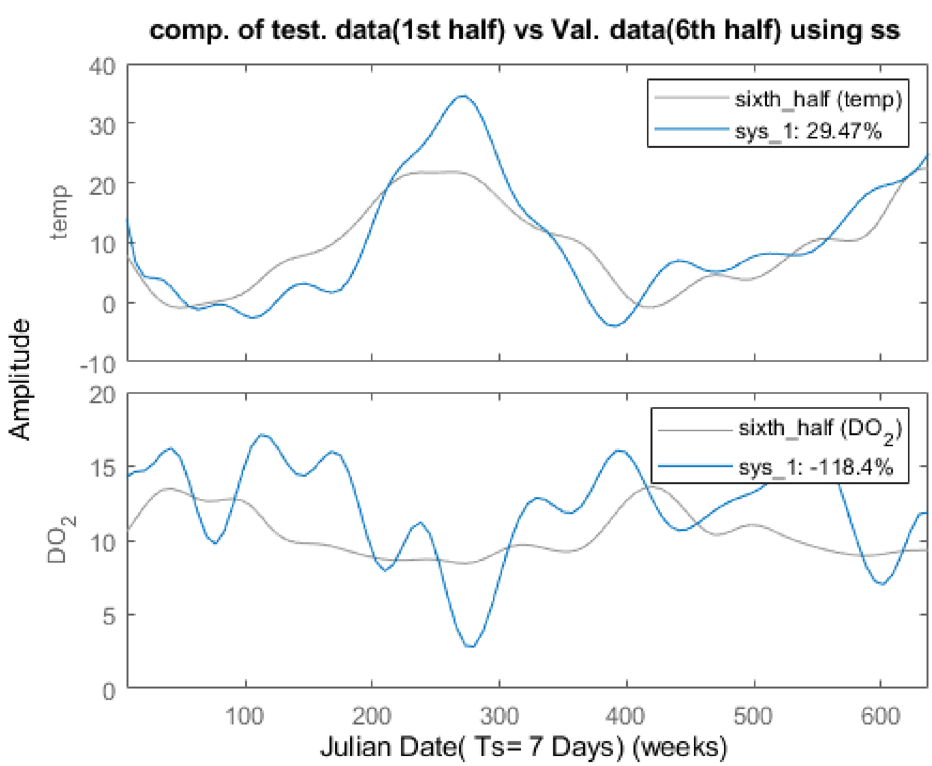

comp. of test. data(1st half) vs test. data using ss
first_half (temp)
sys_1: 68.73%
first_half (DO_2)
sys_1: 43.06%
Amplitude
temp
DO_2
Julian Date(Ts= 7 Days) (weeks)


comp. of test. data(1st half) vs Val. data(2nd half) using ss
second_half (temp)
sys_1: 29.8%
second_half (DO_2)
sys_1: -29.11%
Amplitude
temp
DO_2
Julian Date(Ts= 7 Days) (weeks)


comp. of test. data(1st half) vs Val. data(3rd half) using ss
third_half (temp)
sys_1: -35.87%
third_half (DO2)
sys_1: -159.3%
Amplitude
temp
DO2
Julian Date(Ts= 7 Days) (weeks)


comp. of test. data(1st half) vs Val. data(4th half) using ss
fourth_half (temp)
sys_1: 34.42%
fourth_half (DO2)
sys_1: -91.7%
Amplitude
temp
DO2
Julian Date(Ts= 7 Days) (weeks)





Published with MATLAB® R2020b

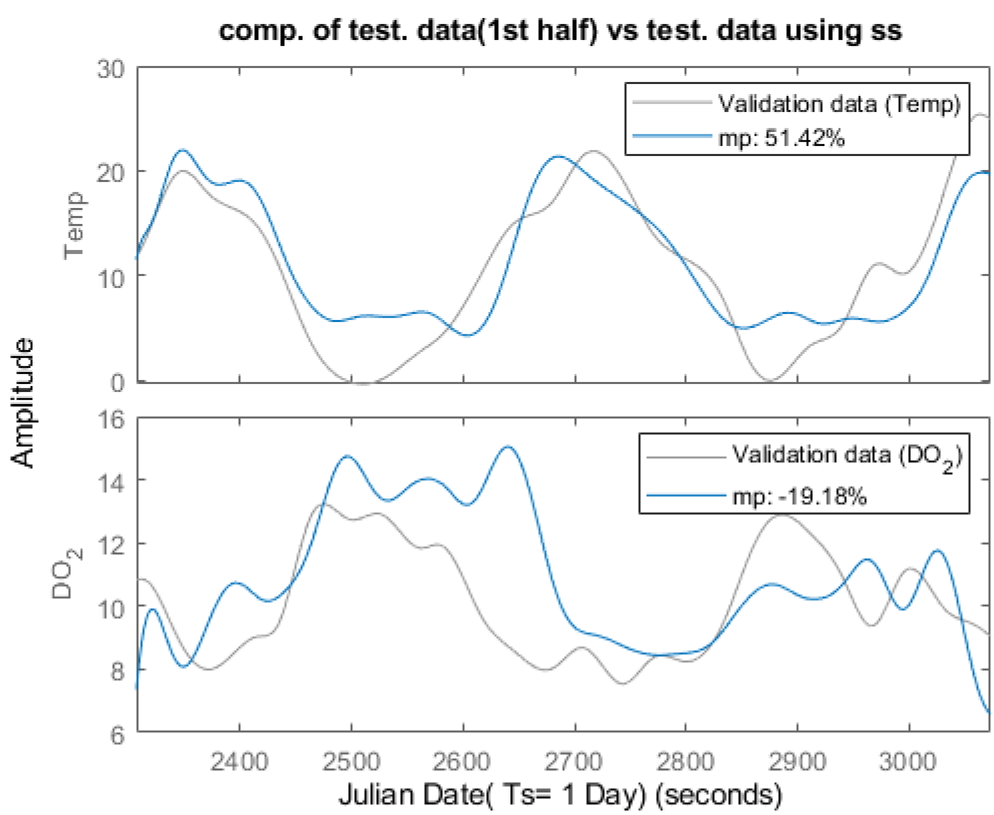
2. Data Into Six Portions Using SSEST

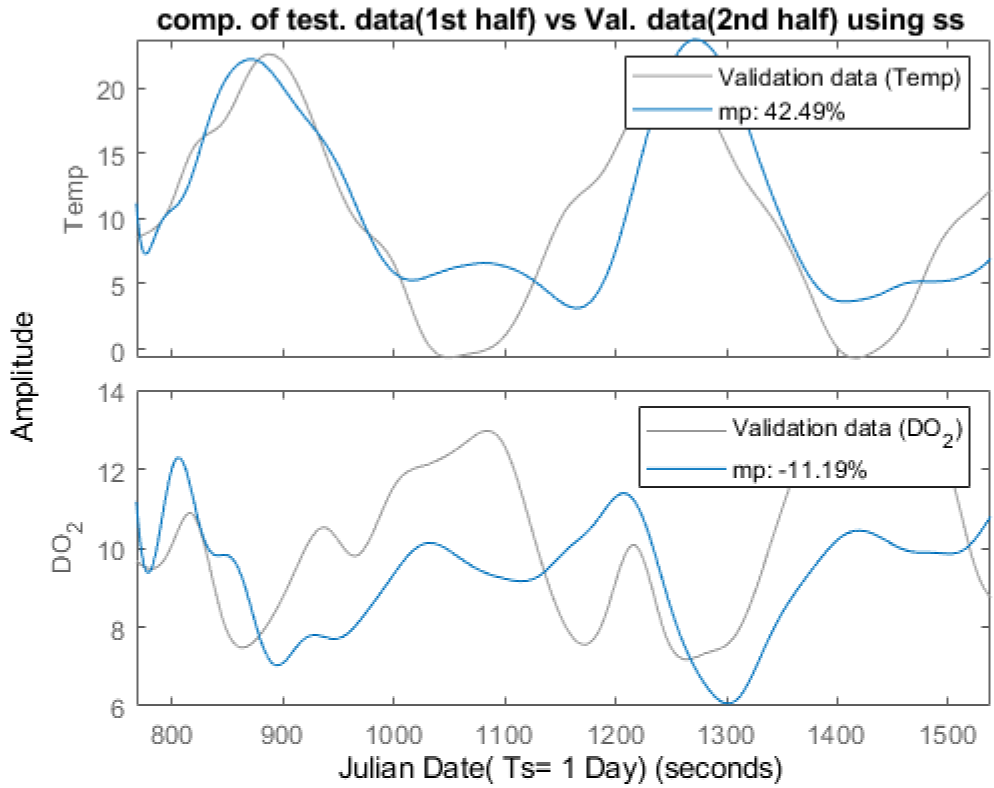
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset.
%It will compare the result on the basis of daily data excluding three years using state space
%for the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%ss. = state space
close all;
clc;
clear;
load fichter_excluding_three_years_daily.mat;%load fichter data excluding three years
Daily=iddata([Temp,DO_2],[Ambtemp_time,cms],Ts);%converting inputs and outputs into iddata
Daily.InputName={'Ambtemp_time','cms'};%giving name to input of the system
Daily.Outputname={'Temp';'DO_2'};%giving name to output of the system
daily.timeunit='days';%time unit
mp=ssest(Daily(1:768))% using ssest to 1st half
figure(7)
compare(Daily(2307:3073),mp);
title('comp. of test. data(1st half) vs test. data using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure(8)
compare(Daily(768:1538),mp);
title('comp. of test. data(1st half) vs Val. data(2nd half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure(9)
compare(Daily(1539:2306),mp);
title('comp. of test. data(1st half) vs Val. data(3rd half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure(10)
compare(Daily(2307:3073),mp);
title('comp. of test. data(1st half) vs Val. data(4th half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure(11)
compare(Daily(3074:3841),mp);
title('comp. of test. data(1st half) vs Val. data(5th half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
figure(12)
compare(Daily(3842:4548),mp);
title('comp. of test. data(1st half) vs Val. data(6th half) using ss')
xlabel('Julian Date( Ts= 1 Day)')
```

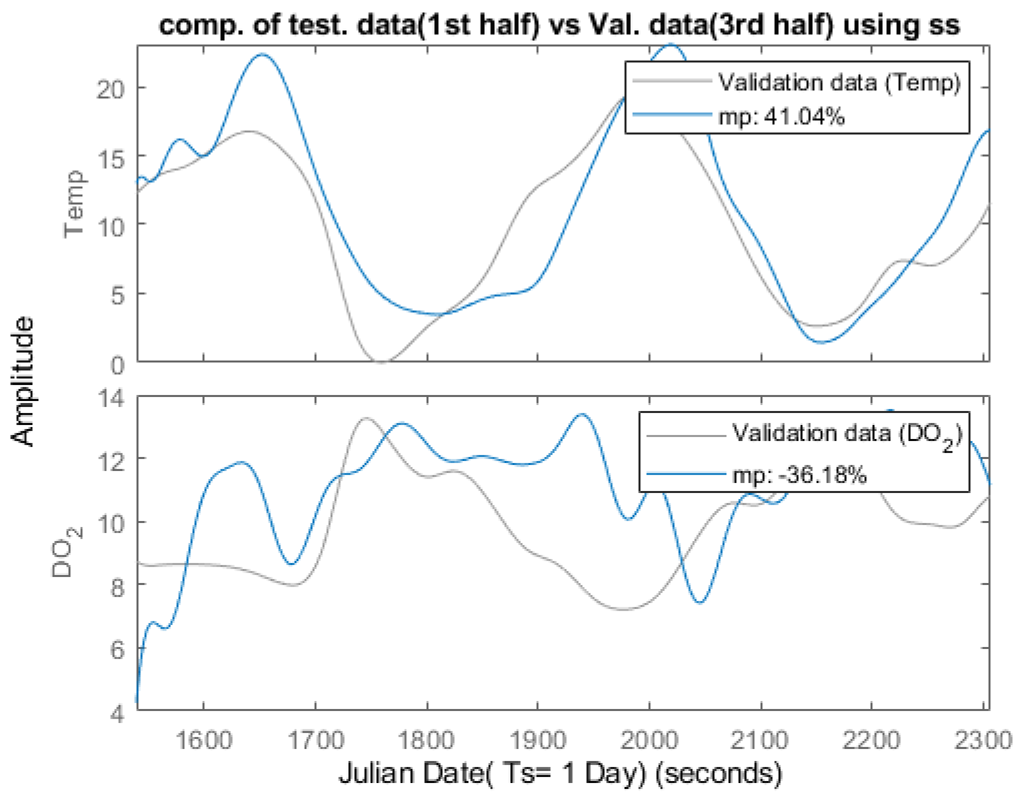
```
mp =
```

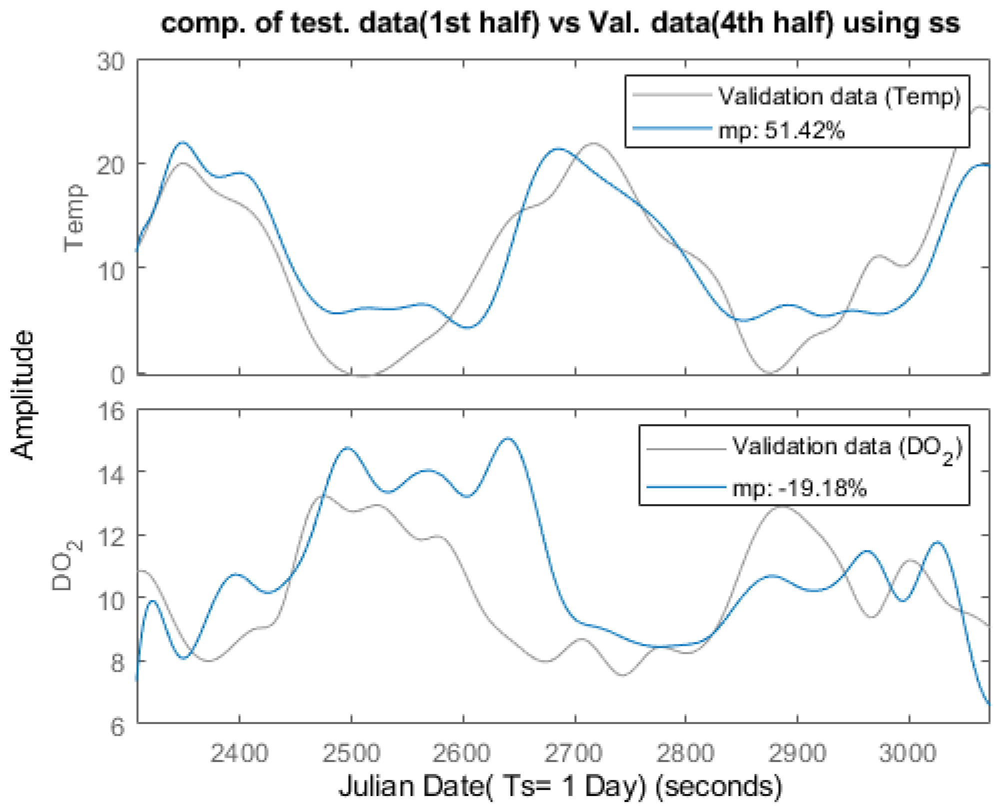
```
Continuous-time identified state-space model:
    dx/dt = A x(t) + B u(t) + K e(t)
     y(t) = C x(t) + D u(t) + e(t)

A =
             x1          x2          x3          x4          x5          x6
 x1     0.02546     0.03122     -0.1581    0.004597     0.04881    -0.01266
 x2      0.1484     0.03596    -0.02406    -0.09204    -0.03357     0.03261
 x3     0.05428     0.06247    -0.02072      0.1109     -0.1215    -0.01681
 x4     0.03738     0.01894   -0.002605     0.02509    -0.04289     -0.1602
 x5      0.0806     0.06845     0.03643     0.02151    -0.04438    -0.03726
 x6    -0.04279   -0.001795     0.02678     0.03529   -0.006316     0.02673
 x7     0.02746      0.0392   -0.005373    -0.04057     0.07131     0.09824
 x8    0.006775    -0.03617    -0.04576    -0.03449     0.06218     -0.1741

             x7          x8
 x1    0.002714     0.01721
 x2     0.05222     -0.0426
 x3    0.009611    -0.01646
 x4    -0.02904     0.09084
 x5     -0.1585     0.09922
 x6  -0.0005907     0.05735
 x7     -0.1655     0.05825
 x8     -0.1033     -0.1244

B =
     Ambtemp_time           cms
 x1    -2.339e-08    -0.0007568
 x2    -1.627e-07     -0.001231
 x3     -2.85e-06     -0.001585
 x4    -7.399e-06     -0.002348
 x5    -1.619e-05     -0.003221
 x6     3.112e-06    -4.598e-05
 x7    -5.539e-06     -0.000909
 x8     2.425e-05      0.001144

C =
              x1          x2          x3          x4          x5          x6
 Temp      -5.97        51.4    -0.08201     -0.9863   -0.007259     0.04035
 DO_2     -40.24       -17.1      0.5615      0.3189    -0.01918    -0.01265

              x7          x8
 Temp  0.0006352    0.003135
 DO_2   0.004261    0.005541

D =
       Ambtemp_time           cms
 Temp             0             0
 DO_2             0             0

K =
          Temp       DO_2
 x1   -0.01316   -0.04199
 x2     0.0342  -0.009143
```

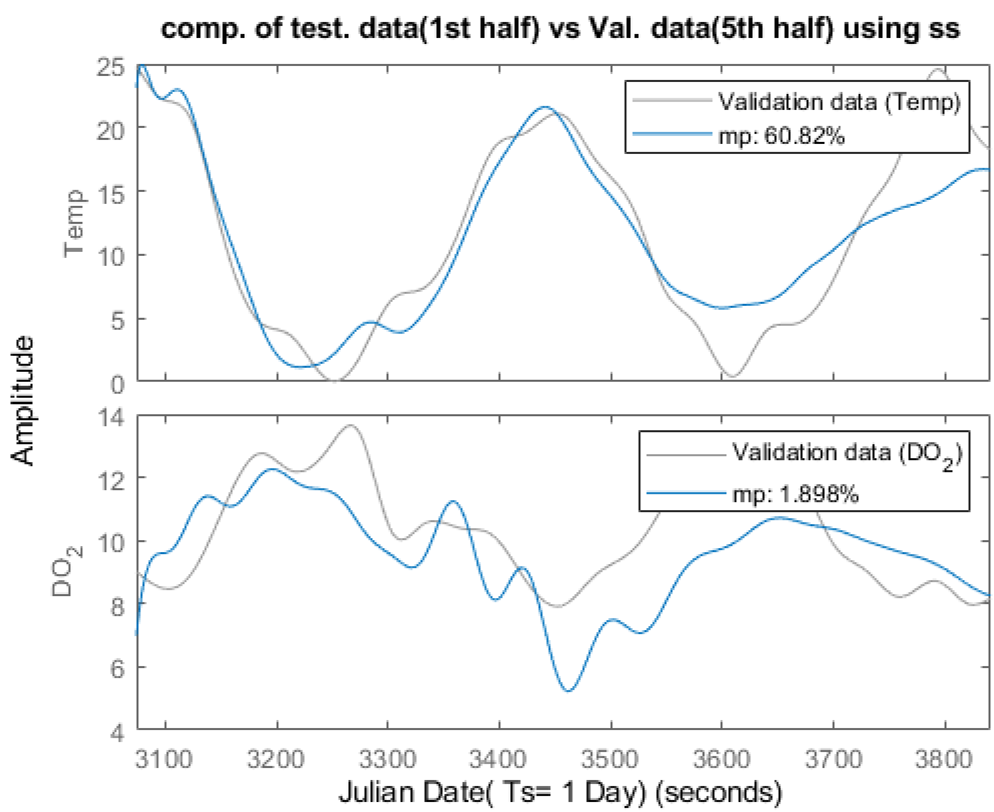
```
    x3      0.1507      0.5018
    x4     -0.3982     -0.1488
    x5       -1.27      -3.342
    x6        2.58      0.2612
    x7       2.249       5.577
    x8       5.091       9.362

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 112
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using SSEST on time domain data.
Fit to estimation data: [100;99.99]% (prediction focus)
FPE: 7.118e-16, MSE: 1.082e-07
```

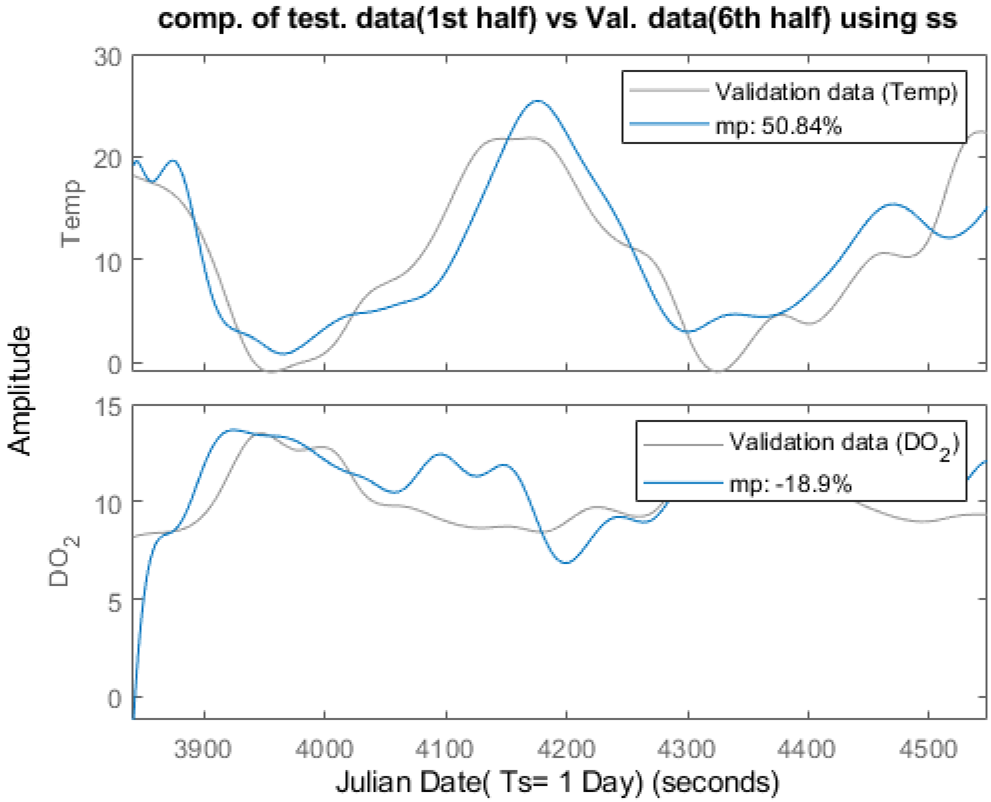



comp. of test. data(1st half) vs Val. data(2nd half) using ss
Validation data (Temp)
mp: 42.49%
Validation data (DO2)
mp: -11.19%
Amplitude
Temp
DO2
Julian Date(Ts= 1 Day) (seconds)


comp. of test. data(1st half) vs Val. data(3rd half) using ss
Validation data (Temp)
mp: 41.04%
Validation data (DO2)
mp: -36.18%
Amplitude
Temp
DO2
Julian Date(Ts= 1 Day) (seconds)


comp. of test. data(1st half) vs Val. data(4th half) using ss
Validation data (Temp)
mp: 51.42%
Validation data (DO2)
mp: -19.18%
Amplitude
Temp
DO2
Julian Date(Ts= 1 Day) (seconds)


comp. of test. data(1st half) vs Val. data(5th half) using ss
Validation data (Temp)
mp: 60.82%
Validation data (DO2)
mp: 1.898%
Amplitude
Temp
DO2
Julian Date(Ts= 1 Day) (seconds)



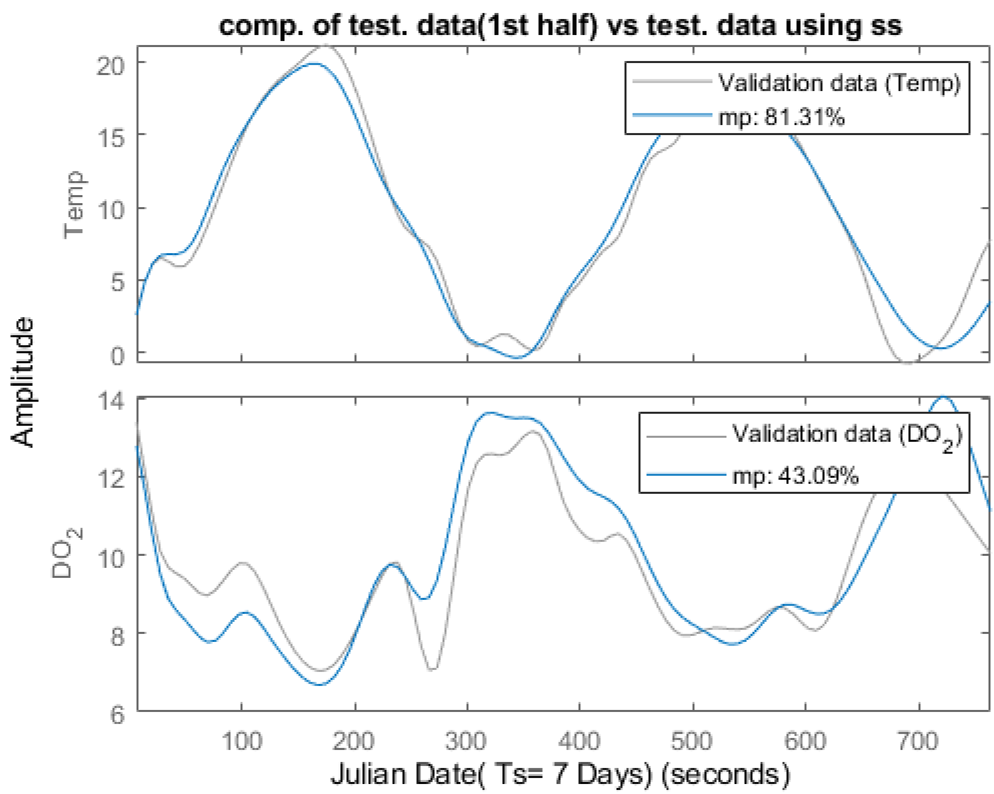
Published with MATLAB® R2020b

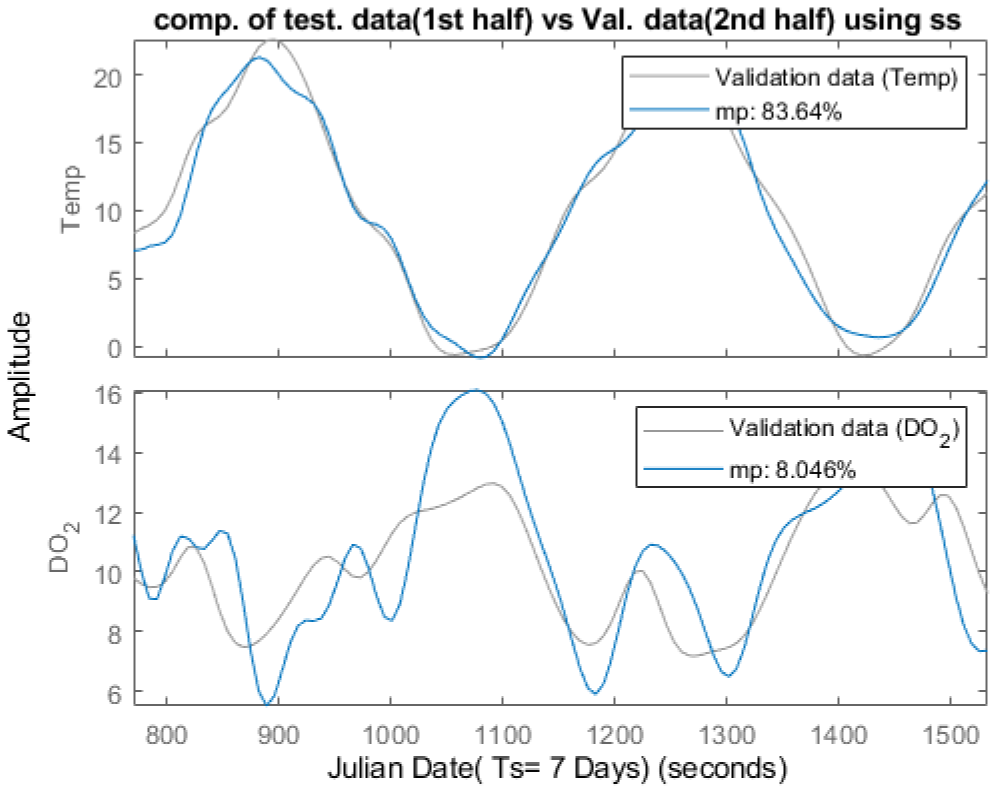
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset.
%It will compare the result on the basis of weekly data excluding three years using state space
%for the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%ss. = state space
close all;
clc;
clear;
load fichter_excluding_three_years_weekly;%load fichter data excluding three years
Daily=iddata([Temp,DO_2],[Ambtemp_time, cms],Ts);%converting inputs and outputs into iddata
Daily.InputName={'Ambtemp_time';'cms'};%giving names to inputs of the system
Daily.Outputname={'Temp';'DO_2'};%giving names to outputs of the system
weekly.timeunit='weeks';%time unit
mp=ssest(Daily(1:109))%using ssest to first half
figure(7)
compare(Daily(1:109),mp);
title('comp. of test. data(1st half) vs test. data using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure(8)
compare(Daily(110:219),mp);
title('comp. of test. data(1st half) vs Val. data(2nd half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure(9)
compare(Daily(220:329),mp);
title('comp. of test. data(1st half) vs Val. data(3rd half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure(10)
compare(Daily(330:449),mp);
title('comp. of test. data(1st half) vs Val. data(4th half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure(11)
compare(Daily(450:559),mp);
title('comp. of test. data(1st half) vs Val. data(5th half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
figure(12)
compare(Daily(560:650),mp);
title('comp. of test. data(1st half) vs Val. data(6th half) using ss')
xlabel('Julian Date( Ts= 7 Days)')
```

```
mp =
  Continuous-time identified state-space model:
      dx/dt = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
                 x1           x2           x3           x4           x5           x6
```

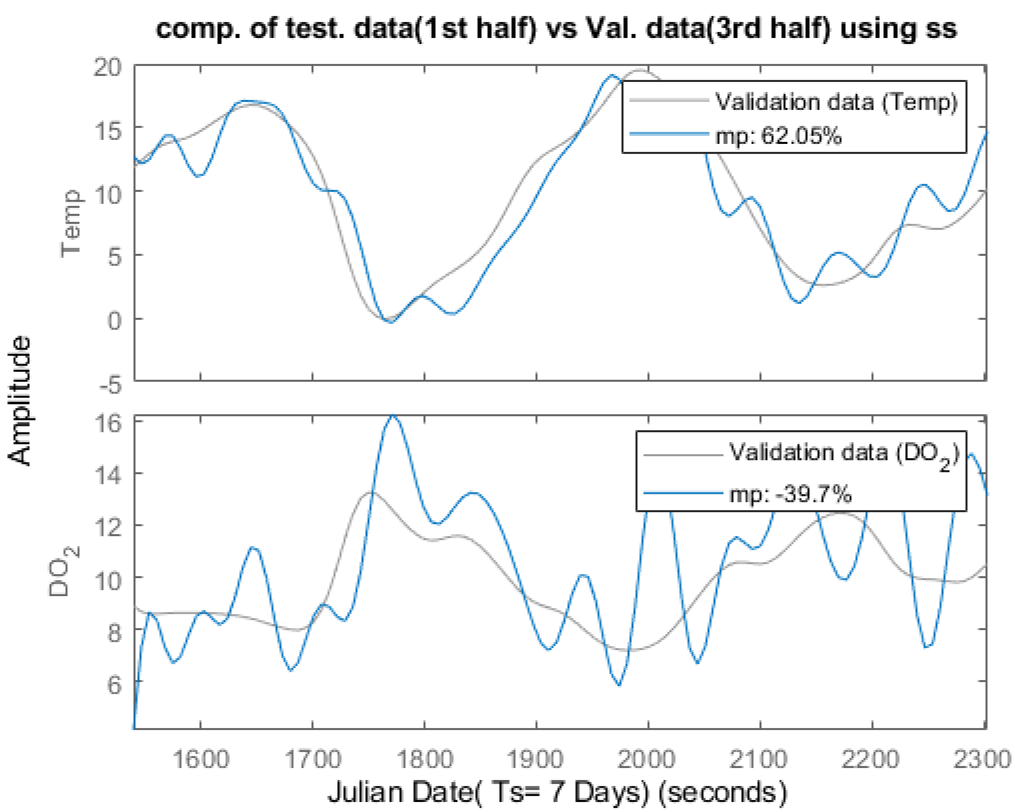
```
 x1     0.02067     0.02171     0.04295    -0.02356    -0.02569    -0.03044
 x2     0.01468    -0.01694     -0.0153    -0.02706    -0.02115     0.01591
 x3    -0.05439     0.02832    -0.01471     -0.0209   0.0002725     0.05506
 x4     0.01198    -0.06764     0.04137   -0.004042     0.01159     0.02816
 x5     0.04217     0.05359    -0.01652    -0.02312   -0.001963     0.01573
 x6     0.01823    -0.02043     -0.0729    -0.01607    -0.03704    -0.01933
 x7  -0.0006069    0.005105    -0.01502    -0.07498    -0.05101    0.008949
 x8   -0.005706    0.005361   -0.005286    -0.01093   -0.002749     0.01743
 x9   -0.004802    -0.01244      0.0231    -0.02959    -0.03731   -0.006821

             x7          x8          x9
 x1   -0.004797    -0.00234    0.003998
 x2     0.03368    -0.02422     0.01078
 x3     0.01675   -0.007351    -0.02897
 x4     0.02148   -0.007435    0.009232
 x5     0.06989    -0.01695     0.03844
 x6    -0.06606    -0.04285     0.03115
 x7    -0.03153     0.03723    -0.06146
 x8    -0.07597    0.006875    0.009832
 x9     0.02554    -0.01755     -0.0169

B =
     Ambtemp_time           cms
 x1     5.636e-06      -0.00357
 x2     7.095e-06     -0.003222
 x3    -8.191e-05       0.01687
 x4      -0.00036      0.002879
 x5     0.0005458      -0.01187
 x6    -0.0003267     0.0009846
 x7     0.0004066       0.02169
 x8      8.73e-05       0.01776
 x9     -5.76e-05      -0.00228

C =
           x1       x2       x3       x4       x5       x6       x7
 Temp    1.634   -15.56   0.5854  -0.9887   0.1182   -0.509   -1.052
 DO_2   -8.849    2.871   -1.146    1.168   0.7248   0.4558   0.2908

           x8       x9
 Temp  0.04251   0.2573
 DO_2  0.05541    0.122

D =
       Ambtemp_time           cms
 Temp             0             0
 DO_2             0             0

K =
          Temp       DO_2
 x1  -0.002376   -0.01432
 x2   -0.01448  -0.006427
 x3     0.0353   -0.04647
 x4    0.01915    0.07926
 x5    0.06771     0.0107
```

```
   x6  -0.006096   -0.02784
   x7   -0.04256   -0.01275
   x8    0.01276    0.06985
   x9    0.04406     0.1517

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 135
   Use "idssdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using SSEST on time domain data.
Fit to estimation data: [98.55;97.43]% (prediction focus)
FPE: 2.502e-05, MSE: 0.01157
```

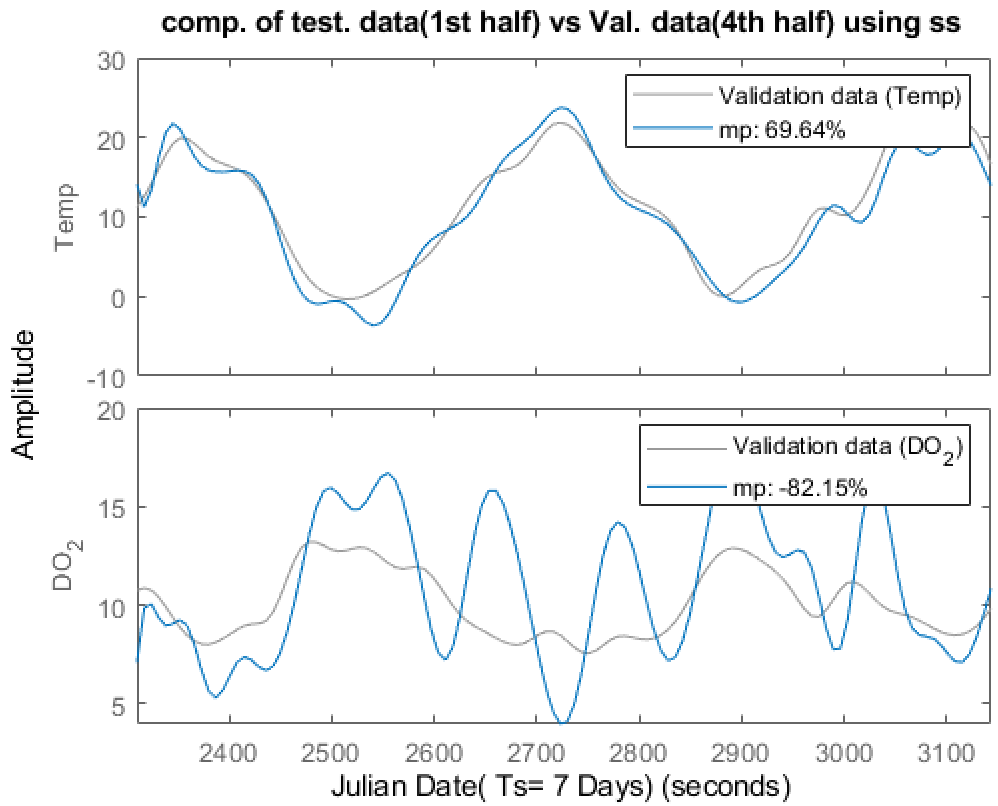


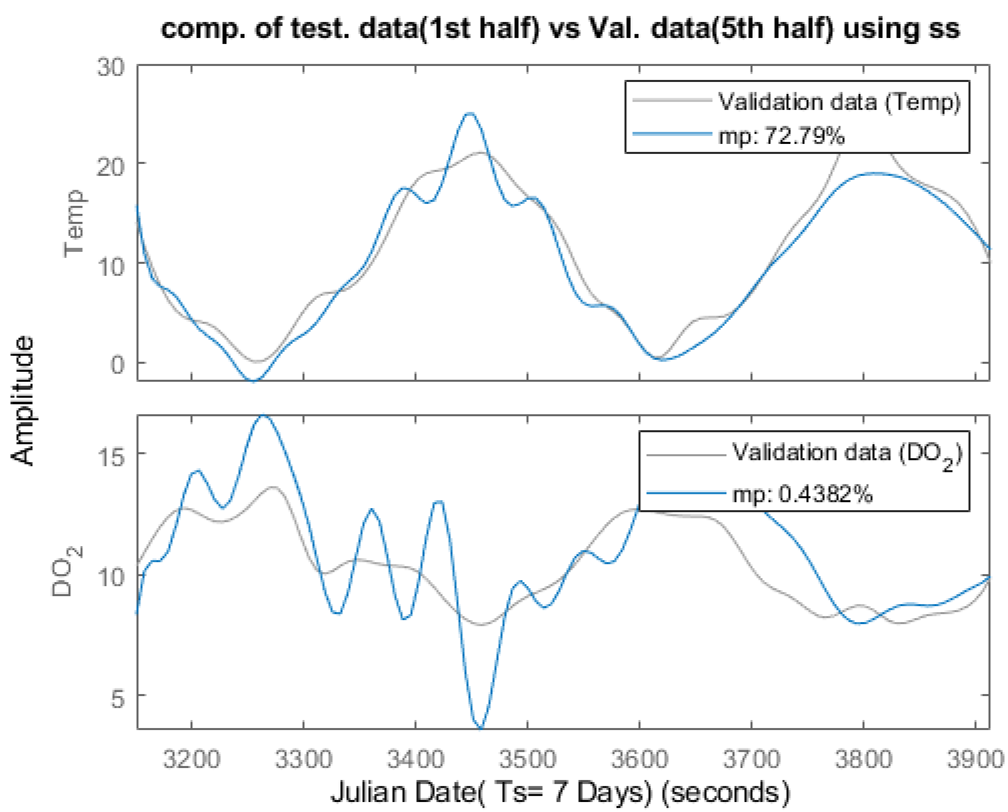


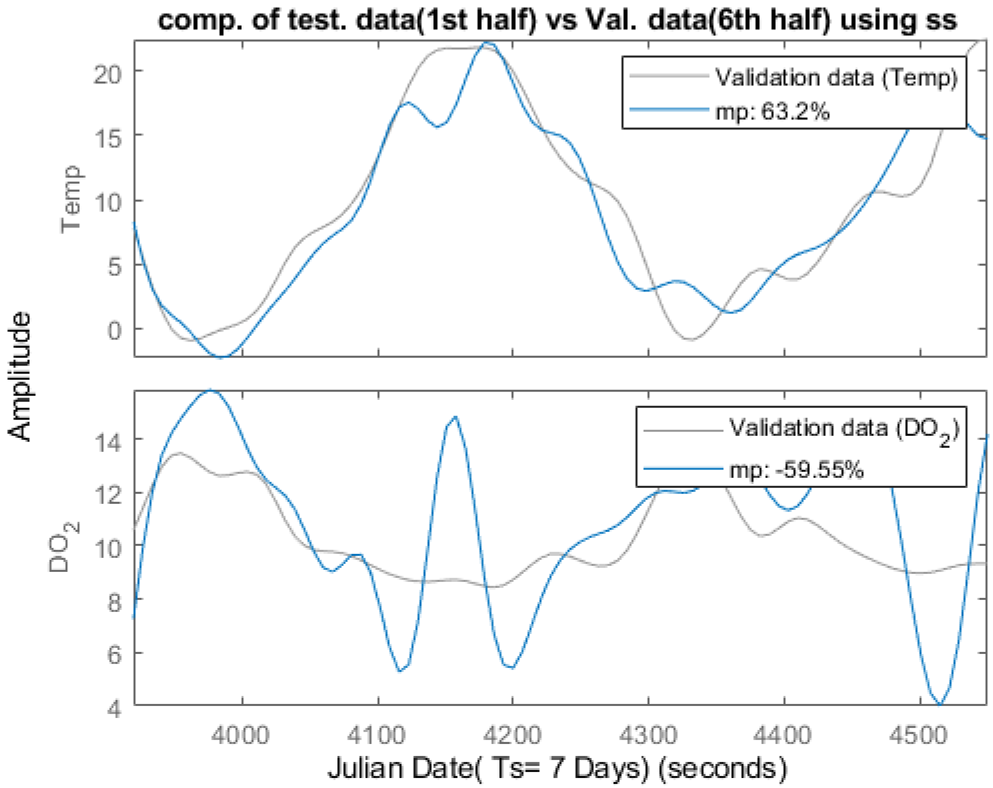
comp. of test. data(1st half) vs Val. data(2nd half) using ss
Validation data (Temp)
mp: 83.64%
Validation data (DO_2)
mp: 8.046%
Temp
DO_2
Amplitude
Julian Date(Ts= 7 Days) (seconds)



comp. of test. data(1st half) vs Val. data(3rd half) using ss
Validation data (Temp)
mp: 62.05%
Validation data (DO_2)
mp: -39.7%
Temp
DO_2
Amplitude
Julian Date(Ts= 7 Days) (seconds)


comp. of test. data(1st half) vs Val. data(4th half) using ss
Validation data (Temp)
mp: 69.64%
Validation data (DO2)
mp: -82.15%
Amplitude
Temp
DO2
Julian Date(Ts= 7 Days) (seconds)


comp. of test. data(1st half) vs Val. data(5th half) using ss
Validation data (Temp)
mp: 72.79%
Validation data (DO2)
mp: 0.4382%
Amplitude
Temp
DO2
Julian Date(Ts= 7 Days) (seconds)



Published with MATLAB® R2020b

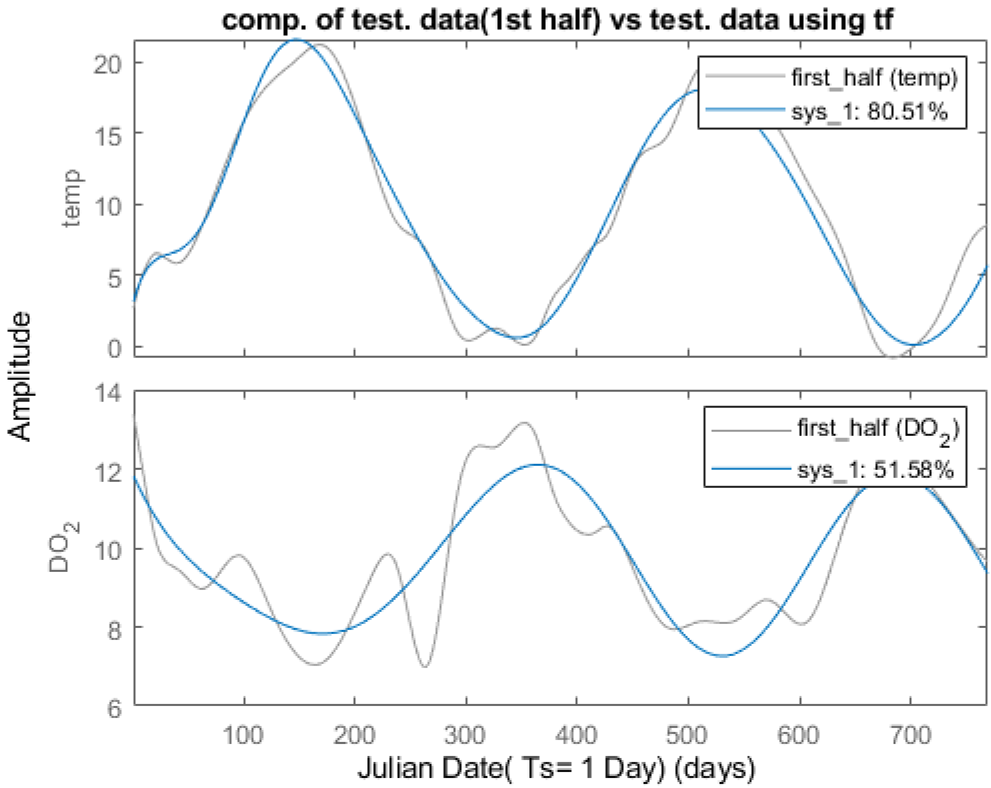
3. Data into Six Portions Using TFEST

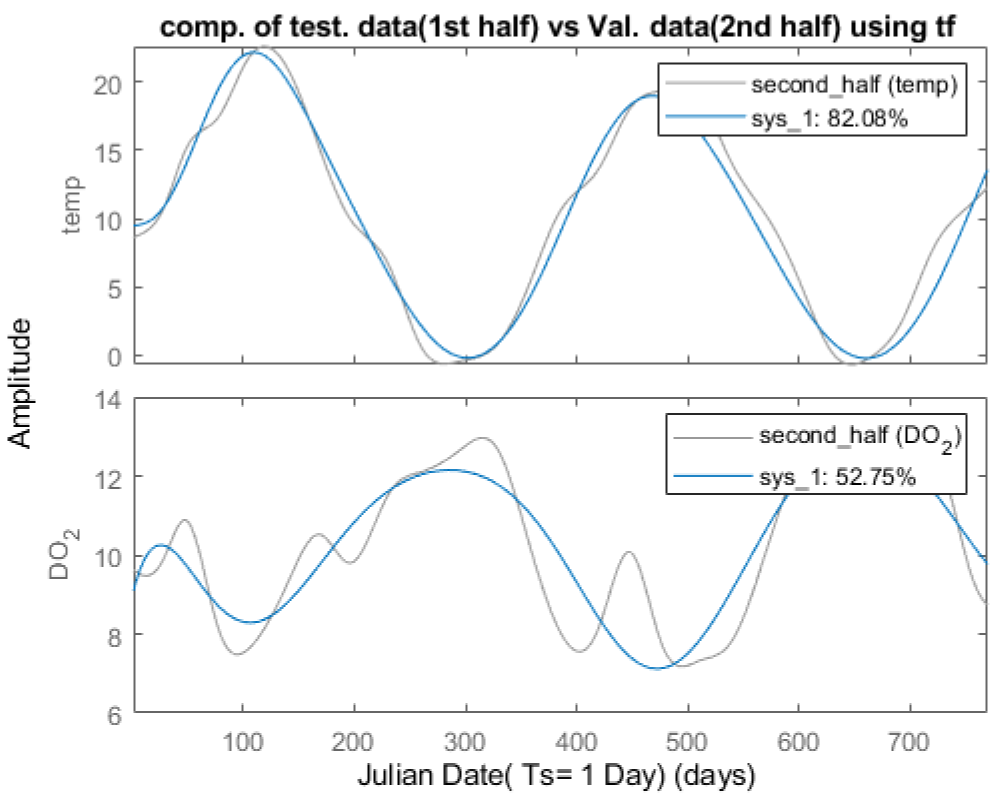
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset.
%It will compare the result on the basis of daily data excluding three years using state space
%for the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%ss. = state space
close all;
clc;
clear;
load fichter_excluding_three_years_daily.mat;%load fichter data excluding three years
u=[Ambtemp_time,cms];%load inputs of the system
y=[Temp,DO_2];%load outputs of the system
daily=iddata(y,u,Ts);%converting inputs and outputs into iddata
%giving names to inputs and outputs of the system
daily.inputname(1)={'Ambtemp'};
daily.inputname(2)={'cms'};
daily.timeunit='days';
daily.outputname(1)={'temp'};
daily.outputname(2)={'DO_2'}
sys=tfest(daily,4,0)
first_half= iddata(y(1:769,:),u(1:769,:),Ts);%converting 1st half into iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='days';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y(770:1539,:),u(770:1539,:),Ts);%converting 2nd half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='days';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y(1540:2308,:),u(1540:2308,:),Ts);%converting 3rd half into iddata
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='days';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
fourth_half=iddata(y(2309:3078,:),u(2309:3078,:),Ts);%converting 4th half into iddata
fourth_half.inputname(1)={'Ambtemp'};
fourth_half.inputname(2)={'cms'};
fourth_half.timeunit='days';
fourth_half.outputname(1)={'temp'};
fourth_half.outputname(2)={'DO_2'};
fifth_half=iddata(y(3079:3848,:),u(3079:3848,:),Ts);%converting 5th half into iddata
```

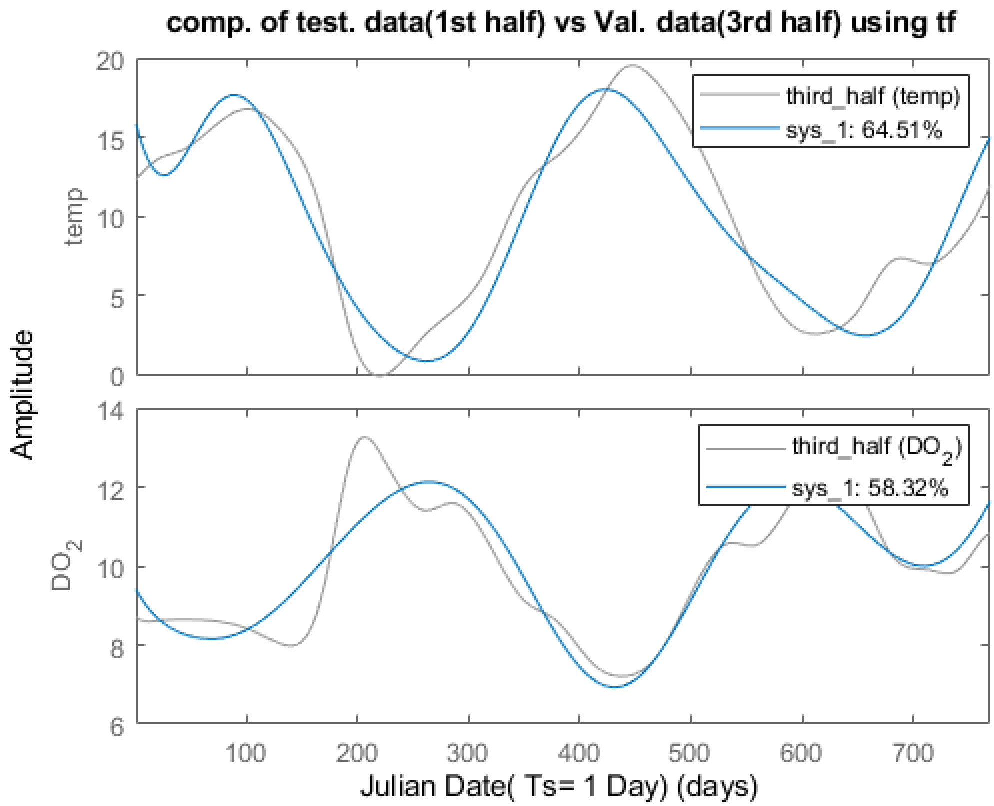
```
fifth_half.inputname(1)={'Ambtemp'};
fifth_half.inputname(2)={'cms'};
fifth_half.timeunit='days';
fifth_half.outputname(1)={'temp'};
fifth_half.outputname(2)={'DO_2'};
sixth_half=iddata(y(3849:4548,:),u(3849:4548,:),Ts);%converting 6th half into iddata
sixth_half.inputname(1)={'Ambtemp'};
sixth_half.inputname(2)={'cms'};
sixth_half.timeunit='days';
sixth_half.outputname(1)={'temp'};
sixth_half.outputname(2)={'DO_2'};
sys_1=tfest(first_half,4,0);% using tfest to 1st half
sys_2=tfest(second_half,4,0);%using tfest to 2nd half
sys_3=tfest(third_half,4,0);%using tfest to 3rd half
sys_4=tfest(fourth_half,4,0);%using tfest to 4th half
sys_5=tfest(fifth_half,4,0);%using tfest to 5th half
sys_6=tfest(sixth_half,4,0);%using tfest to 6th half
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using tf')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using tf')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) using tf')
xlabel('Julian Date( Ts= 1 Day)')
figure()
compare(sys_1,fourth_half)
title('comp. of test. data(1st half) vs Val. data(4th half) using tf')
xlabel('Julian Date( Ts= 1 Day)')
figure ()
compare(sys_1,fifth_half)
title('comp. of test. data(1st half) vs Val. data(5th half) using tf')
xlabel('Julian Date( Ts= 1 Day)')
figure ()
compare(sys_1,sixth_half)
title('comp. of test. data(1st half) vs Val. data(6th half) using tf')
xlabel('Julian Date( Ts= 1 Day)')
```

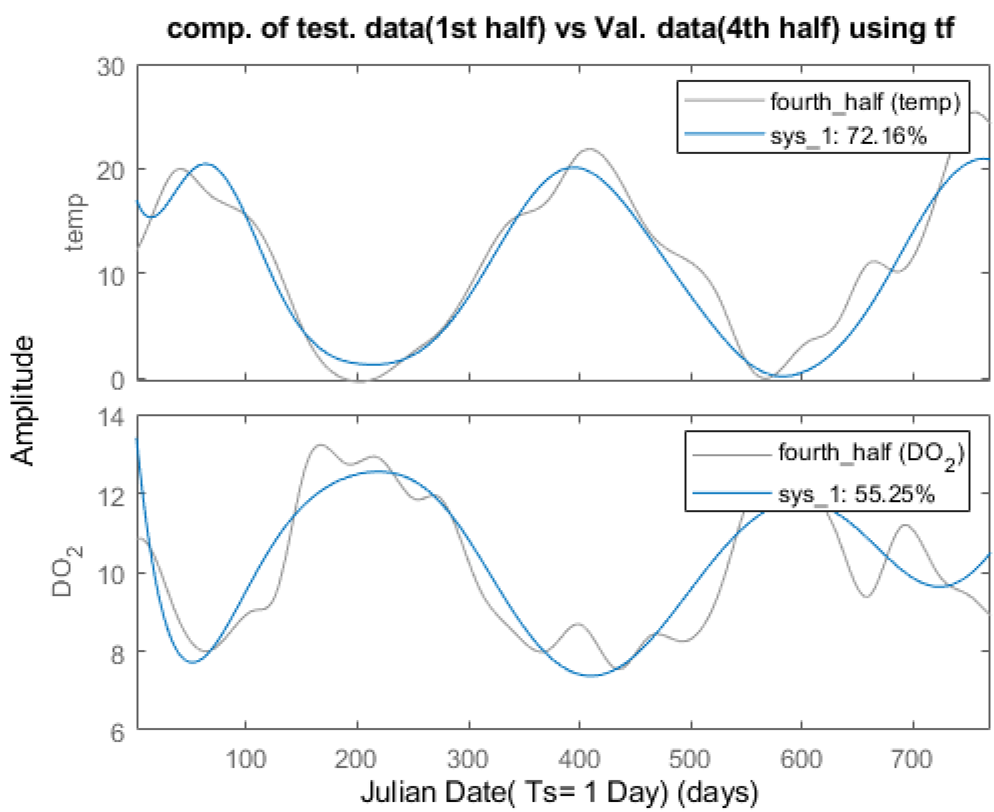
```
daily =

Time domain dataset with 4548 samples.
Sample time: 1 days

Outputs       Unit (if specified)
   temp
   DO_2

Inputs        Unit (if specified)
```

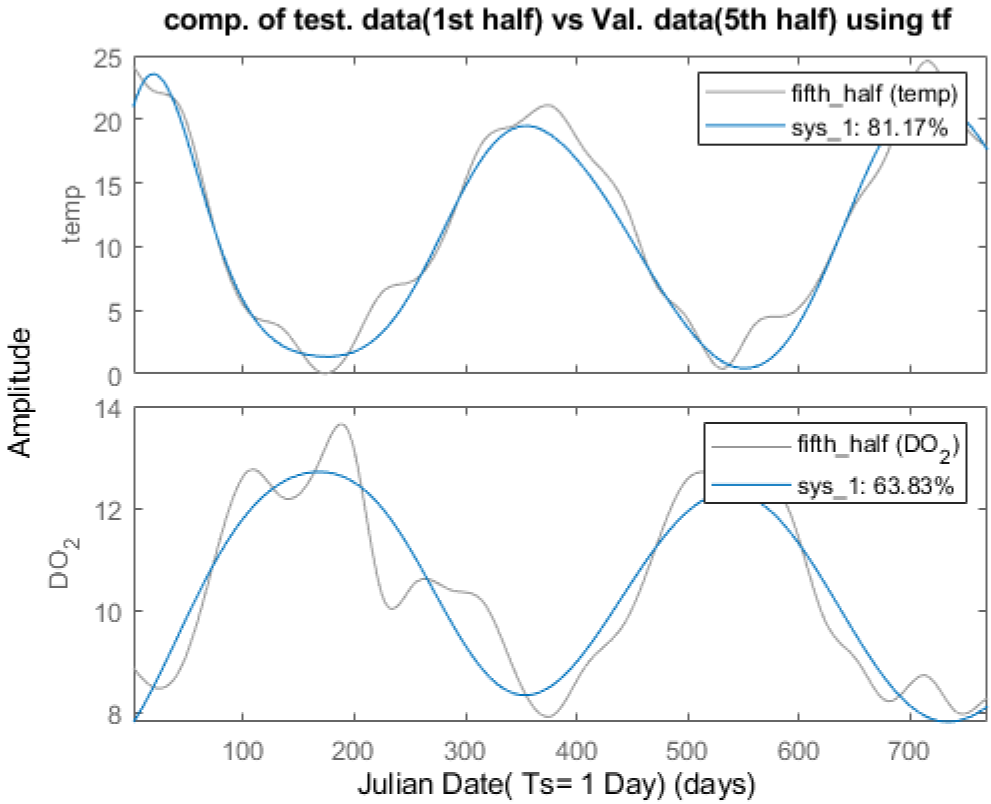
```
   Ambtemp
   cms


sys =

  From input "Ambtemp" to output...
                                 7.942e-09
   temp:  ----------------------------------------------------------
          s^4 + 0.05495 s^3 + 0.0003844 s^2 + 1.581e-05 s + 2.231e-08

                                -2.603e-11
   DO_2:  ----------------------------------------------------------
          s^4 + 0.001599 s^3 + 2.08e-05 s^2 + 9.502e-09 s + 4.246e-12

  From input "cms" to output...
                                 -1.019e-09
   temp:  -----------------------------------------------------------
          s^4 + 0.004269 s^3 + 0.0002942 s^2 + 1.233e-06 s + 8.243e-10

                                 4.678e-09
   DO_2:  -----------------------------------------------------------
          s^4 + 0.001729 s^3 + 0.0002826 s^2 + 3.413e-07 s + 1.491e-18

Continuous-time identified transfer function.

Parameterization:
   Number of poles: [4 4;4 4]   Number of zeros: [0 0;0 0]
   Number of free coefficients: 20
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "daily".
Fit to estimation data: [71.78;31.84]%
FPE: 4.757, MSE: 5.248
```

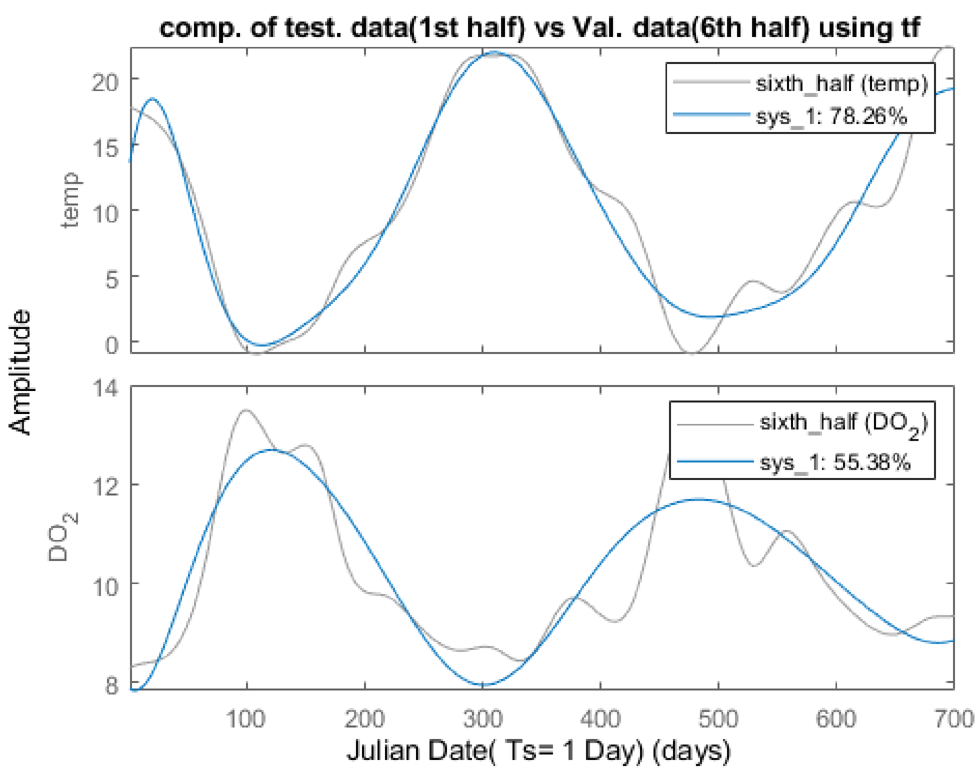

comp. of test. data(1st half) vs test. data using tf
first_half (temp)
sys_1: 80.51%
first_half (DO2)
sys_1: 51.58%
temp
DO2
Amplitude
Julian Date(Ts= 1 Day) (days)
0
5
10
15
20
6
8
10
12
14
100
200
300
400
500
600
700


comp. of test. data(1st half) vs Val. data(2nd half) using tf
second_half (temp)
sys_1: 82.08%
second_half (DO2)
sys_1: 52.75%
temp
DO2
Amplitude
Julian Date(Ts= 1 Day) (days)
0
5
10
15
20
6
8
10
12
14
100
200
300
400
500
600
700


comp. of test. data(1st half) vs Val. data(3rd half) using tf
third_half (temp)
sys_1: 64.51%
third_half (DO_2)
sys_1: 58.32%
Amplitude
temp
DO_2
Julian Date(Ts= 1 Day) (days)


comp. of test. data(1st half) vs Val. data(4th half) using tf
fourth_half (temp)
sys_1: 72.16%
fourth_half (DO_2)
sys_1: 55.25%
Amplitude
temp
DO_2
Julian Date(Ts= 1 Day) (days)





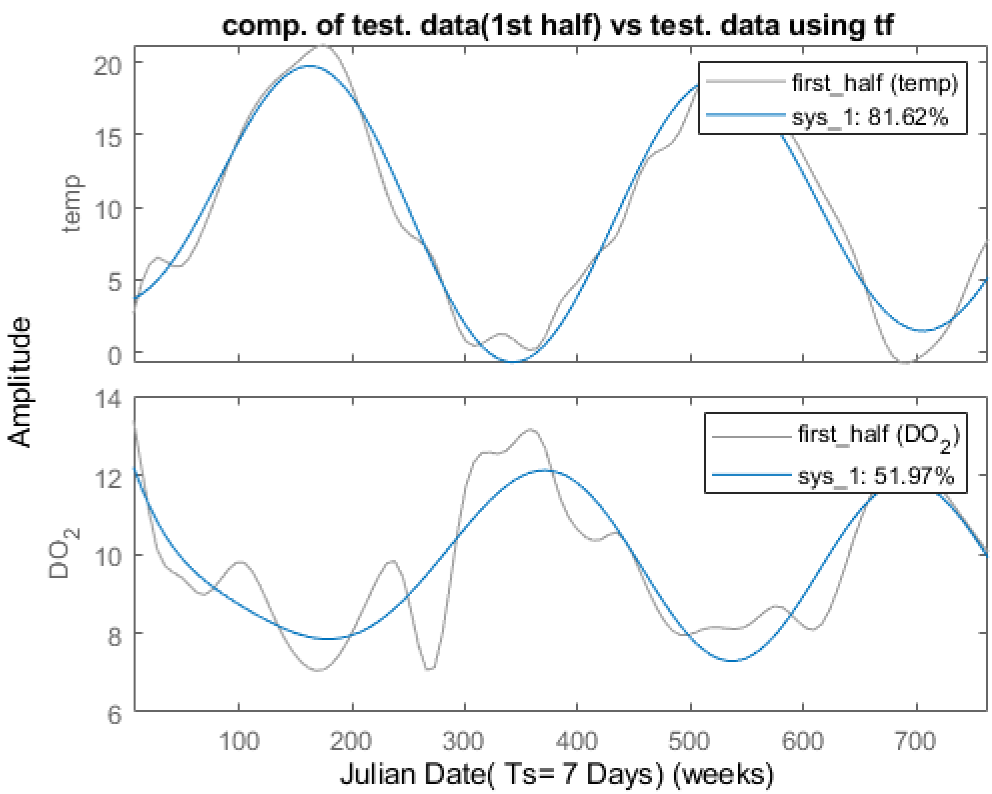
Published with MATLAB® R2020b

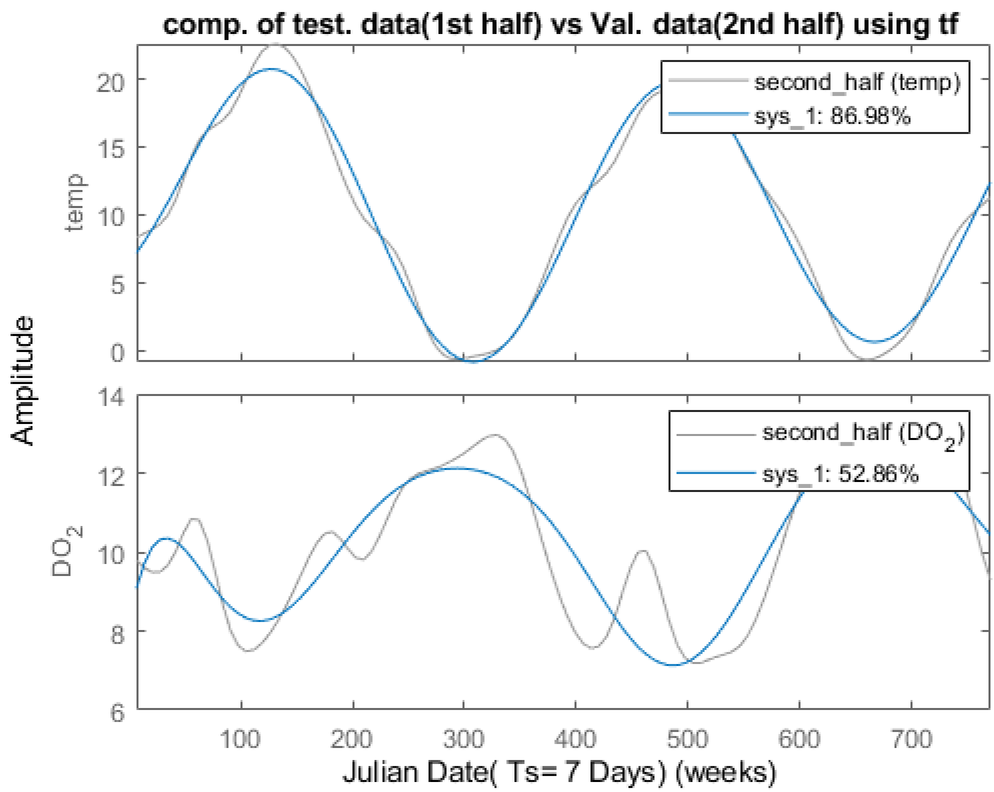
```
% Prasis Timilsena
% Master thesis: System Identification for irregularly spaced data
%this script is for System Identification of fichter water dataset.
%It will compare the result on the basis of weekly data excluding three years using state space
%for the comparision.
%Abbreviated word
%Comp. = Comparision
%Test. = Testing
%val. = Validation
%ss. = state space
close all;
clc;
clear;
load fichter_excluding_three_years_weekly.mat;%load fichter data excluding three years
y7=[Temp,DO_2];%load output of the system
u7=[Ambtemp_time,cms];%load inputs of the system
weekly=iddata(y7,u7,Ts);%converting inputs and outputs into iddata
%giving names to inputs and outputs
weekly.inputname(1)={'Ambtemp'};
weekly.inputname(2)={'cms'};
weekly.timeunit="weeks";
weekly.outputname(1)={'temp'};
weekly.outputname(2)={'DO_2'};
sys=tfest((weekly),4,0)
first_half= iddata(y7(1:109,:),u7(1:109,:),Ts);% converting 1st half into iddata
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='weeks';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y7(110:219,:),u7(110:219,:),Ts);%converting 2nd half into iddata
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='weeks';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
third_half=iddata(y7(220:329,:),u7(220:329,:),Ts);%converting 3rd half into iddata
third_half.inputname(1)={'Ambtemp'};
third_half.inputname(2)={'cms'};
third_half.timeunit='weeks';
third_half.outputname(1)={'temp'};
third_half.outputname(2)={'DO_2'};
fourth_half=iddata(y7(330:449,:),u7(330:449,:),Ts);%converting 4th half into iddata
fourth_half.inputname(1)={'Ambtemp'};
fourth_half.inputname(2)={'cms'};
fourth_half.timeunit='weeks';
fourth_half.outputname(1)={'temp'};
fourth_half.outputname(2)={'DO_2'};
fifth_half=iddata(y7(450:559,:),u7(450:559,:),Ts);%converting 5th half into iddata
fifth_half.inputname(1)={'Ambtemp'};
fifth_half.inputname(2)={'cms'};
fifth_half.timeunit='weeks';
```

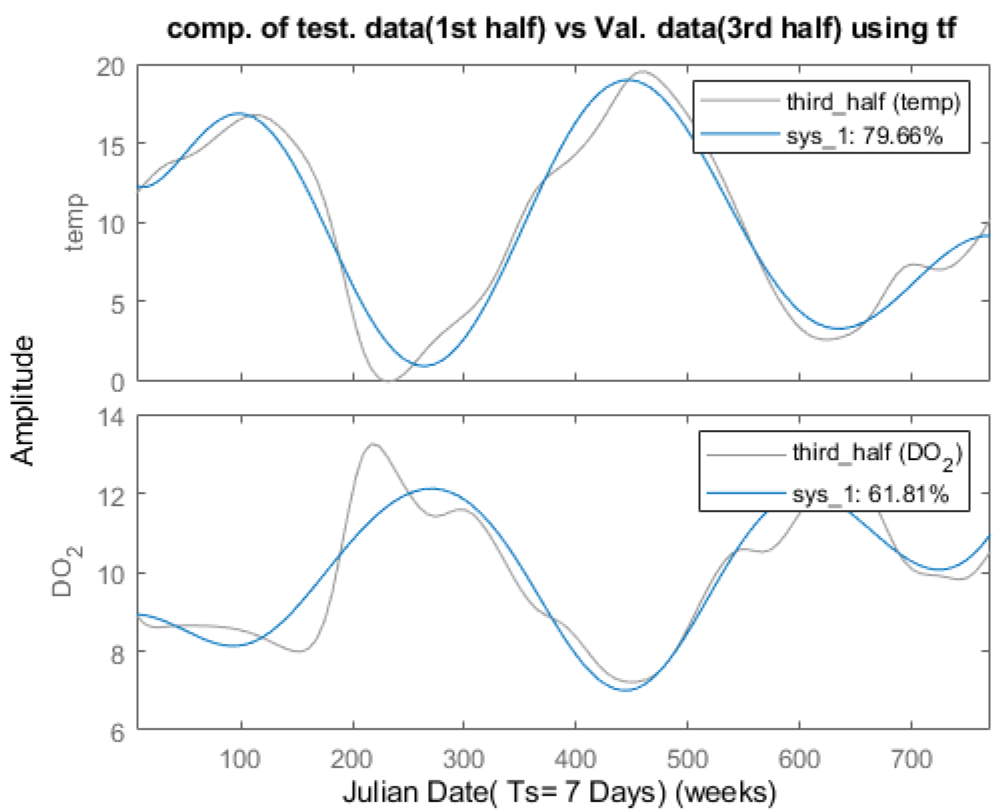
```
fifth_half.outputname(1)={'temp'};
fifth_half.outputname(2)={'DO_2'};
sixth_half=iddata(y7(560:650,:),u7(560:650,:),Ts);%converting 6th half into iddata
sixth_half.inputname(1)={'Ambtemp'};
sixth_half.inputname(2)={'cms'};
sixth_half.timeunit='weeks';
sixth_half.outputname(1)={'temp'};
sixth_half.outputname(2)={'DO_2'};
sys_1=tfest((first_half),4,0);%using tfest to 1st half
sys_2=tfest((second_half),4,0);%using tfest to 2nd half
sys_3=tfest((third_half),4,0);%using tfest to 3rd half
sys_4=tfest((fourth_half),4,0);%using tfest to 4th half
sys_5=tfest((fifth_half),4,0);%using tfest to 5th half
sys_6=tfest((sixth_half),4,0);%using tfest to 6th half
figure()
compare(sys_1,first_half)
title('comp. of test. data(1st half) vs test. data using tf')
xlabel('Julian Date( Ts= 7 Days)')
figure()
compare(sys_1,second_half)
title('comp. of test. data(1st half) vs Val. data(2nd half) using tf')
xlabel('Julian Date( Ts= 7 Days)')
figure()
compare(sys_1,third_half)
title('comp. of test. data(1st half) vs Val. data(3rd half) using tf')
xlabel('Julian Date( Ts= 7 Days)')
figure()
compare(sys_1,fourth_half)
title('comp. of test. data(1st half) vs Val. data(4th half) using tf')
xlabel('Julian Date( Ts= 7 Days)')
figure ()
compare(sys_1,fifth_half)
title('comp. of test. data(1st half) vs Val. data(5th half) using tf')
xlabel('Julian Date( Ts= 7 Days)')
figure ()
compare(sys_1,sixth_half)
title('comp. of test. data(1st half) vs Val. data(6th half) using tf')
xlabel('Julian Date( Ts= 7 Days)')
```

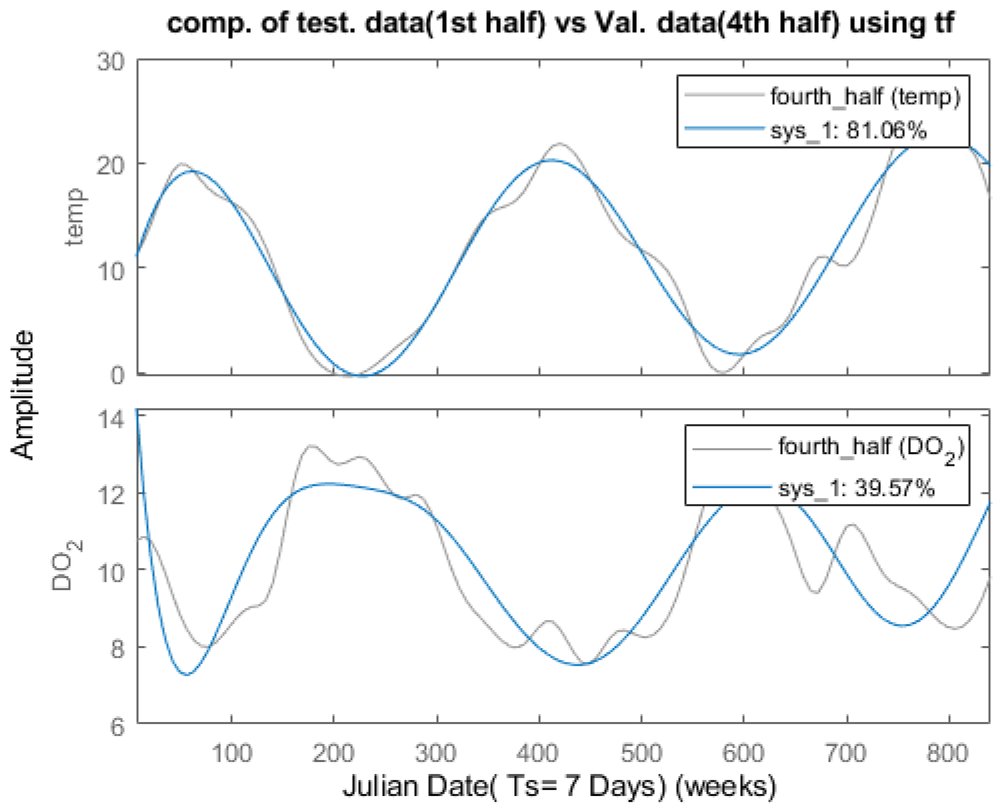
```
sys =

  From input "Ambtemp" to output...
                                   2.387e-10
   temp:  -----------------------------------------------------------
          s^4 + 0.001212 s^3 + 0.0003002 s^2 + 2.594e-07 s + 1.99e-09

                                   1.296e-10
   DO_2:  ------------------------------------------------------------
          s^4 + 0.0006132 s^3 + 0.0003003 s^2 + 1.43e-07 s + 1.426e-10

  From input "cms" to output...
                                   8.045e-10
```

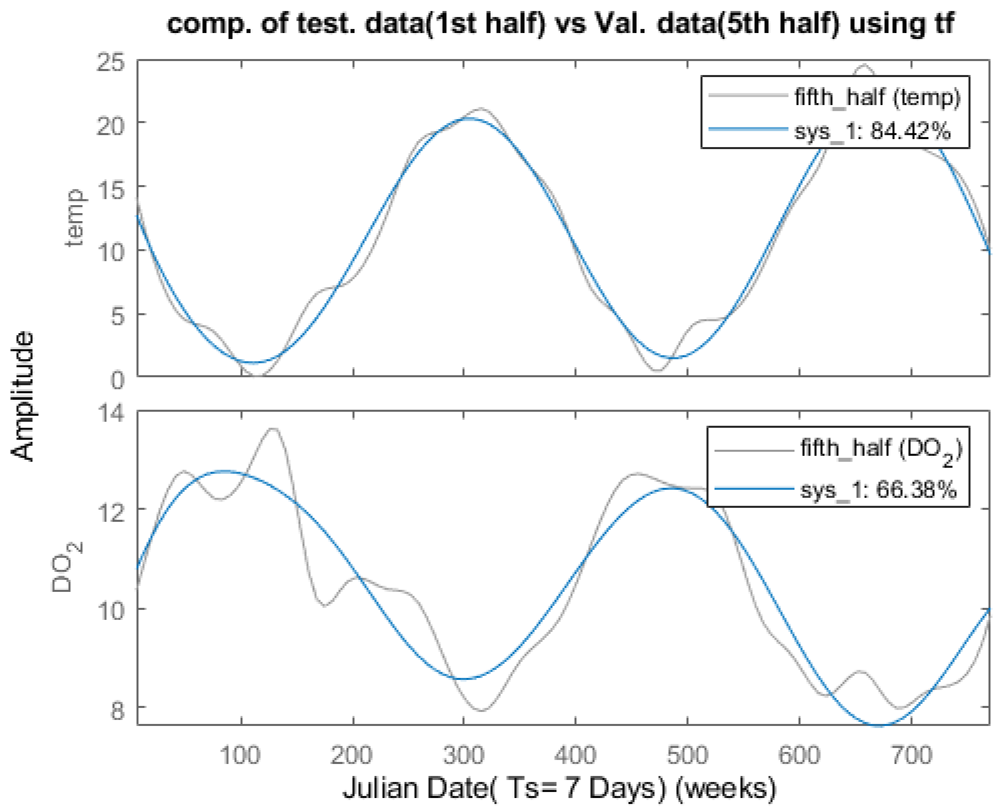
```
  temp:  ------------------------------------------------------------
         s^4 + 0.002049 s^3 + 0.0002984 s^2 + 4.975e-07 s + 7.767e-10

                                 -1.249e-11
  DO_2:  -----------------------------------------------------------
         s^4 + 0.0009116 s^3 + 5.081e-06 s^2 + 2.816e-09 s + 1.6e-12

Continuous-time identified transfer function.

Parameterization:
  Number of poles: [4 4;4 4]   Number of zeros: [0 0;0 0]
  Number of free coefficients: 20
  Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "weekly".
Fit to estimation data: [73.63;47.98]%
FPE: 2.656, MSE: 4.182
```

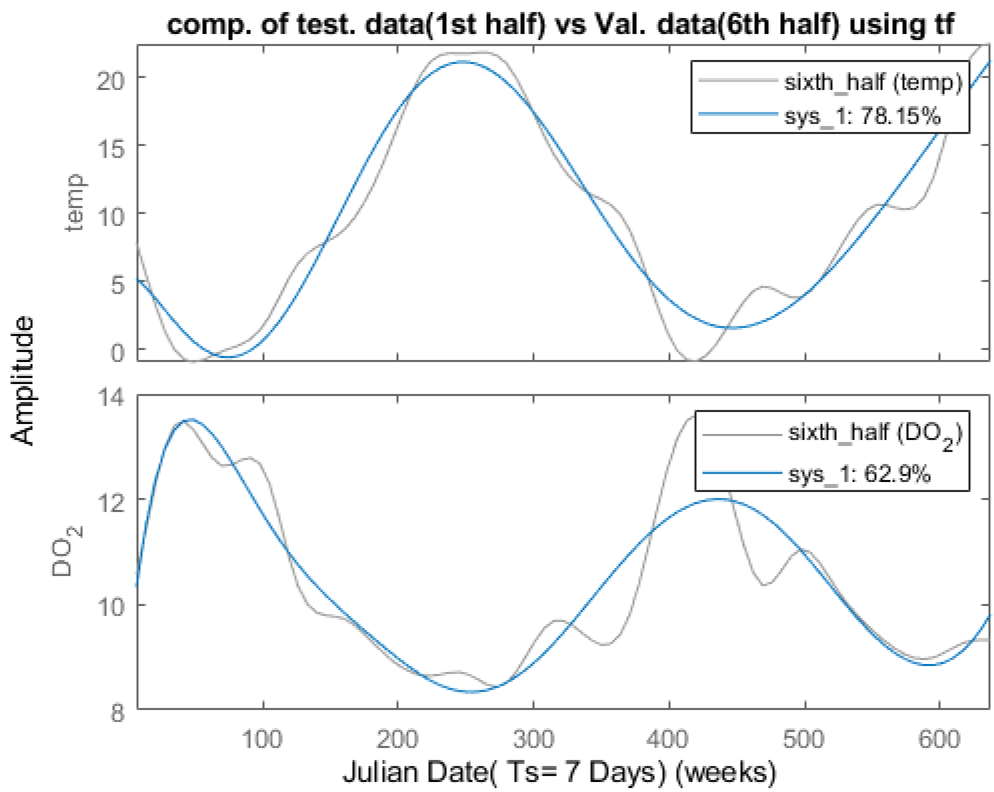



comp. of test. data(1st half) vs Val. data(2nd half) using tf
second_half (temp)
sys_1: 86.98%
second_half (DO2)
sys_1: 52.86%
Amplitude
temp
DO2
Julian Date(Ts= 7 Days) (weeks)


comp. of test. data(1st half) vs Val. data(3rd half) using tf
third_half (temp)
sys_1: 79.66%
third_half (DO2)
sys_1: 61.81%
Amplitude
temp
DO2
Julian Date(Ts= 7 Days) (weeks)


comp. of test. data(1st half) vs Val. data(4th half) using tf
fourth_half (temp)
sys_1: 81.06%
fourth_half (DO2)
sys_1: 39.57%
Amplitude
temp
DO2
Julian Date(Ts= 7 Days) (weeks)


comp. of test. data(1st half) vs Val. data(5th half) using tf
fifth_half (temp)
sys_1: 84.42%
fifth_half (DO2)
sys_1: 66.38%
Amplitude
temp
DO2
Julian Date(Ts= 7 Days) (weeks)



Published with MATLAB® R2020b

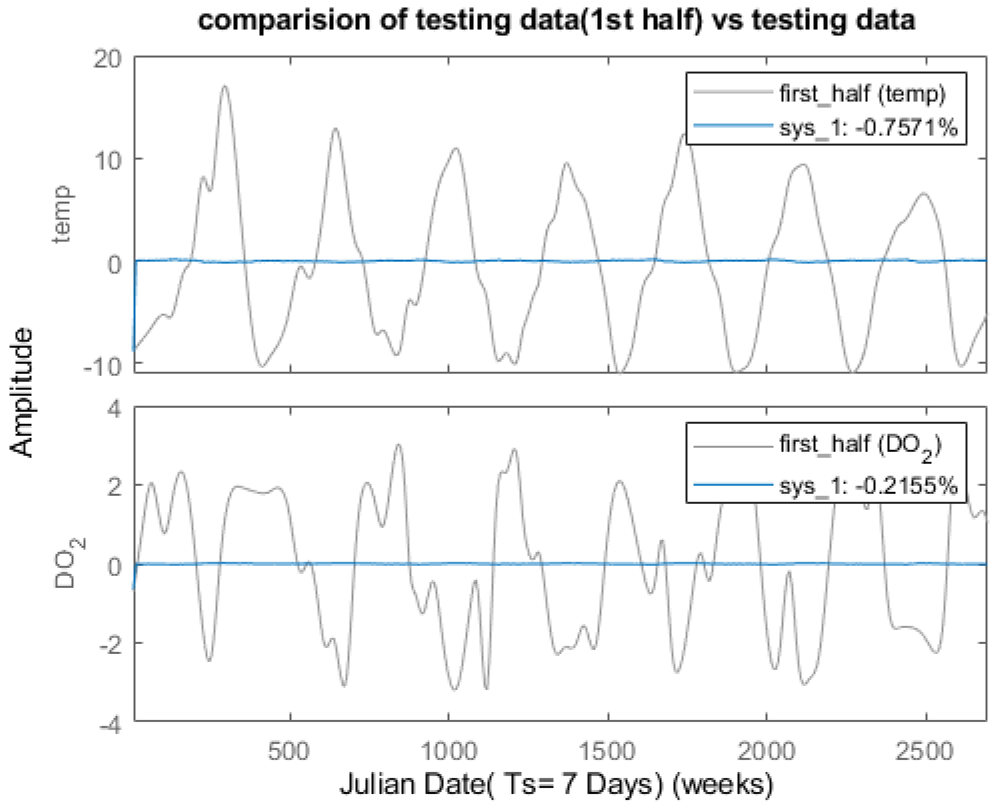
Appendix 5. System Identification Using CONSTID

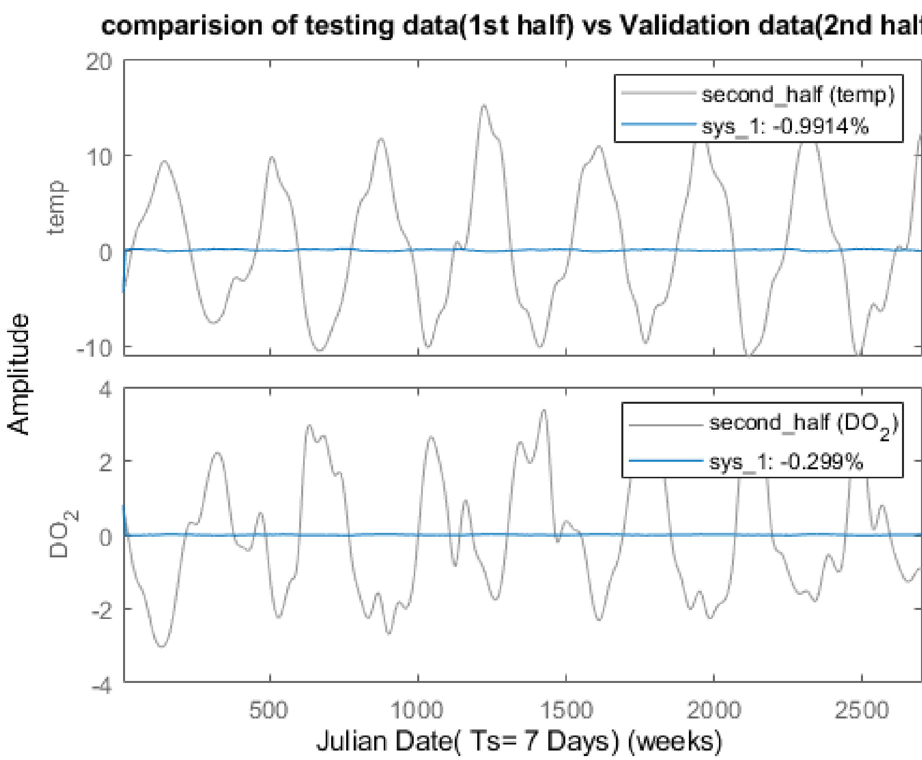
1. System Identification Using SIDGPMF for MIMO

```
%this script is for the system identification of fitcher
%data set.
close all;
clc;
clear;
load fitcher_weekly.mat;% this will load the testing data for daily measurements
%this data set is collected for the fitcher data set and Ambient
%temperature data set. It has the inputs Ambient temperature of Pocatello
%and the cms is the flow of the river.
%The outputs are temperature of the water in the river and the dissolved
%oxygen of that is present in the water of the river.
% use tfest or ssregest
y7=[Temp,DO_2];
u7=[Ambtemp_time,cms];
y7_avg=y7-mean(y7);
weekly=iddata(y7_avg,u7,Ts,'InterSample',{'foh';'foh'});
weekly.inputname(1)={'Ambtemp'};
weekly.inputname(2)={'cms'};
weekly.timeunit='weeks';
weekly.outputname(1)={'temp'};
weekly.outputname(2)={'DO_2'};
sys=tfest((weekly),4)
first_half= iddata(y7_avg(1:385,:),u7(1:385,:),Ts,'InterSample',{'foh';'foh'});
first_half.inputname(1)={'Ambtemp'};
first_half.inputname(2)={'cms'};
first_half.timeunit='weeks';
first_half.outputname(1)={'temp'};
first_half.outputname(2)={'DO_2'};
second_half=iddata(y7_avg(386:771,:),u7(386:771,:),Ts,'InterSample',{'foh';'foh'});
second_half.inputname(1)={'Ambtemp'};
second_half.inputname(2)={'cms'};
second_half.timeunit='weeks';
second_half.outputname(1)={'temp'};
second_half.outputname(2)={'DO_2'};
lambda=10;
i=2;
j=2;
n=2;
sys_1=sidgpmf(first_half,i,j,lambda,n);
sys_2=sidgpmf(second_half,i,j,lambda,n);
figure()
compare(sys_1,first_half)
title('comparision of testing data(1st half) vs testing data')
xlabel('Julian Date( Ts= 7 Days)')
figure ()
compare(sys_1,second_half)
```

```
title('comparision of testing data(1st half) vs Validation data(2nd half)')
xlabel('Julian Date( Ts= 7 Days)')
```

```
sys =

  From input "Ambtemp" to output...
            0.0003511 s^3 + 6.138e-06 s^2 + 1.149e-07 s + 2.564e-10
   temp:  ----------------------------------------------------------
          s^4 + 0.002698 s^3 + 0.0003524 s^2 + 6.53e-07 s + 1.308e-08

             0.000113 s^3 - 5.252e-07 s^2 + 2.734e-08 s - 2.385e-12
   DO_2:  -----------------------------------------------------------
          s^4 + 0.002242 s^3 + 0.0003053 s^2 + 6.514e-07 s + 4.287e-09

  From input "cms" to output...
            -0.003814 s^3 - 9.202e-06 s^2 - 2.909e-07 s - 2.521e-09
   temp:  -----------------------------------------------------------
          s^4 + 0.0003029 s^3 + 0.0003624 s^2 + 5.324e-08 s + 1.31e-08

             0.0008974 s^3 - 6.535e-06 s^2 + 1.581e-07 s - 1.61e-10
   DO_2:  -----------------------------------------------------------
          s^4 + 0.0009135 s^3 + 0.0003754 s^2 + 2.573e-07 s + 2.11e-08

Continuous-time identified transfer function.

Parameterization:
   Number of poles: [4 4;4 4]   Number of zeros: [3 3;3 3]
   Number of free coefficients: 32
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "weekly".
Fit to estimation data: [72.18;48.47]%
FPE: 3.341, MSE: 4.648
```





Published with MATLAB® R2020b

2. System Identification Using COE for SISO

```
%this script is for the system identification of fichter data set. close all;
clc; %clear;
load fitcher_weekly.mat; % this will load the testing data for weekly  measurements
%This data set is collected from the Fichter data set and Ambient
%temperature data set. It has the inputs Ambient temperature of Pocatello %and the cms is the flow of the river.
%The outputs are temperature of the water in the river and the dissolved %oxygen of that is present in the water of the river.

y7=[Temp]; % coe is MISO, so only estimate water temp as a first trial
u7=[Ambtemp_time]; % try single input case y7_avg=y7-mean(y7);
weekly=iddata(y7_avg,u7,Ts,'InterSample','foh'); weekly.inputname(1)={'Ambtemp'};
%weekly.inputname(2)={'cms'}; weekly.timeunit='days';
weekly.outputname(1)={'temp (about mean)'}; %weekly.outputname(2)={'DO_2'};
sys=tfest(weekly,4) % using matlab system ID function tfest compare(weekly,sys);
title('sys from tfest on its estimation dataset')

first_half= iddata(y7_avg(1:385,:),u7(1:385,:),Ts,'InterSample','foh');
first_half.inputname(1)={'Ambtemp'}; %first_half.inputname(2)={'cms'};
first_half.timeunit='days'; first_half.outputname(1)={'temp (about mean)'};
%first_half.outputname(2)={'DO_2'};
second_half=iddata(y7_avg(386:771,:),u7(386:771,:),Ts,'InterSample','foh');
second_half.inputname(1)={'Ambtemp'}; %second_half.inputname(2)={'cms'};
second_half.timeunit='days'; second_half.outputname(1)={'temp (about mean)'};
%second_half.outputname(2)={'DO_2'}; lambda0= 1/7; % cutoff frequency on the order of 1/7 cycle per week: % nn=[4 4 1]; sys_1=coe(first_half, nn, lambda0)

sys_2=coe(second_half, nn, lambda0)

figure() compare(first_half,sys_1,'r') title('comparision of testing data(1st half) vs testing data')

xlabel('Julian Date( Ts= 7 Days)') figure () compare(sys_1,second_half) title('comparision of testing data(1st half) vs Validation data(2nd half)') xlabel('Julian Date( Ts= 7 Days)') figure () compare(weekly,sys_1,'r',sys_2,'b',sys,'g')

title('comparision of all 3 estimates on entire data set') xlabel('Julian Date( Ts= 7 Days)')
```

```
sys =   From input "Ambtemp" to output "temp (about mean)":

  0.01137 s^3 + 3.899e-05 s^2 - 1.696e-08 s + 1.079e-11
```

--- s^4 + 0.0451 s^3 + 0.0003991 s^2 + 5.424e-06 s + 3.174e-20 Continuous-time identified transfer function.

Parameterization:

Number of poles: 4 Number of zeros: 3

Number of free coefficients: 8 Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:

Estimated using TFEST on time domain data "weekly".

Fit to estimation data: 69.65%

FPE: 4.731, MSE: 4.586 COE initiated from IVGPMF

sys_1 = Continuous-time OE model: y(t) = [B(s)/F(s)]u(t)

B(s) = 0.006813 s^3 + 3.171e-05 s^2 + 2.005e-05 s - 4.214e-09

F(s) = s^4 + 0.01603 s^3 + 0.003236 s^2 + 4.602e-05 s + 1.006e-06

Input delays (listed by channel): 7 Parameterization:

Polynomial orders: nb=4 nf=4 nk=0

Number of free coefficients: 8 Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:

Estimated using Contsid COE method on time domain data.

Fit to estimation data: 51.19

FPE: 1.208e+01, MSE 6.731e+01 COE initiated from IVGPMF

sys_2 = Continuous-time OE model: y(t) = [B(s)/F(s)]u(t)

B(s) = 0.006493 s^3 + 2.294e-05 s^2 + 5.037e-06 s + 3.439e-09

F(s) = s^4 + 0.02052 s^3 + 0.001795 s^2 + 1.226e-05 s + 4.498e-07

Input delays (listed by channel): 7 Parameterization:

Polynomial orders: nb=4 nf=4 nk=0

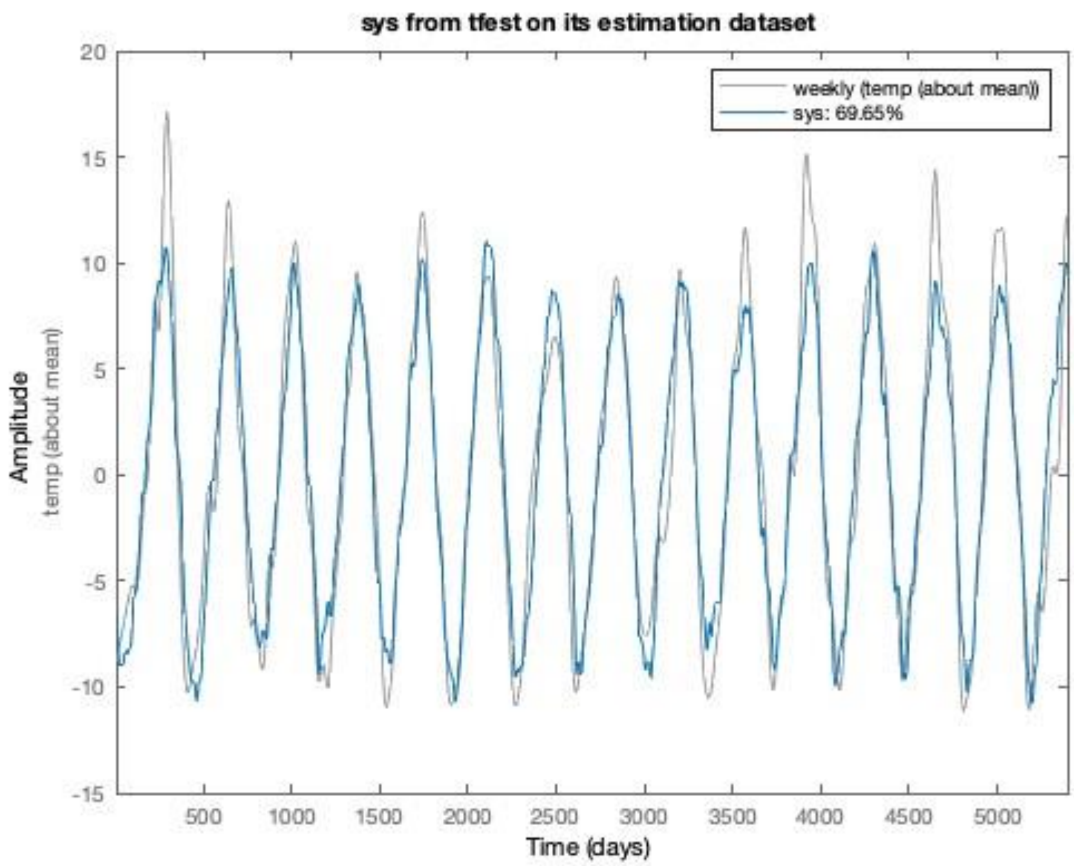
Number of free coefficients: 8 Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

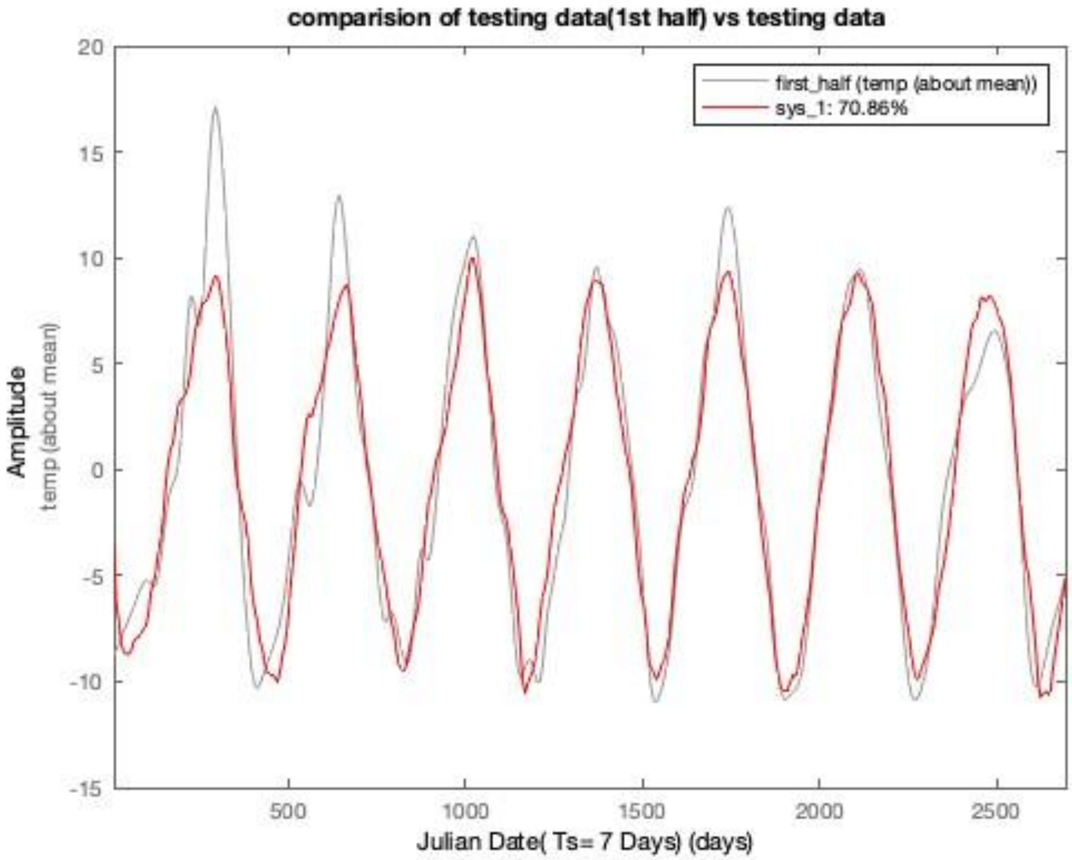
Status:

Estimated using Contsid COE method on time domain data.

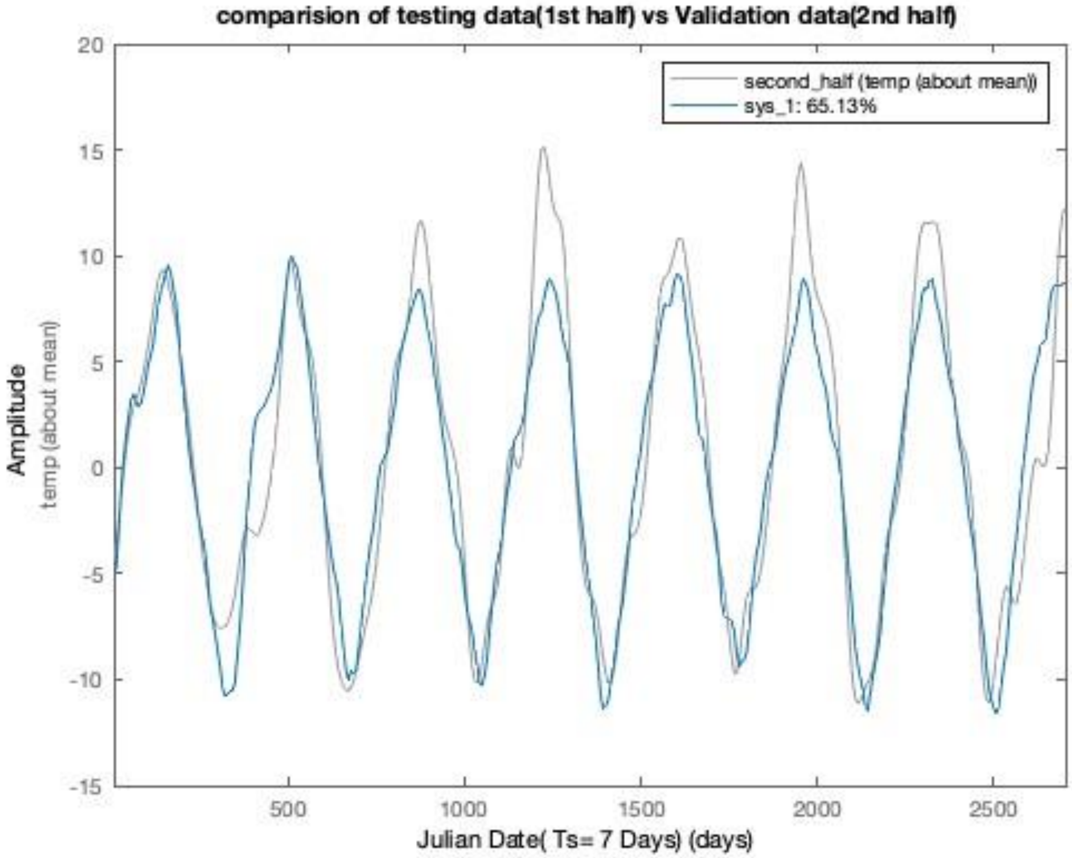
Fit to estimation data: 70.59

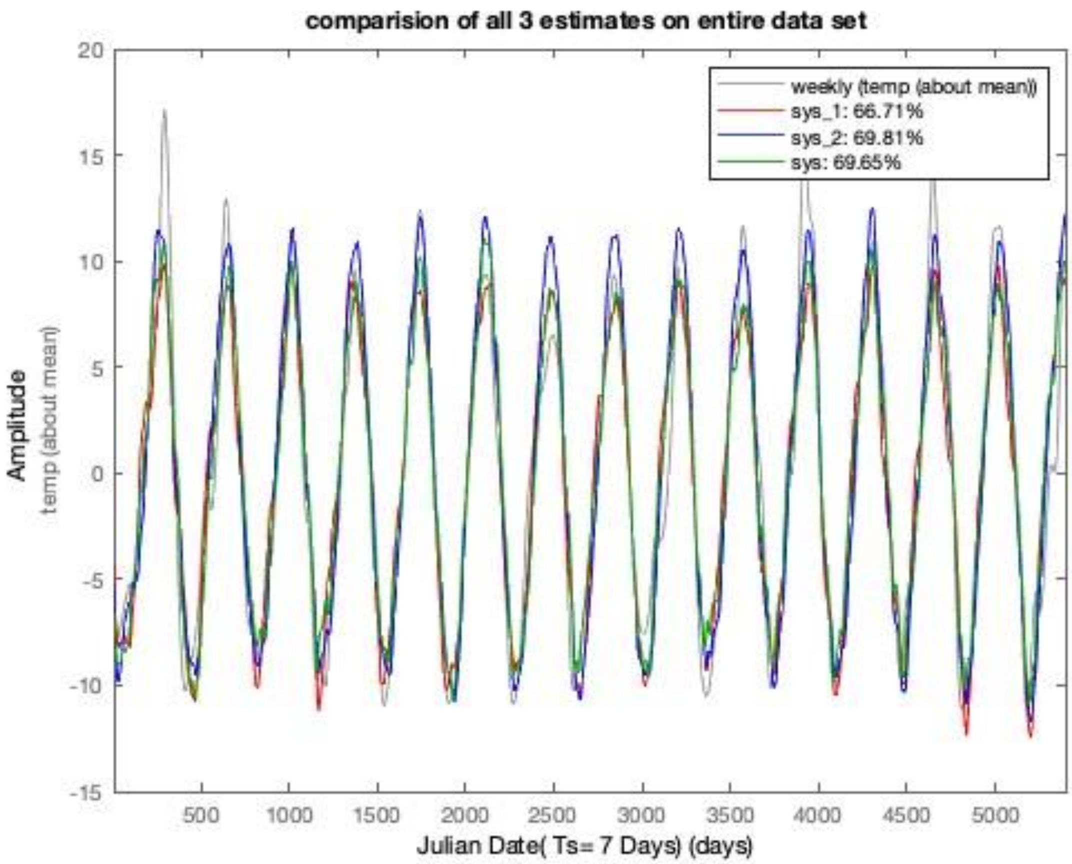
FPE: 4.468e+00, MSE 4.064e+01





comparision of testing data(1st half) vs testing data
first_half (temp (about mean))
sys_1: 70.86%
Amplitude
temp (about mean)
Julian Date(Ts= 7 Days) (days)
20
15
10
5
0
-5
-10
-15
500
1000
1500
2000
2500


comparision of testing data(1st half) vs Validation data(2nd half)
second_half (temp (about mean))
sys_1: 65.13%
Amplitude
temp (about mean)
Julian Date(Ts= 7 Days) (days)
20
15
10
5
0
-5
-10
-15
500
1000
1500
2000
2500



Published with MATLAB® R2022a

Thank You