ARISTOTLE UNIVERSITY OF THESSALONIKI
DEPARTMENT OF MATHEMATICS
POSTGRADUATE PROGRAM
"THEORETICAL COMPUTER SCIENCE AND THEORY OF CONTROL AND SYSTEMS"

Discretization of Singular Systems And Error Estimation

MASTER THESIS

RALLIS I.KARAMICHALIS

Supervisor: Nikolaos Karampetakis
Associate Professor
Aristotle University of Thessaloniki

Thessaloniki, June 2012

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
"ΘΕΩΡΗΤΙΚΗ ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΘΕΩΡΙΑ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΛΕΓΧΟΥ"

Διακριτοποίηση Ιδιόμορφων Συστημάτων Και Εκτίμηση Σφάλματος

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ράλλης Ι.Καραμιχάλης

Επιβλέπων: Νικόλαος Καραμπετάκης
Αν. Καθηγητής
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Θεσσαλονίκη, Ιούνιος 2012

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
"ΘΕΩΡΗΤΙΚΗ ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΘΕΩΡΙΑ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΛΕΓΧΟΥ "

Διακριτοποιήση Ιδιόμορφων Συστημάτων και Εκτίμηση Σφάλματος

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ράλλης Ι.Καραμιχάλης

Επιβλέπων: Νικόλαος Καραμπετάκης
Αν. Καθηγητής
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 28η Ιουνίου 2012.

.....	.....	.....
Ν. Καραμπετάκης	Α.–Ι. Βαρδουλάκης	Μ. Γουσίδου-Κουτίτα
Αν. Καθηγητής Α.Π.Θ.	Καθηγητής Α.Π.Θ.	Αν.Καθηγητής Α.Π.Θ.

Θεσσαλονίκη, Ιούνιος 2012

……..

Ράλλης Ι.Καραμιχάλης
Πτυχιούχος Μαθηματικός Α.Π.Θ.

Στους γονείς μου,
Γιάννη, Μελαχροινή

ΠΕΡΙΛΗΨΗ

Στον ψηφιακό έλεγχο όπως και σε πολλούς άλλους κλάδους της Μηχανικής, προβάλλει ως επιτακτική η ανάγκη να μετατρέψουμε συστήματα συνεχούς χρόνου σε διακριτά state space συστήματα. Η διαδικασία της διακριτοποίησης ωστόσο εισάγει ένα σφάλμα μεταξύ της συνεχούς και της διακριτοποιημένης λύσης. Πιο συγκεκριμένα, μελετάμε Γραμμικά Χρονικά Αναλλοίωτα διαφορικά συστήματα της μορφής $E\dot{x}(t) = Ax(t) + Bu(t)$ όπου $E, A \in M(n \times n, F)$ (όπου Μ είναι το σύνολο όλων των τετραγωνικών πινάκων με στοιχεία από το σώμα $F = R$ ή C), $\det E = 0$ και $B \in M(n \times l, F)$ είναι σταθεροί πίνακες. Υποθέτουμε επίσης οτι το διάνυσμα κατάστασης $x(t) \in M(n \times 1, F)$ όπου κάθε $x_i(t) : F \to F$ και $u(t) \in M(l \times 1, F)$ όπου κάθε $u_i(t) : F \to F$. Από τη βιβλιογραφία των μεθόδων διακριτοποίησης, μπορούμε να διακρίνουμε δύο κύριες κατηγορίες, Η πρώτη από αυτές (βλέπε [3] και [4]) βασίζεται στο ανάπτυγμα Laurent του πίνακα $(sE - A)^{-1}$ ενώ η δεύτερη χρησιμοποιεί την Weierstrass κανονική μορφή. Και οι δύο μέθοδοι είναι κατά κάποιο τρόπο ισοδύναμοι καθώς χρησιμοποιούν συγκράτηση μηδενικής ταξης (zero order hold) για την είσοδο u(t). Η εργασία αυτή επομένως αποτελεί μια επέκταση στη δεύτερη αυτή κατηγορία καθώς χρησιμοποιεί συγκράτηση πρώτης τάξης (first order hold). Συνεπώς, σε αυτή την εργασία έχουμε δύο ενδιαφέροντα αποτελέσματα. Αρχικά, ένα πιο κλειστό και γενικότερης μορφής άνω φράγμα για τη νόρμα της διαφοράς μεταξύ της συνεχούς και της διακριτής λύσης $||x(kT) - x_k||$, δίνεται στο 4^{o} κεφάλαιο. Το αποτέλεσμα αυτό είναι καινούργιο και επεκτείνει τα ήδη γνωστά αποτελέσματα απο το [3], όπου και χρησιμοποιείται συγκράτηση μηδενικής τάξης. Επιπλέον, το νέο αυτό άνω φράγμα που παρουσιάζεται, μας επιτρέπει να επιλέξουμε κατάλληλα την περίοδο δειγματοληψίας T σε ένα διάστημα της μορφής [0,α]. Με αυτό τον τρόπο υποθέτοντας οτι επιθυμούμε η διαφορά μεταξύ της συνεχούς και της διακριτής λύσης τη χρονική στιγμή t=kT να μην υπερβαίνει μία δοθείσα τιμή, μπορούμε να υπολογίσουμε την περίοδο δειγματοληψίας T η οποία ικανοποιεί τις απαιτήσεις μας. Πιο συγκεκριμένα, στο δεύτερο κεφάλαιο υπολογίζουμε μια αναδρομική φόρμουλα για τα διακριτά σημεία x_k και αποδεικνύουμε επαγωγικά έναν αναλυτικό τύπο γι' αυτά. Στο τρίτο κεφάλαιο εν συνεχεία, παίρνουμε την πρώτη εκδοχή του άνω φράγματος της νόρμας της διαφοράς. Στο τέταρτο κεφάλαιο

απλοποιούμε τη μορφή του άνω φράγματος και το καθιστούμε εφικτό υπολογιστικά, καθώς εξ΄ αρχής συμπεριλάμβανε πληθώρα παραμέτρων, και τέλος στο πέμπτο και τελευταίο κεφάλαιο παρουσιάζουμε την όλη θεωρία και υπολογίζουμε την περίοδο δειγματοληψίας και το άνω φράγμα για ένα συγκεκριμένο παράδειγμα. Στο τέλος της εργασίας επίσης, συμπεριλαμβάνεται κώδικας στο Mathematica για τον υπολογισμό των τιμών των παραμέτρων ενός συστήματος καθώς και για το σχεδιασμό κατάλληλων γραφικών παραστάσεων μεταξύ της συνεχούς και της διακριτής λύσης.

ABSTRACT

This paper proposes a discretization technique of a differential system of the form $E\dot{x}(t) = Ax(t) + Bu(t)$ based on the matrix pencil theory. The methodology used is a first order hold discretization for the input function u(t). This is an extension to what is already known, where only zero order hold discretization has been used. Moreover a new, sharper, and more general form of upper bound for the discretization error is presented. These two error bounds are compared in the last chapter throughout an example which illustrates the whole theory and penalizes our choice for the sampling period.

KEY WORDS

Discretization technique, first order hold approximation, estimate sampling period, discretized solution, upper bound

CONTENTS

Chapter 1

Introduction

In digital control, and in several areas of engineering, we need to convert continuous-time into discrete-time state space equations. The discretization process though, introduces an error between the continuous and the discretized solution. More specifically, we study Linear Time Invariant (LTI) differential systems of the form

$$E\dot{x}(t) = Ax(t) + Bu(t) \tag{1.1}$$

where $E, A \in M(n \times n, F)$ where M denotes the set of all square matrices with elements in the field $F = R$ or C, $\det E = 0$ and $B \in M(n \times l, F)$ are constant matrices. We also assume that state vector $x(t) \in M(n \times 1, F)$, where each $x_i(t) : F \mapsto F$, has consistent initial conditions and that input vector $u(t) \in M(l \times 1, F)$ where also each $u_i(t) : F \mapsto F$.

From the literature of the discretization methods for descriptor differential systems, we may group them in two different interesting methods. The first one, (see [4] and [5]) based on the Laurent expansion of $(sE - A)^{-1}$ and the second one based on the matrix pencil theory, using Weierstrass canonical form. Both of them are somehow equivalent using zero order hold approximation. This thesis is an extension to the second method, using first order hold approximation. Consequently, in this thesis, we provide two interesting results.

First, a sharper and more general upper bound for the norm of the difference between the continuous solution and the discretized solution $\|x(kT) - x_k\|$ has been given in chapter four. This results extends the already known upper bound from [3] which uses zero order hold approximation.

In addition to this, the new upper bound proposed here penalizes our choice for the sampling period T in an interval of the form $[0, a]$. This interval is comparable to the results given in [3]. In this way, assuming we want the difference between the continuous solution at the moments $t = kT$ and the relevant discrete-time points x_k to not exceed a given value, we can estimate a period T satisfying our demands.

More diagrammatically, in chapter two we find a recursive formula for points x_k and we conclude with an analytic formula. In chapter three, we get a first formula for the upper bound of the norm $\|x(kT) - x_k\|$. In chapter four, after some simplifications we end in a more simplified form of the upper bound and in the last chapter we illustrate the whole theory via an example.

Chapter 2

Preliminary Results and Problem Formulation

2.1 Preliminaries - The Weierstrass Canonical Form (WCF)

Linear generalized differential systems of type $E\dot{x}(t) = Ax(t)$, $E, A \in F^{n\times n}$ with $detE = 0$, where $x \in F^{n\times 1}$ and x_0 an initial value, are required in modelling of many physical, electrical and mechanical problems. Systems of this type are related to matrix pencil theory since the algebraic geometric and dynamic properties stem from the structure of the associated pencil $sE - A$.

Definition

Given $E, A \in F^{m\times n}$ and an indeterminate s, the matrix pencil $sE - A$ is called regular when $m = n$ and $det(sE - A) \neq 0$. In any other case, the pencil will be called singular.

The pencil $sE - A$ is said to be striclty equivalent to the pencil $s\tilde{E} - \tilde{A}$ if and only if there exist $P, Q \in C^{n\times n}$ such as $P(sE-A)Q = s\tilde{E}-\tilde{A}$ where $detP, detQ \neq 0$. The class of strict equivalent matrix pencils is characterized by a uniquely defined element, known as a complex Weierstrass canonical form, $sE_w - A_w$.

In the case where $sE - A$ is a regular pencil and $detE = 0$ we have elementary divisors of the following type:

- e.d of the type $(s - a)^p$ are called finite elementary divisors (f.e.d)
- e.d of the type $\hat{s}^q$ are called infinite elementary divisors (i.e.d)

Then, the Weierestrass canonical form of the regular pencil $sE - A$ is defined by

$$sE_w - A_w := \text{block diag}(sI_p - J_p, sH_q - I_q)$$

where the first normal Jordan block $sI_p - J_p$ is uniquely defined by the set of f.e.d

$$(s - a_1)^{p_1}, \cdots, (s - a_n)^{p_n}, \qquad \sum_{j=1}^{n} p_j = p$$

of $sE - A$ and has the form

$$sI_p - J_p := \text{block diag}(sI_{p_1} - J_{p_1}(a_1), \cdots, sI_{p_n} - J_{p_n}(a_n))$$

and the q blocks of the second uniquely defined block $sH_q - I_q$ correspond to the i.e.d

$$(\hat{s})^{q_1}, \cdots, (\hat{s})^{q_\sigma}, \qquad \sum_{j=1}^{\sigma} q_j = q$$

of $sE - A$ and has the form

$$sH_q - I_q := \text{block diag}(sH_{q_1} - I_{q_1}, \cdots, sH_{q_\sigma} - I_{q_\sigma})$$

Therefore, H_q is a nilpotent matrix of index $q^* = max[q_j : j = 1, 2, \cdots, \sigma]$, that is $H_q^{q^*} = 0$ and $I_p, J_p(a), H_q$ are the matrices

$$I_{p_i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1 \end{bmatrix} \in R^{p_i \times p_i}, \quad J_{p_i}(a_i) = \begin{bmatrix} a & 1 & & 0 \\ & a & \ddots & \\ & & \ddots & 1 \\ 0 & & & a \end{bmatrix} \in C^{p_i \times p_i}, \quad H_{q_i} = \begin{bmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & 0 \end{bmatrix} \in R^{q_i \times q_i}$$

Now, we consider the transformation $x(t) = Qy(t)$ and we get the following results.

2.2 Discretized recursive formula

As it has been already mentioned about the mathematical tools used during the discretization process, only Weierstrass canonical form (WCF) is required for this discretization technique. As this thesis extends [3] using first order hold approximation instead of zero order hold in order to get better results, some commonly used lemmas are presented without their proofs, although full references are provided. We already know that system (1.1), has the following continuous time solution, see [3] and [6]:

$$x(t) = Q_{n,p}\left(e^{J_p(t-t_0)}y_p(t_0) + \int_{t_0}^{t} e^{J_p(t-s)}B_{p,l}u(s)ds\right) - Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}u^{(i)}(t) \tag{2.1}$$

where

$$Q = \begin{bmatrix} Q_{n,p} & Q_{n,q} \end{bmatrix}, \qquad B = \begin{bmatrix} B_{p,l} \\ B_{q,l} \end{bmatrix} \quad , \quad y(t_0) = \begin{bmatrix} y_p(t_0) \\ y_q(t_0) \end{bmatrix} = Q^{-1}x(t_0)$$

and $u^{(i)}(t)$ the i^{th}-derevative of the input function u(t).

However, (2.1) can be transformed in a more useful format. We have:

$$\begin{aligned} x(t) =& Q_{n,p}e^{J_p(t-t_0)}y_p(t_0) + Q_{n,q}y_q(t_0) + Q_{n,p}\int_{t_0}^{t} e^{J_p(t-s)}B_{p,l}u(s)ds \\ & - Q_{n,q}y_q(t_0) - Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}u^{(i)}(t) \\ =& \begin{bmatrix} Q_{n,p} & Q_{n,q} \end{bmatrix}\begin{bmatrix} e^{J_p(t-t_0)} & O_{p,q} \\ O_{q,p} & I_q \end{bmatrix}\begin{bmatrix} y_p(t_0) \\ y_q(t_0) \end{bmatrix} + Q_{n,p}\int_{t_0}^{t} e^{J_p(t-s)}B_{p,l}u(s)ds \\ & + Q_{n,q}\left(-y_q(t_0) - \sum_{i=0}^{q^*-1} H_q^i B_{q,l}u^{(i)}(t)\right) \end{aligned}$$

In order now for the system (1.1) to obtain consistent initial conditions we should consider that

$$\begin{bmatrix} y_p(t_0) \\ y_q(t_0) \end{bmatrix} = Q^{-1}x(t_0), \quad \text{and} \quad -y_q(t_0) = \sum_{i=0}^{q^*-1} H_q^i B_{q,l} u^{(i)}(t_0))$$

and as a result we obtain

$$\begin{aligned} x(t) =& Q \begin{bmatrix} e^{J_p(t-t_0)} & O_{p,q} \\ O_{q,p} & I_q \end{bmatrix} Q^{-1}x(t_0) + Q_{n,p}\int_{t_0}^{t} e^{J_p(t-s)}B_{p,l}u(s)ds \\ &+ Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}\left(u^{(i)}(t_0) - u^{(i)}(t))\right) \end{aligned}$$

Moreover, by definition, the state-transition matrix of the autonomous linear descriptor differential system, $E\dot{x}(t) = Ax(t)$, is given by

$$\Phi(t,t_0) = Q \begin{bmatrix} e^{J_p(t-t_0)} & O_{p,q} \\ O_{q,p} & I_q \end{bmatrix} Q^{-1},$$

Finally, after noticing that

$$\Phi(t,s)Q_{n,p} = Q_{n,p}e^{J_p(t-s)}$$

since

$$\Phi(t,s)Q_{n,p} = \Phi(t,s)\begin{bmatrix} Q_{n,p} & Q_{n,q} \end{bmatrix}\begin{bmatrix} I_{p,p} \\ O_{q,p} \end{bmatrix} = \begin{bmatrix} Q_{n,p}e^{J_p(t-s)} & O_{n,q} \end{bmatrix}\begin{bmatrix} I_{p,p} \\ O_{q,p} \end{bmatrix} = Q_{n,p}e^{J_p(t-s)}$$

we get

$$x(t) = \Phi(t,t_0)x(t_0) + \int_{t_0}^{t}\Phi(t,s)Q_{n,p}B_{p,l}u(s)ds + Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}\left(u^{(i)}(t_0) - u^{(i)}(t)\right) \tag{2.2}$$

Now, let $T > 0$ be the constant sampling period. We also assume that $t_0 = 0$. We further assume that the input function $u(\tau)$ is not constant in the interval $[kT, kT+T)$ and we approximate it by using the Triangular First Order Hold (TFOH) approximation. As a result, input function $u(\tau)$ is given by the formula

$$u(\tau) = u(kT) + \frac{u((k+1)T) - u(kT)}{T}(\tau - kT)$$

$\forall \tau \in [kT, kT+T)$. For use of simplicity, hereafter, we use the notation $x_k := x(kT) \ \ \forall k = 0, 1, 2, \ldots$ From equation (2.2) by setting $t = kT$ and $t = (k+1)T$, we get

$$x_k = \Phi(kT,0)x_0 + Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_0^{(i)} - u_k^{(i)}) + \int_0^{kT}\Phi(kT,s)Q_{n,p}B_{p,l}u(s)ds \tag{2.3}$$

$$x_{k+1} = \Phi((k+1)T,0)x_0 + Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_0^{(i)} - u_{k+1}^{(i)}) + \int_0^{(k+1)T}\Phi((k+1)T,s)Q_{n,p}B_{p,l}u(s)ds \tag{2.4}$$

Lemma 2.2.1
The following equalitites hold:

$$\Phi(T,0)\Phi(kT,s) = \Phi((k+1)T,s)$$
$$\Phi(T,0)Q_{n,q} = Q_{n,q}$$

Proof:
We know that

$$\Phi(t,t_0) = Q\begin{bmatrix} e^{J_p(t-t_0)} & O_{p,q} \\ O_{q,p} & I_q \end{bmatrix} Q^{-1}$$

so

$$\Phi(T,0)\Phi(kT,s) = Q\begin{bmatrix} e^{J_pT} & O_{p,q} \\ O_{q,p} & I_q \end{bmatrix} Q^{-1} Q \begin{bmatrix} e^{J_p(kT-s)} & O_{p,q} \\ O_{q,p} & I_q \end{bmatrix} Q^{-1}$$

$$= Q\begin{bmatrix} e^{J_p((k+1)T-s)} & O_{p,q} \\ O_{q,p} & I_q \end{bmatrix} Q^{-1} = \Phi((k+1)T,s).$$

We also have that $Q = \begin{bmatrix} Q_{n,p} & Q_{n,q} \end{bmatrix}$. Now, we define $Q^{-1} = \begin{bmatrix} \tilde{Q}_{p,n} \\ \tilde{Q}_{q,n} \end{bmatrix}$ and

$$Q^{-1}Q = \begin{bmatrix} \tilde{Q}_{p,n}Q_{n,p} & \tilde{Q}_{p,n}Q_{n,q} \\ \tilde{Q}_{q,n}Q_{n,p} & \tilde{Q}_{q,n}Q_{n,q} \end{bmatrix} = I_n$$

with

$$\tilde{Q}_{p,n}Q_{n,p} = I_p, \quad \tilde{Q}_{p,n}Q_{n,q} = O_{p,q}, \quad \tilde{Q}_{q,n}Q_{n,p} = O_{q,p}, \quad \tilde{Q}_{q,n}Q_{n,q} = I_q.$$

So we have,

$$\Phi(T,0)\,Q_{n,q} = Q\begin{bmatrix} e^{J_pT} & O_{p,q} \\ O_{q,p} & I_q \end{bmatrix} Q^{-1}Q_{n,q} = \begin{bmatrix} Q_{n,p} & Q_{n,q} \end{bmatrix} \begin{bmatrix} e^{J_pT} & O_{p,q} \\ O_{q,p} & I_q \end{bmatrix} \begin{bmatrix} \tilde{Q}_{p,n}Q_{n,q} \\ \tilde{Q}_{q,n}Q_{n,q} \end{bmatrix}$$
$$= \begin{bmatrix} Q_{n,p} & Q_{n,q} \end{bmatrix} \begin{bmatrix} e^{J_pT} & O_{p,q} \\ O_{q,p} & I_q \end{bmatrix} \begin{bmatrix} O_{p,q} \\ I_q \end{bmatrix} = \begin{bmatrix} Q_{n,p} & Q_{n,q} \end{bmatrix} \begin{bmatrix} O_{p,q} \\ I_q \end{bmatrix} = Q_{n,q} \quad \blacktriangle$$

From (2.3), (2.4),the above lemma, and multiplying (2.3) by $\Phi(T,0)$ and substracting from (2.4), we finally get

$$x_{k+1} - \Phi(T,0)x_k = \int_{kT}^{(k+1)T} \Phi((k+1)T,s)Q_{n,p}B_{p,l}u(s)ds + Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}((u_0^{(i)} - u_{k+1}^{(i)})$$
$$- \underbrace{\Phi(T,0)Q_{n,q}}_{\mathbf{Q_{n,q}}} \sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_0^{(i)} - u_k^{(i)})$$

and therefore the following recursive formula derives

$$x_{k+1} = \Phi(T,0)x_k + \int_{kT}^{(k+1)T} \Phi((k+1)T,s)Q_{n,p}B_{p,l}u(s)ds + Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_k^{(i)} - u_{k+1}^{(i)}) \qquad (2.5)$$

But,

$$\begin{aligned}
&\int_{kT}^{(k+1)T} \Phi((k+1)T,s)Q_{n,p}B_{p,l}u(s)ds \overset{s=kT+w}{=} \int_0^T \Phi((k+1)T,kT+w)Q_{n,p}B_{p,l}u(kT+w)dw \\
&= \int_0^T \Phi(T-w,0)Q_{n,p}B_{p,l}\Big(u_k + \frac{u_{k+1}-u_k}{T}w\Big)dw \\
&= \int_0^T \Phi(T-w,0)Q_{n,p}B_{p,l}u_k dw + \int_0^T \Phi(T-w,0)Q_{n,p}B_{p,l}\frac{u_{k+1}-u_k}{T}w dw \\
&\overset{\lambda=T-w}{=} \int_0^T \Phi(\lambda,0)d\lambda Q_{n,p}B_{p,l}u_k + \int_0^T \Phi(\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_{k+1}-u_k}{T}
\end{aligned}$$

so (2.5) is transformed into,

$$\begin{aligned}
x_{k+1} =&\Phi(T,0)x_k + \int_0^T \Phi(\lambda,0)d\lambda Q_{n,p}B_{p,l}u_k + \int_0^T \Phi(\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_{k+1}-u_k}{T} \\
&+ Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_k^{(i)} - u_{k+1}^{(i)})
\end{aligned} \tag{2.6}$$

2.3 Discretized analytic formula

Theorem 2.3.1
From the recursive formula (2.6) we can get using induction the following analytic formula

$$\begin{aligned}
x_k =&\Phi(kT,0)x_0 + \sum_{j=0}^{k-1}\int_0^T \Phi(jT+\lambda,0)d\lambda Q_{n,p}B_{p,l}u_{k-j-1} \\
&+ \sum_{j=0}^{k-1}\int_0^T \Phi(jT+\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_{k-j}-u_{k-j-1}}{T} + Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_0^{(i)} - u_k^{(i)})
\end{aligned} \tag{2.7}$$

Proof:
First of all, for $k=0$ in (2.6) we have

$$\begin{aligned}
x_1 =&\Phi(T,0)x_0 + \int_0^T \Phi(\lambda,0)d\lambda Q_{n,p}B_{p,l}u_0 + \int_0^T \Phi(\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_1-u_0}{T} \\
&+ Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_0^{(i)} - u_1^{(i)})
\end{aligned}$$

which is obviously true (case $k=1$ in (2.7)). We assume that this is true for $k-1$, that is

$$\begin{aligned}
x_{k-1} =& \Phi((k-1)T,0)x_0 + \sum_{j=0}^{k-2}\int_0^T \Phi(jT+\lambda,0)d\lambda Q_{n,p}B_{p,l}u_{k-j-2} \\
&+ \sum_{j=0}^{k-2}\int_0^T \Phi(jT+\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_{k-j-1}-u_{k-j-2}}{T} + Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_0^{(i)} - u_{k-1}^{(i)})
\end{aligned}$$

and we prove it for k. From the recursive formula (2.6), we have

$$\begin{aligned}x_k =&\Phi(T,0)x_{k-1} + \int_0^T \Phi(\lambda,0)d\lambda Q_{n,p}B_{p,l}u_{k-1} + \int_0^T \Phi(\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_k - u_{k-1}}{T}\\ &+ Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_{k-1}^{(i)} - u_k^{(i)})\end{aligned}$$

and after substituting x_{k-1} in the above relation we get,

$$\begin{aligned}x_k =&\Phi(T,0)\Bigg(\Phi((k-1)T,0)x_0 + \sum_{j=0}^{k-2}\int_0^T \Phi(jT+\lambda,0)d\lambda Q_{n,p}B_{p,l}u_{k-j-2}\\ &+ \sum_{j=0}^{k-2}\int_0^T \Phi(jT+\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_{k-j-1}-u_{k-j-2}}{T} + Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_0^{(i)} - u_{k-1}^{(i)})\Bigg)\\ &+ \int_0^T \Phi(\lambda,0)d\lambda Q_{n,p}B_{p,l}u_{k-1} + \int_0^T \Phi(\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_k - u_{k-1}}{T}\\ &+ Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_{k-1}^{(i)} - u_k^{(i)})\\ =&\Phi(kT,0)x_0 + \sum_{j=0}^{k-2}\int_0^T \Phi((j+1)T+\lambda,0)d\lambda Q_{n,p}B_{p,l}u_{k-j-2}\\ &+ \sum_{j=0}^{k-2}\int_0^T \Phi((j+1)T+\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_{k-j-1}-u_{k-j-2}}{T}\\ &+ \underbrace{\Phi(T,0)Q_{n,q}}_{\mathbf{Q_{n,q}}}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_0^{(i)} - u_{k-1}^{(i)}) + \int_0^T \Phi(\lambda,0)d\lambda Q_{n,p}B_{p,l}u_{k-1}\\ &+ \int_0^T \Phi(\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_k - u_{k-1}}{T} + Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_{k-1}^{(i)} - u_k^{(i)})\end{aligned}$$

and now setting $i = j+1$ we get,

$$\begin{aligned}x_k =&\Phi(kT,0)x_0 + \sum_{i=1}^{k-1}\int_0^T \Phi(iT+\lambda,0)d\lambda Q_{n,p}B_{p,l}u_{k-i-1}\\ &+ \sum_{i=1}^{k-1}\int_0^T \Phi(iT+\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_{k-i}-u_{k-i-1}}{T}\\ &+ \int_0^T \Phi(\lambda,0)d\lambda Q_{n,p}B_{p,l}u_{k-1} + \int_0^T \Phi(\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_k - u_{k-1}}{T}\\ &+ Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_0^{(i)} - u_{k-1}^{(i)} + u_{k-1}^{(i)} - u_k^{(i)})\end{aligned}$$

and after grouping similar terms we get,

$$\begin{aligned} x_k =&\Phi(kT,0)x_0 + \sum_{i=0}^{k-1}\int_0^T \Phi(iT+\lambda,0)d\lambda Q_{n,p}B_{p,l}u_{k-i-1} \\ &+ \sum_{i=0}^{k-1}\int_0^T \Phi(iT+\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_{k-i}-u_{k-i-1}}{T} + Q_{n,q}\sum_{i=0}^{q^*-1} H_q^i B_{q,l}(u_0^{(i)}-u_k^{(i)}) \end{aligned}$$

which was what we wanted to prove. ▲

At this point, we can discretize also the i-th order derivatives $u_k^{(i)}$ using Euler approximations, that is

$$u^{(i)}(kT) = u_k^{(i)} = \frac{u_{k+1}^{(i-1)} - u_k^{(i-1)}}{T}$$

and from the above equation we can easily get that

$$u_k^{(i)} = \frac{1}{T^i}\sum_{j=0}^{i}(-1)^j \binom{i}{j} u_{k+j} \tag{2.8}$$

Using now (2.8) in (2.6) we get the completely discretized model:

$$\begin{aligned} x_k =&\Phi(kT,0)x_0 + \sum_{j=0}^{k-1}\int_0^T \Phi(jT+\lambda,0)d\lambda Q_{n,p}B_{p,l}u_{k-j-1} \\ &+ \sum_{j=0}^{k-1}\int_0^T \Phi(jT+\lambda,0)(T-\lambda)d\lambda Q_{n,p}B_{p,l}\frac{u_{k-j}-u_{k-j-1}}{T} \\ &+ Q_{n,q}\sum_{i=0}^{q^*-1}\frac{1}{T^i}H_q^i B_{q,l}\sum_{j=0}^{i}(-1)^j\binom{i}{j}(u_j - u_{k+j}) \end{aligned} \tag{2.9}$$

Chapter 3

Error Analysis and Upper Bound

In this section,having already found an analytic formula for the discretized solution x_k, we provide an analytic expression for the norm of the difference between the continuous time solution at the moments $t = kT$ and the discrete points x_k of the disretized solution. Moreover we bound this norm and we get a more general form of upper bound. From (2.2) and (2.7) we have,

$$
\begin{aligned}
x(kT) - x_k &= \int_0^{kT} \Phi(kT, s) Q_{n,p} B_{p,l} u(s) ds - \sum_{j=0}^{k-1} \int_0^T \Phi(jT + \lambda, 0) d\lambda Q_{n,p} B_{p,l} u_{k-j-1} \\
&- \sum_{j=0}^{k-1} \int_0^T \Phi(jT + \lambda)(T - \lambda) d\lambda Q_{n,p} B_{p,l} \frac{u_{k-j} - u_{k-j-1}}{T} \\
&= \sum_{j=0}^{k-1} \int_{jT}^{(j+1)T} \Phi(kT, s) Q_{n,p} B_{p,l} \Big(u_j + \frac{u_{j+1} - u_j}{T} (s - jT) \Big) ds \\
&- \sum_{j=0}^{k-1} \int_0^T \Phi(jT + \lambda, 0) d\lambda Q_{n,p} B_{p,l} u_{k-j-1} - \sum_{j=0}^{k-1} \int_0^T \Phi(jT + \lambda, 0)(T - \lambda) d\lambda Q_{n,p} B_{p,l} \frac{u_{k-j} - u_{k-j-1}}{T} \\
&= \sum_{j=0}^{k-1} \int_{jT}^{(j+1)T} \Phi(kT, s) Q_{n,p} B_{p,l} u_j ds + \sum_{j=0}^{k-1} \int_{jT}^{(j+1)T} \Phi(kT, s) Q_{n,p} B_{p,l} \frac{u_{j+1} - u_j}{T} (s - jT) ds \\
&- \sum_{j=0}^{k-1} \int_0^T \Phi(jT + \lambda, 0) d\lambda Q_{n,p} B_{p,l} u_{k-j-1} - \sum_{j=0}^{k-1} \int_0^T \Phi(jT + \lambda, 0)(T - \lambda) d\lambda Q_{n,p} B_{p,l} \frac{u_{k-j} - u_{k-j-1}}{T}
\end{aligned}
$$

We now set $s = jT + \lambda$ and we have,

$$
\begin{aligned}
x(kT) - x_k &= \sum_{j=0}^{k-1} \int_0^T \Phi(kT, jT + \lambda) Q_{n,p} B_{p,l} u_j d\lambda + \sum_{j=0}^{k-1} \int_0^T \Phi(kT, jT + \lambda) Q_{n,p} B_{p,l} \frac{u_{j+1} - u_j}{T} \lambda d\lambda \\
&- \sum_{j=0}^{k-1} \int_0^T \Phi(jT + \lambda, 0) d\lambda Q_{n,p} B_{p,l} u_{k-j-1} - \sum_{j=0}^{k-1} \int_0^T \Phi(jT + \lambda, 0)(T - \lambda) d\lambda Q_{n,p} B_{p,l} \frac{u_{k-j} - u_{k-j-1}}{T}
\end{aligned}
$$

and now $i = k - j - 1$ and we get,

$$
\begin{aligned}
x(kT) - x_k &= \sum_{j=0}^{k-1} \int_0^T \Phi(kT, jT+\lambda) d\lambda Q_{n,p} B_{p,l} u_j + \sum_{j=0}^{k-1} \int_0^T \Phi(kT, jT+\lambda) \lambda d\lambda Q_{n,p} B_{p,l} \frac{u_{j+1} - u_j}{T} \\
&- \sum_{i=0}^{k-1} \int_0^T \Phi((k-i-1)T+\lambda, 0) d\lambda Q_{n,p} B_{p,l} u_i - \sum_{i=0}^{k-1} \int_0^T \Phi((k-i-1)T+\lambda, 0)(T-\lambda) d\lambda Q_{n,p} B_{p,l} \times \\
&\quad \times \frac{u_{i+1} - u_i}{T} \\
&\overset{i=j}{=} \sum_{j=0}^{k-1} \int_0^T \Phi((k-j)T-\lambda, 0) d\lambda Q_{n,p} B_{p,l} u_j + \sum_{j=0}^{k-1} \int_0^T \Phi((k-j)T-\lambda, 0) \lambda d\lambda Q_{n,p} B_{p,l} \frac{u_{j+1} - u_j}{T} \\
&- \sum_{j=0}^{k-1} \int_0^T \Phi((k-j-1)T+\lambda, 0) d\lambda Q_{n,p} B_{p,l} u_j - \sum_{j=0}^{k-1} \int_0^T \Phi((k-j-1)T+\lambda, 0)(T-\lambda) d\lambda Q_{n,p} B_{p,l} \times \\
&\quad \times \frac{u_{j+1} - u_j}{T} \\
&= \sum_{j=0}^{k-1} \int_0^T \Big(\Phi((k-j)T-\lambda, 0) - \Phi((k-j-1)T+\lambda, 0) \Big) d\lambda Q_{n,p} B_{p,l} u_j \\
&+ \sum_{j=0}^{k-1} \int_0^T \Big(\Phi((k-j)T-\lambda, 0)\lambda - \Phi((k-j-1)T+\lambda, 0)(T-\lambda) \Big) d\lambda Q_{n,p} B_{p,l} \frac{u_{j+1} - u_j}{T} \\
&= \sum_{j=0}^{k-1} \Bigg(\int_0^T \Big(\Phi((k-j)T, 0)\Phi(-\lambda, 0) - \Phi((k-j)T, 0)\Phi(\lambda - T, 0) \Big) d\lambda Q_{n,p} B_{p,l} u_j \\
&\quad + \int_0^T \Big(\Phi((k-j)T, 0)\Phi(-\lambda, 0)\lambda - \Phi((k-j)T, 0)\Phi(\lambda - T, 0)(T-\lambda) \Big) d\lambda Q_{n,p} B_{p,l} \frac{u_{j+1} - u_j}{T} \Bigg) \\
&= \sum_{j=0}^{k-1} \Bigg[\Phi((k-j)T, 0) \Bigg(\int_0^T \Big(\Phi(-\lambda, 0) - \Phi(\lambda - T, 0) \Big) d\lambda Q_{n,p} B_{p,l} u_j \\
&\quad + \int_0^T \Big(\Phi(-\lambda, 0)\lambda - \Phi(\lambda - T, 0)(T-\lambda) \Big) d\lambda Q_{n,p} B_{p,l} \frac{u_{j+1} - u_j}{T} \Bigg) \Bigg]
\end{aligned}
$$

so,

$$
\begin{aligned}
x(kT) - x_k = \sum_{j=0}^{k-1} \Bigg[\Phi((k-j)T, 0) \Bigg(& \int_0^T \Big(\Phi(-\lambda, 0) \left(1 - \frac{\lambda}{T}\right) - \Phi(\lambda - T, 0) \frac{\lambda}{T} \Big) d\lambda Q_{n,p} B_{p,l} u_j \\
& + \int_0^T \Big(\Phi(-\lambda, 0) \frac{\lambda}{T} - \Phi(\lambda - T, 0) \left(1 - \frac{\lambda}{T}\right) \Big) d\lambda Q_{n,p} B_{p,l} u_{j+1} \Bigg) \Bigg]
\end{aligned}
$$

and we finally get that the upper bound through the discretization process is given by

$$\|x(kT)-x_k\| \le \sum_{j=0}^{k-1}\Bigg[\underbrace{\|\Phi((k-j)T,0)\|}_{\mathbf{A}}\Bigg(\int_0^T \underbrace{\|\Phi(-\lambda,0)\left(1-\frac{\lambda}{T}\right)-\Phi(\lambda-T,0)\frac{\lambda}{T}\|}_{\mathbf{B}}\, d\lambda\|Q_{n,p}\|\|B_{p,l}\|\|u_j\| + \int_0^T \underbrace{\|\Phi(-\lambda,0)\frac{\lambda}{T}-\Phi(\lambda-T,0)\left(1-\frac{\lambda}{T}\right)\|}_{\mathbf{C}}\, d\lambda\|Q_{n,p}\|\|B_{p,l}\|\|u_{j+1}\|\Bigg)\Bigg] \tag{3.1}$$

Lemma 3.1

The following inequality holds

$$\|e^{J_p t}\| \le \sum_{i=1}^{n}\left[e^{a_i t}\left(\sum_{m=0}^{p_i-1}\left(\left(\frac{t^m}{m!}\right)^2(p_i-m)\right)\right)^{\frac{1}{2}}\right] \tag{3.2}$$

Proof:

The matrix J_p is

$$J_p = \begin{bmatrix} J_{p_1}(a_1) & & & \\ & J_{p_2}(a_2) & & \\ & & \ddots & \\ & & & J_{p_n}(a_n) \end{bmatrix}$$

where $a_1, a_2, \ldots, a_n$ are the non zero finite divisors of the matrix $sE-A$.
Moreover,

$$J_{p_i}(a_i) = \begin{bmatrix} a_i & 1 & & \\ & a_i & \ddots & \\ & & \ddots & \vdots \\ & & & a_i \end{bmatrix} \text{ for every } i = 1,2,\ldots,n$$

and

$$e^{J_p t} = \begin{bmatrix} e^{J_{p_1}(a_1)t} & & & \\ & e^{J_{p_2}(a_2)t} & & \\ & & \ddots & \\ & & & e^{J_{p_n}(a_n)t} \end{bmatrix} \text{ with } p_1+p_2+\cdots+p_n = p$$

Now, based on the well-known inequality $\sqrt{x+y} \le \sqrt{x}+\sqrt{y}$, we have

$$\|e^{J_p t}\| \le \|e^{J_{p_1}(a_1)t}\| + \cdots + \|e^{J_{p_n}(a_n)t}\| = \sum_{i=1}^{n}\|e^{J_{p_i}(a_i)t}\|$$

But,

$$\|e^{J_{p_i}(a_i)t}\| = e^{a_i t}\begin{bmatrix} 1 & t & \frac{t^2}{2!} & \cdots & \frac{t^{(p_i-1)}}{(p_i-1)!} \\ 0 & 1 & t & \cdots & \frac{t^{(p_i-2)}}{(p_i-2)!} \\ & & \ldots & \ddots & \vdots \\ & & & \ddots & t \\ & & & & 1 \end{bmatrix}$$

and as a result

$$\|e^{J_{p_i}}(a_i)t\| = \left[(e^{a_i t})^2 p_i + (te^{a_i t})^2(p_i-1) + \left(\frac{t^2}{2!}e^{a_i t}\right)^2 (p_i-2) + \cdots + \left(\frac{t^{p_i-1}}{(p_i-1)!}e^{a_i t}\right)^2 (p_i-(p_i-1))\right]^{\frac{1}{2}}$$

$$= \left[e^{2a_i t}\sum_{m=0}^{p_i-1}\left(\left(\frac{t^m}{m!}\right)^2 (p_i-m)\right)\right]^{\frac{1}{2}} = e^{a_i t}\left[\sum_{m=0}^{p_i-1}\left(\left(\frac{t^m}{m!}\right)^2 (p_i-m)\right)\right]^{\frac{1}{2}}$$

so

$$\|e^{J_p t}\| \le \sum_{i=1}^{n}\|e^{J_{p_i}(a_i)t}\| = \sum_{i=1}^{n}\left[e^{a_i t}\left(\sum_{m=0}^{p_i-1}\left(\left(\frac{t^m}{m!}\right)^2 (p_i-m)\right)\right)^{\frac{1}{2}}\right] \quad \blacktriangle$$

Lemma 3.2

The following inequality holds

$$\|\Phi((k-j)T,0)\| \le \|Q\|\|Q^{-1}\|\left(\left(\sum_{i=1}^{n} e^{a_i T}\left(\sum_{m=0}^{p_i-1}\left(\frac{T^m}{m!}\right)^2 (p_i-m)\right)^{\frac{1}{2}}\right)^{k-j} + \sqrt{q}\right) \tag{3.3}$$

Proof:

We know that

$$\Phi\left((k-j)T,0\right) = Q\begin{bmatrix} e^{J_p(k-j)T} & O_{p,q} \\ O_{q,p} & I_q \end{bmatrix} Q^{-1}$$

Thus, based on the following property of the Euclidean Norm, $\sqrt{x+y} \le \sqrt{x} + \sqrt{y}$, we have

$$\|\Phi\left((k-j)T,0\right)\| \le \|Q\|\left(\|e^{J_p(k-j)T}\| + \|I_q\|\right)\|Q^{-1}\|$$

But

$$\|I_q\| = \sqrt{\underbrace{1^2+1^2+\cdots+1^2}_{q}} = \sqrt{q} \quad \text{and} \quad \|e^{J_p(k-j)T}\| = \|e^{J_p T}e^{J_p T}\cdots e^{J_p T}\| \le \left(\|e^{J_p T}\|\right)^{k-j}$$

therefore

$$\|\Phi\left((k-j)T,0\right)\| \le \|Q\|\|Q^{-1}\|\left(\left(\|e^{J_p T}\|\right)^{k-j} + \sqrt{q}\right)$$

and from relation (3.2) we get what we wanted to prove. $\blacktriangle$

Having these in mind we have that

$$\Phi(-\lambda,0)\left(1-\frac{\lambda}{T}\right) - \Phi(\lambda-T,0)\frac{\lambda}{T} = Q\begin{bmatrix}\left(1-\frac{\lambda}{T}\right)e^{-J_p\lambda} - \frac{\lambda}{T}e^{J_p(\lambda-T)} & O_{p,q} \\ O_{q,p} & \left(1-\frac{2\lambda}{T}\right)I_q\end{bmatrix}Q^{-1}$$

and so

$$\|\Phi(-\lambda,0)\left(1-\frac{\lambda}{T}\right) - \Phi(\lambda-T,0)\frac{\lambda}{T}\| \le \|Q\|\|Q^{-1}\|\left(\left(1-\frac{\lambda}{T}\right)\|e^{-J_p\lambda}\| + \frac{\lambda}{T}\|e^{J_p(\lambda-T)}\| + |1-\frac{2\lambda}{T}|\sqrt{q}\right) \tag{3.4}$$

Similarly,we have

$$\|\Phi(-\lambda,0)\frac{\lambda}{T}-\Phi(\lambda-T,0)\left(1-\frac{\lambda}{T}\right)\|\leq\|Q\|\|Q^{-1}\|\left(\frac{\lambda}{T}\|e^{-J_p\lambda}\|+\left(1-\frac{\lambda}{T}\right)\|e^{J_p(\lambda-T)}\|+|1-\frac{2\lambda}{T}|\sqrt{q}\right) \tag{3.5}$$

Now,using (3.4),(3.5) and (3.3), relation (3.1) is transformed into

$$\begin{aligned}\|x(kT)-x_k\|\leq\sum_{j=0}^{k-1}&\left(\left(\sum_{i=1}^{n}e^{a_iT}\left(\sum_{m=0}^{p_i-1}\left(\frac{T^m}{m!}\right)^2(p_i-m)\right)^{\frac{1}{2}}\right)^{k-j}+\sqrt{q}\right)\times\\&\times\left(\int_0^T\left[\left(1-\frac{\lambda}{T}\right)\|e^{-J_p\lambda}\|+\frac{\lambda}{T}\|e^{J_p(\lambda-T)}\|+|1-\frac{2\lambda}{T}|\sqrt{q}\right]d\lambda\|u_j\|+\right.\\&\left.+\int_0^T\left[\frac{\lambda}{T}\|e^{-J_p\lambda}\|+\left(1-\frac{\lambda}{T}\right)\|e^{J_p(\lambda-T)}\|+|1-\frac{2\lambda}{T}|\sqrt{q}\right]d\lambda\|u_{j+1}\|\right)\|Q\|^2\|Q^{-1}\|^2\|Q_{n,p}\|\|B_{p,l}\|\end{aligned} \tag{3.6}$$

where $\|e^{-J_p\lambda}\|$ and $\|e^{J_p(\lambda-T)}\|$ are substituted from the quantities

$$\sum_{i=1}^{n}e^{-a_i\lambda}\left(\sum_{m=0}^{p_i-1}\left(\left(\frac{(-1)^m\lambda^m}{m!}\right)^2(p_i-m)\right)\right)^{\frac{1}{2}}$$

and

$$\sum_{i=1}^{n}e^{a_i(\lambda-T)}\left(\sum_{m=0}^{p_i-1}\left(\left(\frac{(\lambda-T)^m}{m!}\right)^2(p_i-m)\right)\right)^{\frac{1}{2}}$$

respectively.

Chapter 4

Choosing the Sampling Period

As we can see, relation (3.6) is very complicated, involving many parameters, which makes it not so useful for practical applications. In this section making use of some lemmas and assuming that sampling period $T < 1$ (as this is the case to the most phenomena studied) we conclude to a new upper bound, computationally much more easier.

Lemma 4.1

Assuming $T \in (0,1)$ the following inequality holds

$$\left(\sum_{m=0}^{p_i-1} \left(\frac{T^m}{m!} \right)^2 (p_i - m) \right)^{\frac{1}{2}} \leq p_i \tag{4.1}$$

Proof:

We have,

$$\left(\sum_{m=0}^{p_i-1} \left(\frac{T^m}{m!} \right)^2 (p_i - m) \right)^{\frac{1}{2}} \leq \left(\sum_{m=0}^{p_i-1} \frac{1}{(m!)^2} (p_i - m) \right)^{\frac{1}{2}} < \left(\sum_{m=0}^{p_i-1} p_i \right)^{\frac{1}{2}} = p_i \quad \blacktriangle$$

Lemma 4.2

Given that $T \in (0,1)$ the following inequality holds

$$\sum_{i=1}^{n} \left[\|e^{a_i T}\| \left(\sum_{m=0}^{p_i-1} \left(\frac{T^m}{m!} \right)^2 (p_i - m) \right)^{\frac{1}{2}} \right] \leq p\epsilon_0 \tag{4.2}$$

with $\epsilon_0 = max\left[1, e^{Re(a_i)}, e^{Re(-a_i)}\right], \quad i = 1, \cdots, n \quad \text{and} \quad \sum p_i = p$

Proof:

We have, $\|e^{a_i T}\| = \|e^{(Re(a_i)+iIm(a_i))T}\| = \|e^{Re(a_i)T}\| \leq \epsilon_0$ so from lemma 4.1

$$\sum_{i=1}^{n} \left[\|e^{a_i T}\| \left(\sum_{m=0}^{p_i-1} \left(\frac{T^m}{m!} \right)^2 (p_i - m) \right)^{\frac{1}{2}} \right] \leq \sum_{i=1}^{n} \epsilon_0 p_i = p\epsilon_0 \quad \blacktriangle$$

Lemma 4.3

Given that $\lambda \in (0,1)$ the following inequality holds

$$\|e^{-a_i\lambda}\| \left(\sum_{m=0}^{p_i-1} \left(\frac{(-1)^m \lambda^m}{m!} \right)^2 (p_i - m) \right)^{\frac{1}{2}} \leq \epsilon_0 \frac{\sqrt{p_0}}{\sqrt{1-\lambda^2}} \tag{4.3}$$

with $\epsilon_0 = max\left[1, e^{Re(a_i)}, e^{Re(-a_i)}\right], \quad i = 1, \cdots, n \quad \text{and} \quad p_0 = max[p_i] \;\; i = 1, \cdots, n$

Proof:

We have,

$$\begin{aligned} \|e^{-a_i\lambda}\| \left(\sum_{m=0}^{p_i-1} \left(\frac{(-1)^m \lambda^m}{m!} \right)^2 (p_i - m) \right)^{\frac{1}{2}} &\leq \epsilon_0 \left(\sum_{m=0}^{p_i-1} \left(\frac{(-1)^m \lambda^m}{m!} \right)^2 p_i \right)^{\frac{1}{2}} \\ \leq \epsilon_0 \left(p_i \sum_{m=0}^{p_i-1} \lambda^{2m} \right)^{\frac{1}{2}} &\leq \epsilon_0 \left(p_i \sum_{m=0}^{\infty} \lambda^{2m} \right)^{\frac{1}{2}} = \epsilon_0 \left(p_i \frac{1}{1-\lambda^2} \right)^{\frac{1}{2}} \leq \epsilon_0 \frac{\sqrt{p_0}}{\sqrt{1-\lambda^2}} \quad \blacktriangle \end{aligned}$$

Lemma 4.4

Given that $0 < \lambda < T < 1$ the following inequality holds

$$\|e^{a_i(\lambda-T)}\| \left(\sum_{m=0}^{p_i-1} \left(\frac{(\lambda - T)^m}{m!} \right)^2 (p_i - m) \right)^{\frac{1}{2}} \leq \epsilon_0 \frac{\sqrt{p_0}}{\sqrt{1-(T-\lambda)^2}} \tag{4.4}$$

with $\epsilon_0 = max\left[1, e^{Re(a_i)}, e^{Re(-a_i)}\right], \quad i = 1, \cdots, n \quad \text{and} \quad p_0 = max[p_i] \;\; i = 1, \cdots, n$

Proof:

We have,

$$\begin{aligned} \|e^{a_i(\lambda-T)}\| \left(\sum_{m=0}^{p_i-1} \left(\frac{(\lambda - T)^m}{m!} \right)^2 (p_i - m) \right)^{\frac{1}{2}} \epsilon_0 \left(\sum_{m=0}^{p_i-1} \left(\frac{(\lambda - T)^m}{m!} \right)^2 p_i \right)^{\frac{1}{2}} &\leq \epsilon_0 \left(p_i \sum_{m=0}^{p_i-1} (\lambda - T)^{2m} \right)^{\frac{1}{2}} \\ \leq \epsilon_0 \left(p_i \sum_{m=0}^{\infty} (\lambda - T)^{2m} \right)^{\frac{1}{2}} &= \epsilon_0 \left(p_i \frac{1}{1-(\lambda-T)^2} \right)^{\frac{1}{2}} \leq \epsilon_0 \frac{\sqrt{p_0}}{\sqrt{1-(\lambda-T)^2}} \quad \blacktriangle \end{aligned}$$

Having in mind these lemmas, we can now prove the new upper bound formula.

Theorem 4.1

Setting $u_1 = \underset{j=0,\cdots,k-1}{max} \|u_j\|$ and $u_2 = \underset{j=0,\cdots,k-1}{max} \|u_{j+1}\|$ we have:

$$\begin{aligned} \|x(kT) - x_k\| \leq n\epsilon_0\sqrt{p_0} \left(\frac{(p\epsilon_0)^{k+1} - p\epsilon_0}{p\epsilon_0 - 1} + k\sqrt{q} \right) \left(\frac{T}{2}\sqrt{q}(u_1+u_2) + 2u_1 \arcsin T + 2(u_1-u_2)\frac{\sqrt{1-T^2}-1}{T} \right) \\ \times \|Q\|^2 \|Q^{-1}\|^2 \|Q_{n,p}\| \|B_{p,l}\| \end{aligned} \tag{4.5}$$

Proof:

From (4.1),(4.2),(4.3) and (4.4), relation (3.6) is transformed into

$$\|x(kT)-x_k\| \le \sum_{j=0}^{k-1}\left[\left((p\epsilon_0)^{k-j}+\sqrt{q}\right)\left(\int_0^T\left[\left(1-\frac{\lambda}{T}\right)\frac{n\epsilon_0\sqrt{p_0}}{\sqrt{1-\lambda^2}}+\frac{\lambda}{T}\frac{n\epsilon_0\sqrt{p_0}}{\sqrt{1-(\lambda-T)^2}}+|1-\frac{2\lambda}{T}|\sqrt{q}\right]d\lambda\|u_j\|\right.\right.$$
$$\left.\left.+\int_0^T\left[\frac{\lambda}{T}\frac{n\epsilon_0\sqrt{p_0}}{\sqrt{1-\lambda^2}}+\left(1-\frac{\lambda}{T}\right)\frac{n\epsilon_0\sqrt{p_0}}{\sqrt{1-(\lambda-T)^2}}+|1-\frac{2\lambda}{T}|\sqrt{q}\right]d\lambda\|u_{j+1}\|\right)\right]\|Q\|^2\|Q^{-1}\|^2\|Q_{n,p}\|\|B_{p,l}\| \tag{4.6}$$

But,

$$\int_0^T\left(1-\frac{\lambda}{T}\right)\frac{1}{\sqrt{1-\lambda^2}}d\lambda=\int_0^T\frac{1}{\sqrt{1-\lambda^2}}d\lambda-\int_0^T\frac{\lambda}{T}\frac{1}{\sqrt{1-\lambda^2}}=$$
$$=\arcsin T-\arcsin 0+\frac{1}{T}\left(\sqrt{1-T^2}-1\right)=\arcsin T+\frac{\sqrt{1-T^2}-1}{T} \tag{4.7}$$

$$\int_0^T\frac{\lambda}{T}\frac{1}{\sqrt{1-\lambda^2}}d\lambda\overset{u=\lambda-T}{=}\int_{-T}^0\frac{u+T}{T}\frac{1}{\sqrt{1-u^2}}du=\int_{-T}^0\frac{u}{T}\frac{1}{\sqrt{1-u^2}}du+\int_{-T}^0\frac{1}{\sqrt{1-u^2}}$$
$$=-\frac{1}{T}\int_{-T}^0\frac{-u}{\sqrt{1-u^2}}du+\int_{-T}^0\frac{1}{\sqrt{1-u^2}}du=-\frac{1}{T}\left(1-\sqrt{1-T^2}\right)+\arcsin 0-\arcsin(-T)$$
$$=\frac{\sqrt{1-T^2}-1}{T}+\arcsin T \tag{4.8}$$

$$\int_0^T\frac{\lambda}{T}\frac{1}{\sqrt{1-\lambda^2}}d\lambda=-\frac{1}{T}\int_0^T\frac{-\lambda}{\sqrt{1-\lambda^2}}d\lambda=-\frac{1}{T}\left(\sqrt{1-T^2}-1\right)=\frac{1-\sqrt{1-T^2}}{T} \tag{4.9}$$

and

$$\int_0^T\left(1-\frac{\lambda}{T}\right)\frac{1}{\sqrt{1-(\lambda-T)^2}}d\lambda=\int_0^T\frac{1}{\sqrt{1-(\lambda-T)^2}}d\lambda-\int_0^T\frac{\lambda}{T}\frac{1}{\sqrt{1-(\lambda-T)^2}}=$$
$$=\arcsin T-\frac{1}{T}\left(\sqrt{1-T^2}-1+T\arcsin T\right)=\frac{1-\sqrt{1-T^2}}{T} \tag{4.10}$$

Also,

$$\sum_{j=0}^{k-1}\left((p\epsilon_0)^{k-j}+\sqrt{q}\right)=\frac{(p\epsilon_0)^{k+1}-p\epsilon_0}{p\epsilon_0-1}+k\sqrt{q} \tag{4.11}$$

and

$$\int_0^T|1-\frac{2\lambda}{T}|d\lambda=\int_0^{\frac{T}{2}}\left(1-\frac{2\lambda}{T}\right)d\lambda+\int_{\frac{T}{2}}^T\left(\frac{2\lambda}{T}-1\right)d\lambda=\frac{T}{4}+\frac{T}{4}=\frac{T}{2} \tag{4.12}$$

We set now, $u_1 = \underset{j=0,\cdots,k-1}{max}\|u_j\|$ and $u_2 = \underset{j=0,\cdots,k-1}{max}\|u_{j+1}\|$, and from (4.6) we have,

$$\|x(kT) - x_k\| \leq n\epsilon_0\sqrt{p_0}\sum_{j=0}^{k-1}\left[\left((p\epsilon_0)^{k-j} + \sqrt{q}\right)\right]\times$$
$$\times\left(\int_0^T\left[\left(1-\frac{\lambda}{T}\right)\frac{1}{\sqrt{1-\lambda^2}} + \frac{\lambda}{T}\frac{1}{\sqrt{1-(\lambda-T)^2}} + |1-\frac{2\lambda}{T}|\sqrt{q}\right]d\lambda u_1\right.$$
$$\left.+\int_0^T\left[\frac{\lambda}{T}\frac{1}{\sqrt{1-\lambda^2}} + \left(1-\frac{\lambda}{T}\right)\frac{1}{\sqrt{1-(\lambda-T)^2}} + |1-\frac{2\lambda}{T}|\sqrt{q}\right]d\lambda u_2\right)\|Q\|^2\|Q^{-1}\|^2\|Q_{n,p}\|\|B_{p,l}\| \tag{4.13}$$

and now using (4.7), (4.8), (4.9), (4.10), (4.11) and (4.12) into (4.13) we get

$$\|x(kT) - x_k\| \leq n\epsilon_0\sqrt{p_0}\left(\frac{(p\epsilon_0)^{k+1} - p\epsilon_0}{p\epsilon_0 - 1} + k\sqrt{q}\right)\left[\left(2\arcsin T + 2\frac{\sqrt{1-T^2}-1}{T} + \frac{T}{2}\sqrt{q}\right)u_1\right.$$
$$\left.+\left(2\frac{1-\sqrt{1-T^2}}{T} + \frac{T}{2}\sqrt{q}\right)u_2\right]\|Q\|^2\|Q^{-1}\|^2\|Q_{n,p}\|\|B_{p,l}\|$$
$$= n\epsilon_0\sqrt{p_0}\left(\frac{(p\epsilon_0)^{k+1} - p\epsilon_0}{p\epsilon_0 - 1} + k\sqrt{q}\right)\left(\frac{T}{2}\sqrt{q}(u_1+u_2) + 2u_1\arcsin T + 2(u_1-u_2)\frac{\sqrt{1-T^2}-1}{T}\right)$$
$$\times\|Q\|^2\|Q^{-1}\|^2\|Q_{n,p}\|\|B_{p,l}\| \quad \blacktriangle$$

Chapter 5

Evaluating the Results of Error Analysis via an Example

Let us now consider a system of the form $E\dot{x}(t) = Ax(t) + Bu(t)$, with

$$E = \begin{bmatrix} -1.5 & 2 & 1.5 & 0.5 \\ 0.5 & 0 & -0.5 & -0.5 \\ 0.5 & -1 & -0.5 & 0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 0 & -1 & 1 \\ 0.5 & 0 & -0.5 & -0.5 \\ -0.5 & 1 & 1.5 & -0.5 \\ 0.5 & -1 & -0.5 & 0.5 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 1 \\ 2 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$$

that is

$$\begin{bmatrix} -1.5 & 2 & 1.5 & 0.5 \\ 0.5 & 0 & -0.5 & -0.5 \\ 0.5 & -1 & -0.5 & 0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 1 \\ 0.5 & 0 & -0.5 & -0.5 \\ -0.5 & 1 & 1.5 & -0.5 \\ 0.5 & -1 & -0.5 & 0.5 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 2 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \quad (5.1)$$

Then, there exist nonsingular matrices

$$P = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

such that

$$PEQ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad PAQ = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For system (5.1) we have $p = q = 2 = p_0$, $n = 1$, $\epsilon_0 = e$. Assume also that $u_1 = \max\limits_{j=0,\cdots,k-1} \|u_j\| = 1$ and $u_2 = \max\limits_{j=0,\cdots,k-1} \|u_{j+1}\| = 1$. Moreover,

$$Q_{4,2} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad B_{2,2} = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} \quad \text{and} \quad Q^{-1} = \begin{bmatrix} -0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0 & -0.5 & -0.5 \\ 0 & 0 & 1 & 0 \\ 0.5 & -1 & -0.5 & 0.5 \end{bmatrix}$$

Therefore, $\|Q\| = 2\sqrt{3} \approx 3.464$, $\|Q^{-1}\| = \sqrt{\frac{21}{4}} \approx 2.291$, $\|Q_{4,2}\| = \sqrt{8} \approx 2.828$ and $\|B_{2,2}\| = \sqrt{6} \approx 2.449$

If we also want the error to not exceed 10^{-1}, we get

$$11495.04\left(\sqrt{2}T + 2\arcsin T\right) \leq 0.1$$

and solving for $T \in (0,1)$ we get $0 < T < 2.548 \times 10^{-6}$ and as a result we can choose $T = 2 \times 10^{-6}$.

In this way, setting a restriction for the maximum error, we can find the sampling period T which guarantees us that this goal is achieved.

Last thing to do, is to compare our new upper bound to the previous one. In [2] the error bound was said to be

$$2kn\epsilon_0\sqrt{p_0}\arcsin T(p^k\epsilon_0^k + \sqrt{q})u_{max}\|Q\|^2\|Q^{-1}\|^2\|Q_{n,p}\|\|B_{p,l}\|$$

In order to compare these quantities we ignore the common factor $\|Q\|^2\|Q^{-1}\|^2\|Q_{n,p}\|\|B_{p,l}\|$. In the following table we see the two bounds for $T = 10^{-6}$. Moreover, we distinguish cases ($u_1 = u_2$, $u_1 > u_2$, $u_1 < u_2$) where $u_{max} = max[u_1, u_2]$. As we can see,although for $k = 1$ old upper bound is sharper than the new one, for $k \geq 2$ the new bound is sharper that the old one.

Case $u_1 = u_2 = 1$		
k	ZOH	FOH
1	0.000052671	0.000089916
2	0.00047623	0.00049640
3	0.00373886	0.00262395
4	0.0269091	0.0141081
5	0.182625	0.076460

Case $u_1 = 2 > 1 = u_2$		
k	ZOH	FOH
1	0.000105344	0.000134878
2	0.00095246	0.00074462
3	0.00747773	0.00393603
4	0.0538183	0.0211628
5	0.365251	0.114694

Case $u_2 = 2 > 1 = u_1$		
k	ZOH	FOH
1	0.000105344	0.000134871
2	0.00095246	0.00074458
3	0.00747773	0.00393582
4	0.0538183	0.0211617
5	0.365251	0.114688

Conclusion

This discretization technique using first order hold discretization for input function $u(t)$ provides us with two interesting results. First of all it allows us to estimate the sampling period T for which the upper bound do not exceed a given value. Afterwards it provides us with a sharper upper bound and as we can see from the tables above in the new, First Order Hold approximation, we can get a better result from the already known Zero Order Hold approximation. This extends in some way the already known theoretical results for the known upper bound of the norm of the difference between continuous and discrete time solution.Although, we should mention that in fact, simulation shows

that the difference between continuous and discrete solution is much smaller (see appendix) even if the theoretical error bound seems to grow up rapidly. Towards this way, research for finding an even sharper upper bound for the error of the discretization process remains a challenge.

Bibliography

[1] L. Dai, Singular Control Systems, vol. 118 of Lecture Notes in Control and Information Sciences, Springer, Berlin, Germany, 1989.

[2] A. D. Karageorgos, A. A. Pantelous, and G. I. Kalogeropoulos, "Designing the Sampling Period of a Descretized LTI Descriptor (Regular)System with Inputs", International Journal of Control, Automation, and Systems,2011.

[3] A. D. Karageorgos, A. A. Pantelous, and G. I. Kalogeropoulos, "Discretizing LTI descriptor (regular) differential input systems with consistent initial conditions", Advances in Decision Sciences, vol. 2010, pp. 1-19, 2010.

[4] N.P.Karampetakis, *"On the discretization of singular systems"*, IMA Journal of Mathematical Control and Information, vol.21, pp.223-242, 2004.

[5] N.P.Karampetakis, A.Gregoriadou *"On a first order hold discretization for singular systems"*, International Conference on Communications, Computing and Control Applications (CCCA), pp.1-6, 2011.

[6] Koumboulis F.N. and Mertzios B.G., 1999, On Kalman's controllability and observability criteria for singular systems, Circuit Systems and Signal Process, Vol.18, No.3, pp.269-290.

Appendix

Mathematica Code

In the following pages we present our Mathematica code for getting the discretized models from a continuous model. Also, in the results presented below, the matrices of these state space systems are calculated and the plots between continuous and various discrete-type solution plots are produced.

```
(* :Title: Methods of Discretizations of Continuous Systems *)
(* :Author: R.Karamichalis *)
(* :Summary: This notebook provides 4 main methods for discretizing a
continuous singular system, namely ZOH, BZOH, TFOH, BFOH and for each
method there is a modification of it *)
(* :Created with: © Wolfram Mathematica 7.0 *)
```

```
StringJoin["Last modified at ", DateString[]]
Der[u_, k_, T_, r_] := (Der[u, k + 1, T, r - 1] - Der[u, k, T, r - 1])/T;
Der[u_, k_, T_, 0] := {input[k*T]};
DerBack[u_, k_, T_, r_] := (DerBack[u, k, T, r - 1] - DerBack[u, k - 1, T, r - 1])/T;
DerBack[u_, k_, T_, 0] := {input[k*T]};
Needs["PlotLegends`"];
```

```
myprint[x_] := Print[Text[Style[x, Black, Bold, 15]]];
printprec = 20; (* =min[printprec,$MachinePrecision] *)
StringJoin["Allowed precision in arbitrary-precision numbers= ",
 ToString[$MaxPrecision], " and precision used for machine-precision numbers= ",
 ToString[$MachinePrecision], " decimal digits. This code will print ",
 ToString[Min[printprec, $MachinePrecision]], " digits."]
```

```
(* ZOH *)
ZOH[pe_, pa_, pb_, x0minus_, T_] :=
  Module[{ps, psi, se, mi, fis, ah, bis, i, j, x0}, ps = s*pe - pa;
   psi = Inverse[ps]; se = Series[psi, {s, Infinity, 2}];
   mi = Max[Exponent[se, s]]; fis = {Coefficient[se, s^(-1)]};
   For[i = 0, i <= mi,
    fis = Append[fis, Coefficient[se*s, s^(i + 1)]]; i++];
    ah = Simplify[MatrixExp[fis[[1]].pa*T]];
    bis = {Simplify[Integrate[MatrixExp[fis[[1]].pa*w], {w, 0, T}].fis[[1]].pb +
    Sum[(-1)^i*fis[[i + 1]].pb*T^(1 - i), {i, 1, mi + 1}]] //Expand};
   For[j = 1, j <= mi + 1,
    bis = Append[bis,Sum[(-1)^(i - j)*fis[[i + 1]].pb*T^(1 - i)*
    Binomial[i, i - j], {i, j, mi + 1}]]; j++;];
    x0 = fis[[1]].pe.x0minus + Sum[(MatrixPower[-fis[[2]].pe, i]).fis[[2]].pb.
    {D[input[t], {t, i}] /. t -> 0}, {i, 0, mi}];
   Return[{ah, Sum[bis[[i + 1]]*s^i, {i, 0, Dimensions[bis][[1]] - 1}],
     Table[bis[[i + 1]], {i, 0, Dimensions[bis][[1]] - 1}], x0}];];
```

```
(* ZOH-approx *)
ZOHapprox[pe_, pa_, pb_, x0minus_, T_] :=
  Module[{ps, psi, se, mi, fis, ah, bis, i, j, x0}, ps = s*pe - pa;
   psi = Inverse[ps]; se = Series[psi, {s, Infinity, 2}];
   mi = Max[Exponent[se, s]]; fis = {Coefficient[se, s^(-1)]};
   For[i = 0, i <= mi,
    fis = Append[fis, Coefficient[se*s, s^(i + 1)]]; i++];
    ah = Simplify[MatrixExp[fis[[1]].pa*T]];
    bis = {Simplify[Integrate[MatrixExp[fis[[1]].pa*w], {w, 0, T}].fis[[1]].pb +
    Sum[(-1)^i*fis[[i + 1]].pb*T^(1 - i), {i, 1, mi + 1}]] //Expand};
   For[j = 1, j <= mi + 1,
    bis = Append[bis,Sum[(-1)^(i - j)*fis[[i + 1]].pb*T^(1 - i)*
    Binomial[i, i - j], {i, j, mi + 1}]]; j++;];
    x0 = fis[[1]].pe.x0minus + Sum[(MatrixPower[-fis[[2]].pe, i]).fis[[2]].pb.
    Der[u, 0, T, i], {i, 0, mi}];
   Return[{ah, Sum[bis[[i + 1]]*s^i, {i, 0, Dimensions[bis][[1]] - 1}],
     Table[bis[[i + 1]], {i, 0, Dimensions[bis][[1]] - 1}], x0}];];
```

```
(* BZOH *)
BZOH[pe_, pa_, pb_, x0minus_, T_] :=
  Module[{ps, psi, se, mi, fis, ah, bis, i, j, x0, b0},
   ps = s*pe - pa; psi = Inverse[ps];
   se = Series[psi, {s, Infinity, 2}]; mi = Max[Exponent[se, s]];
   fis = {Coefficient[se, s^(-1)]};
   For[i = 0, i <= mi,
    fis = Append[fis, Coefficient[se*s, s^(i + 1)]]; i++];
    ah = Simplify[MatrixExp[fis[[1]].pa*T]];
   bis = {Simplify[Sum[fis[[i + 1]].pb*T^(1 - i), {i, 1, mi + 1}]] //Expand};
    b0 = Simplify[Integrate[MatrixExp[fis[[1]].pa*w], {w, 0, T}].fis[[1]].pb];
    bis = Append[bis, Simplify[b0 + Sum[(-1)*fis[[i + 1]].pb*T^(1 - i)*
    Binomial[i, 1], {i, 1, mi + 1}]]];
   For[j = 1, j <= mi,
    bis = Append[bis, Sum[(-1)^(j + 1)*fis[[i + 1]].pb*T^(1 - i)*
    Binomial[i, i - (j + 1)], {i, j + 1, mi + 1}]]; j++;];
    x0 = fis[[1]].pe.x0minus + Sum[(MatrixPower[-fis[[2]].pe, i]).fis[[2]].pb.
    {D[input[t], {t, i}] /. t -> 0}, {i, 0, mi}];
   Return[{ah, Sum[bis[[i + 2]]*s^i, {i, -1, Dimensions[bis][[1]] - 2}],
     Table[bis[[i + 2]], {i, -1, Dimensions[bis][[1]] - 2}], N[x0]}];];
```

```
(* BZOH-approx *)
BZOHapprox[pe_, pa_, pb_, x0minus_, T_] :=
  Module[{ps, psi, se, mi, fis, ah, bis, i, j, x0, b0},
   ps = s*pe - pa; psi = Inverse[ps];
   se = Series[psi, {s, Infinity, 2}]; mi = Max[Exponent[se, s]];
   fis = {Coefficient[se, s^(-1)]};
   For[i = 0, i <= mi,
    fis = Append[fis, Coefficient[se*s, s^(i + 1)]]; i++];
    ah = Simplify[MatrixExp[fis[[1]].pa*T]];
    bis = {Simplify[Sum[fis[[i + 1]].pb*T^(1 - i), {i, 1, mi + 1}]] //Expand};
    b0 = Simplify[Integrate[MatrixExp[fis[[1]].pa*w], {w, 0, T}].fis[[1]].pb];
    bis = Append[bis, Simplify[b0 + Sum[(-1)*fis[[i + 1]].pb*T^(1 - i)*
    Binomial[i, 1], {i, 1, mi + 1}]]];
   For[j = 1, j <= mi,
    bis = Append[bis, Sum[(-1)^(j + 1)*fis[[i + 1]].pb*T^(1 - i)*
    Binomial[i, i - (j + 1)], {i, j + 1, mi + 1}]]; j++;];
    x0 = fis[[1]].pe.x0minus + Sum[(MatrixPower[-fis[[2]].pe, i]).fis[[2]].pb.
    DerBack[u, 0, T, i], {i, 0, mi}];
   Return[{ah, Sum[bis[[i + 2]]*s^i, {i, -1, Dimensions[bis][[1]] - 2}],
     Table[bis[[i + 2]], {i, -1, Dimensions[bis][[1]] - 2}], N[x0]}];];
```

```
(* TFOH *)
TFOH[pe_, pa_, pb_, x0minus_, T_] :=
  Module[{ps, psi, se, mi, ah, bis, fis, i, j, x0}, ps = s*pe - pa;
   psi = Inverse[ps]; se = Series[psi, {s, Infinity, 2}];
   mi = Max[Exponent[se, s]]; fis = {Coefficient[se, s^(-1)]};
   For[i = 0, i <= mi,
    fis = Append[fis, Coefficient[se*s, s^(i + 1)]]; i++];
    ah = Simplify[MatrixExp[fis[[1]].pa*T]];
    bis = {Simplify[Integrate[MatrixExp[fis[[1]].pa*w]*w/T, {w, 0, T}].fis[[1]].pb +
    Sum[(-1)^i*fis[[i + 1]].pb*T^(1 - i), {i, 1, mi + 1}]] //Expand};
    bis = Append[bis, Simplify[Integrate[MatrixExp[fis[[1]].pa*w]*(1 - w/T), {w, 0, T}].
    fis[[1]].pb + Sum[(-1)^(i - 1)*i*fis[[i + 1]].pb*T^(1 - i), {i, 1, mi + 1}]]];
   For[j = 2, j <= mi + 1,
    bis = Append[bis, Sum[(-1)^(i - j)*fis[[i + 1]].pb*T^(1 - i)*
    Binomial[i, i - j], {i, j, mi + 1}]]; j++;];
    x0 = fis[[1]].pe.x0minus + Sum[(MatrixPower[-fis[[2]].pe, i]).fis[[2]].pb.
    {D[input[t], {t, i}] /. t -> 0}, {i, 0, mi}];
   Return[{ah, Sum[bis[[i + 1]]*s^i, {i, 0, Dimensions[bis][[1]] - 1}],
     Table[bis[[i + 1]], {i, 0, Dimensions[bis][[1]] - 1}], x0}];];
```

```
(* TFOH-approx *)
TFOHapprox[pe_, pa_, pb_, x0minus_, T_] :=
  Module[{ps, psi, se, mi, ah, bis, fis, i, j, x0}, ps = s*pe - pa;
   psi = Inverse[ps]; se = Series[psi, {s, Infinity, 2}];
   mi = Max[Exponent[se, s]]; fis = {Coefficient[se, s^(-1)]};
   For[i = 0, i <= mi,
    fis = Append[fis, Coefficient[se*s, s^(i + 1)]]; i++];
    ah = Simplify[MatrixExp[fis[[1]].pa*T]];
    bis = {Simplify[Integrate[MatrixExp[fis[[1]].pa*w]*w/T, {w, 0, T}].fis[[1]].pb +
    Sum[(-1)^i*fis[[i + 1]].pb*T^(1 - i), {i, 1, mi + 1}]] //Expand};
    bis = Append[bis, Simplify[Integrate[MatrixExp[fis[[1]].pa*w]*(1 - w/T), {w, 0, T}].
    fis[[1]].pb + Sum[(-1)^(i - 1)*i*fis[[i + 1]].pb*T^(1 - i), {i, 1, mi + 1}]]];
   For[j = 2, j <= mi + 1,
    bis = Append[bis, Sum[(-1)^(i - j)*fis[[i + 1]].pb*T^(1 - i)*
    Binomial[i, i - j], {i, j, mi + 1}]]; j++;];
    x0 = fis[[1]].pe.x0minus + Sum[(MatrixPower[-fis[[2]].pe, i]).fis[[2]].pb.
    Der[u, 0, T, i], {i, 0, mi}];
   Return[{ah, Sum[bis[[i + 1]]*s^i, {i, 0, Dimensions[bis][[1]] - 1}],
     Table[bis[[i + 1]], {i, 0, Dimensions[bis][[1]] - 1}], x0}];];
```

```
(* BFOH *)
BFOH[pe_, pa_, pb_, x0minus_, T_] :=
  Module[{ps, psi, se, mi, ah, bis, fis, i, j, x0}, ps = s*pe - pa;
   psi = Inverse[ps]; se = Series[psi, {s, Infinity, 2}];
   mi = Max[Exponent[se, s]];
   fis = {Coefficient[se, s^(-1)]};
   For[i = 0, i <= mi,
    fis = Append[fis, Coefficient[se*s, s^(i + 1)]]; i++];
    ah = Simplify[MatrixExp[fis[[1]].pa*T]];
    bis = {Simplify[Sum[fis[[i + 1]].pb*T^(1 - i), {i, 1, mi + 1}]]};
    bis = Append[bis,Simplify[Integrate[MatrixExp[fis[[1]].pa*w]*(2 - w/T), {w, 0, T}].
    fis[[1]].pb + Sum[(-1)*fis[[i + 1]].pb*T^(1 - i)*Binomial[i, i - 1], {i, 1, mi + 1}]]];
    bis = Append[bis, Simplify[Integrate[MatrixExp[fis[[1]].pa*w]*((w/T) - 1), {w, 0, T}].
    fis[[1]].pb + Sum[fis[[i + 1]].pb*T^(1 - i)*Binomial[i, i - 2], {i, 2, mi + 1}]]];
   For[j = 2, j <= mi,
    bis = Append[bis, Sum[(-1)^(j + 1)*fis[[i + 1]].pb*T^(1 - i)*
    Binomial[i, i - (j + 1)], {i, j + 1, mi + 1}]]; j++;];
    x0 = fis[[1]].pe.x0minus + Sum[(MatrixPower[-fis[[2]].pe, i]).fis[[2]].pb.
    {D[input[t], {t, i}] /. t -> 0}, {i, 0, mi}];
   Return[{ah, Sum[bis[[i + 2]]*s^i, {i, -1, Dimensions[bis][[1]] - 2}],
     Table[bis[[i + 2]], {i, -1, Dimensions[bis][[1]] - 2}], x0}];];
```

```
(* BFOH-approx*)
BFOHapprox[pe_, pa_, pb_, x0minus_, T_] :=
  Module[{ps, psi, se, mi, ah, bis, fis, i, j, x0}, ps = s*pe - pa;
   psi = Inverse[ps]; se = Series[psi, {s, Infinity, 2}];
   mi = Max[Exponent[se, s]];
   fis = {Coefficient[se, s^(-1)]};
   For[i = 0, i <= mi,
    fis = Append[fis, Coefficient[se*s, s^(i + 1)]]; i++];
    ah = Simplify[MatrixExp[fis[[1]].pa*T]];
    bis = {Simplify[Sum[fis[[i + 1]].pb*T^(1 - i), {i, 1, mi + 1}]]};
    bis = Append[bis, Simplify[Integrate[MatrixExp[fis[[1]].pa*w]*(2 - w/T), {w, 0, T}].
    fis[[1]].pb + Sum[(-1)*fis[[i + 1]].pb*T^(1 - i)*Binomial[i, i - 1], {i, 1, mi + 1}]]];
    bis = Append[bis, Simplify[Integrate[MatrixExp[fis[[1]].pa*w]*((w/T) - 1), {w, 0, T}].
    fis[[1]].pb + Sum[fis[[i + 1]].pb*T^(1 - i)*Binomial[i, i - 2], {i, 2, mi + 1}]]];
   For[j = 2, j <= mi,
    bis = Append[bis, Sum[(-1)^(j + 1)*fis[[i + 1]].pb*T^(1 - i)*
    Binomial[i, i - (j + 1)], {i, j + 1, mi + 1}]]; j++;];
    x0 = fis[[1]].pe.x0minus + Sum[(MatrixPower[-fis[[2]].pe, i]).fis[[2]].pb.
    DerBack[u, 0, T, i], {i, 0, mi}];
   Return[{ah, Sum[bis[[i + 2]]*s^i, {i, -1, Dimensions[bis][[1]] - 2}],
     Table[bis[[i + 2]], {i, -1, Dimensions[bis][[1]] - 2}], x0}];];
```

```
myanalysis[pe_, pa_, pb_, x0minus_, T_, PlithosVimaton_, whichmodels_, matrices_] :=
{

  (* ZOH *)
  If[whichmodels[[1]] == 1,
   name = "ZOH"; res = ZOH[pe, pa, pb, x0minus, T];
   zohalfa = res[[1]]; zohbeta = res[[2]];
   zohbetaval = Transpose[res[[3]]]; zohx0 = res[[4]];
   ord = Dimensions[zohbetaval][[2]]; xlast = Flatten[zohx0];
   zohsequence = {xlast};
   For[pl = 1, pl <= PlithosVimaton, pl++, xnew = zohalfa.xlast;
    For[tmp = 1, tmp <= ord, tmp++,
     xnew = xnew +
       Transpose[zohbetaval][[tmp]].{input[(pl - 1 + tmp - 1)*T]}];
    xlast = xnew; zohsequence = Append[zohsequence, xlast];
    ]; If[matrices, myprint[StringJoin[name, " matrices:"]];
    Print["\!\(\*OverscriptBox[\"A\", \"~\"]\)=",
     NumberForm[MatrixForm[zohalfa], printprec]];
    Print["\!\(\*OverscriptBox[\"B\", \"~\"]\)=",
     NumberForm[zohbeta // MatrixForm, printprec]];
    Print["\!\(\*SubscriptBox[OverscriptBox[\"B\", \"~\"], \"i\"]\)=",
      NumberForm[zohbetaval // MatrixForm, printprec]];
    Print["\!\(\*SubscriptBox[\"x\", \"0\"]\)=",
     NumberForm[zohx0 // MatrixForm, printprec]];
    myprint[StringJoin[name, " results:"]];
    Print[NumberForm[zohsequence // MatrixForm, printprec]];]];

  (* ZOH-approx *)
  If[whichmodels[[2]] == 1, name = "ZOH-approx";
   res = ZOHapprox[pe, pa, pb, x0minus, T];
   zohapproxalfa = res[[1]]; zohapproxbeta = res[[2]];
   zohapproxbetaval = Transpose[res[[3]]]; zohapproxx0 = res[[4]];
   ord = Dimensions[zohapproxbetaval][[2]];
   xlast = Flatten[zohapproxx0]; zohapproxsequence = {xlast};
   For[pl = 1, pl <= PlithosVimaton, pl++, xnew = zohapproxalfa.xlast;
    For[tmp = 1, tmp <= ord, tmp++,
     xnew = xnew +
       Transpose[zohapproxbetaval][[tmp]].Der[u, pl - 1 + tmp - 1, T,
         0]];
    xlast = xnew; zohapproxsequence = Append[zohapproxsequence, xlast];
    ]; If[matrices, myprint[StringJoin[name, " matrices:"]];
    Print["\!\(\*OverscriptBox[\"A\", \"~\"]\)=",
     NumberForm[MatrixForm[zohapproxalfa], printprec]];
    Print["\!\(\*OverscriptBox[\"B\", \"~\"]\)=",
     NumberForm[zohapproxbeta // MatrixForm, printprec]];
    Print["\!\(\*SubscriptBox[OverscriptBox[\"B\", \"~\"], \"i\"]\)=",
      NumberForm[zohapproxbetaval // MatrixForm, printprec]];
    Print["\!\(\*SubscriptBox[\"x\", \"0\"]\)=",
     NumberForm[zohapproxx0 // MatrixForm, printprec]];
    myprint[StringJoin[name, " results:"]];
    Print[NumberForm[zohapproxsequence // MatrixForm, printprec]];]];
```

```
(* BZOH *)
If[whichmodels[[3]] == 1, name = "BZOH";
  res = BZOH[pe, pa, pb, x0minus, T];
  bzohalfa = res[[1]]; bzohbeta = res[[2]];
  bzohbetaval = Transpose[res[[3]]]; bzohx0 = res[[4]];
  ord = Dimensions[bzohbetaval][[2]]; xlast = Flatten[bzohx0];
  bzohsequence = {xlast};
  For[pl = 1, pl <= PlithosVimaton, pl++, xnew = bzohalfa.xlast;
   For[tmp = 1, tmp <= ord, tmp++,
    xnew = xnew +
      Transpose[bzohbetaval][[tmp]].{input[(pl - (tmp - 1))*T]}];
   xlast = xnew; bzohsequence = Append[bzohsequence, xlast];
   ]; If[matrices, myprint[StringJoin[name, " matrices:"]];
   Print["\!\(\*OverscriptBox[\"A\", \"~\"]\)=",
    NumberForm[MatrixForm[bzohalfa], printprec]];
   Print["\!\(\*OverscriptBox[\"B\", \"~\"]\)=",
    NumberForm[bzohbeta // MatrixForm, printprec]];
   Print["\!\(\*SubscriptBox[OverscriptBox[\"B\", \"~\"], \"i\"]\)=",
    NumberForm[bzohbetaval // MatrixForm, printprec]];
   Print["\!\(\*SubscriptBox[\"x\", \"0\"]\)=",
    NumberForm[bzohx0 // MatrixForm, printprec]];
   myprint[StringJoin[name, " results:"]];
   Print[NumberForm[bzohsequence // MatrixForm, printprec]];]];



(* BZOH-approx *)
If[whichmodels[[4]] == 1, name = "BZOH-approx";
  res = BZOHapprox[pe, pa, pb, x0minus, T];
  bzohapproxalfa = res[[1]]; bzohapproxbeta = res[[2]];
  bzohapproxbetaval = Transpose[res[[3]]]; bzohapproxx0 = res[[4]];
  ord = Dimensions[bzohapproxbetaval][[2]];
  xlast = Flatten[bzohapproxx0]; bzohapproxsequence = {xlast};
  For[pl = 1, pl <= PlithosVimaton, pl++, xnew = bzohapproxalfa.xlast;
   For[tmp = 1, tmp <= ord, tmp++,
    xnew = xnew +
      Transpose[bzohapproxbetaval][[tmp]].DerBack[u, pl - (tmp - 1),
        T, 0]];
   xlast = xnew;
   bzohapproxsequence = Append[bzohapproxsequence, xlast];
   ]; If[matrices, myprint[StringJoin[name, " matrices:"]];
   Print["\!\(\*OverscriptBox[\"A\", \"~\"]\)=",
    NumberForm[MatrixForm[bzohapproxalfa], printprec]];
   Print["\!\(\*OverscriptBox[\"B\", \"~\"]\)=",
    NumberForm[bzohapproxbeta // MatrixForm, printprec]];
   Print["\!\(\*SubscriptBox[OverscriptBox[\"B\", \"~\"], \"i\"]\)=",
    NumberForm[bzohapproxbetaval // MatrixForm, printprec]];
   Print["\!\(\*SubscriptBox[\"x\", \"0\"]\)=",
    NumberForm[bzohapproxx0 // MatrixForm, printprec]];
   myprint[StringJoin[name, " results:"]];
   Print[NumberForm[bzohapproxsequence // MatrixForm, printprec]];]];
```

```
(* TFOH *)
If[whichmodels[[5]] == 1, name = "TFOH";
  res = TFOH[pe, pa, pb, x0minus, T];
  tfohalfa = res[[1]]; tfohbeta = res[[2]];
  tfohbetatval = Transpose[res[[3]]]; tfohx0 = res[[4]];
  ord = Dimensions[tfohbetatval][[2]]; xlast = Flatten[tfohx0];
  tfohsequence = {xlast};
  For[pl = 1, pl <= PlithosVimaton, pl++, xnew = tfohalfa.xlast;
   For[tmp = 1, tmp <= ord, tmp++,
    xnew = xnew +
      Transpose[tfohbetatval][[tmp]].{input[(pl - 1 + (tmp - 1))*T]}];
   xlast = xnew; tfohsequence = Append[tfohsequence, xlast];
   ]; If[matrices, myprint[StringJoin[name, " matrices:"]];
   Print["\!\(\*OverscriptBox[\"A\", \"~\"]\)=",
    NumberForm[tfohalfa // MatrixForm, printprec]];
   Print["\!\(\*OverscriptBox[\"B\", \"~\"]\)=",
    NumberForm[tfohbeta // MatrixForm, printprec]];
   Print["\!\(\*SubscriptBox[OverscriptBox[\"B\", \"~\"], \"i\"]\)=",
    NumberForm[tfohbetatval // MatrixForm, printprec]];
   Print["\!\(\*SubscriptBox[\"x\", \"0\"]\)=",
    NumberForm[tfohx0 // MatrixForm, printprec]];
   myprint[StringJoin[name, " results:"]];
   Print[NumberForm[tfohsequence // MatrixForm, printprec]];]];


(* TFOH-approx *)
If[whichmodels[[6]] == 1, name = "TFOH-approx";
  res = TFOHapprox[pe, pa, pb, x0minus, T];
  tfohapproxalfa = res[[1]]; tfohapproxbeta = res[[2]];
  tfohapproxbetatval = Transpose[res[[3]]]; tfohapproxx0 = res[[4]];
  ord = Dimensions[tfohapproxbetatval][[2]];
  xlast = Flatten[tfohapproxx0]; tfohapproxsequence = {xlast};
  For[pl = 1, pl <= PlithosVimaton, pl++, xnew = tfohapproxalfa.xlast;
   For[tmp = 1, tmp <= ord, tmp++,
    xnew = xnew +
      Transpose[tfohapproxbetatval][[tmp]].Der[u, pl - 1 + (tmp - 1),
        T, 0]];
   xlast = xnew;
   tfohapproxsequence = Append[tfohapproxsequence, xlast];
   ]; If[matrices, myprint[name, " matrices:"];
   Print["\!\(\*OverscriptBox[\"A\", \"~\"]\)=",
    NumberForm[tfohapproxalfa // MatrixForm, printprec]];
   Print["\!\(\*OverscriptBox[\"B\", \"~\"]\)=",
    NumberForm[tfohapproxbeta // MatrixForm, printprec]];
   Print["\!\(\*SubscriptBox[OverscriptBox[\"B\", \"~\"], \"i\"]\)=",
    NumberForm[tfohapproxbetatval // MatrixForm, printprec]];
   Print["\!\(\*SubscriptBox[\"x\", \"0\"]\)=",
    NumberForm[tfohapproxx0 // MatrixForm, printprec]];
   myprint[StringJoin[name, " results:"]];
   Print[NumberForm[tfohapproxsequence // MatrixForm, printprec]];]];
```

```
(* BFOH *)
If[whichmodels[[7]] == 1, name = "BFOH";
  res = BFOH[pe, pa, pb, x0minus, T];
  bfohalfa = res[[1]]; bfohbeta = res[[2]];
  bfohbetaval = Transpose[res[[3]]]; bfohx0 = res[[4]];
  ord = Dimensions[bfohbetaval][[2]]; xlast = Flatten[bfohx0];
  bfohsequence = {xlast};
  For[pl = 1, pl <= PlithosVimaton, pl++, xnew = bfohalfa.xlast;
   For[tmp = 1, tmp <= ord, tmp++,
    xnew = xnew +
      Transpose[bfohbetaval][[tmp]].{input[(pl - 1 - (tmp - 2))*T]}];
   xlast = xnew; bfohsequence = Append[bfohsequence, xlast];
   ]; If[matrices, myprint[StringJoin[name, " matrices:"]];
   Print["\!\(\*OverscriptBox[\"A\", \"~\"]\)=",
    NumberForm[bfohalfa // MatrixForm, printprec]];
   Print["\!\(\*OverscriptBox[\"B\", \"~\"]\)=",
    NumberForm[bfohbeta // MatrixForm, printprec]];
   Print["\!\(\*SubscriptBox[OverscriptBox[\"B\", \"~\"], \"i\"]\)=",
    NumberForm[bfohbetaval // MatrixForm, printprec]];
   Print["\!\(\*SubscriptBox[\"x\", \"0\"]\)=",
    NumberForm[bfohx0 // MatrixForm, printprec]];
   myprint[StringJoin[name, " results:"]];
   Print[NumberForm[bfohsequence // MatrixForm, printprec]];]];


(* BFOH-approx *)
If[whichmodels[[8]] == 1, name = "BFOH-approx";
  res = BFOHapprox[pe, pa, pb, x0minus, T];
  bfohapproxalfa = res[[1]]; bfohapproxbeta = res[[2]];
  bfohapproxbetaval = Transpose[res[[3]]]; bfohapproxx0 = res[[4]];
  ord = Dimensions[bfohapproxbetaval][[2]];
  xlast = Flatten[bfohapproxx0]; bfohapproxsequence = {xlast};
  For[pl = 1, pl <= PlithosVimaton, pl++, xnew = bfohapproxalfa.xlast;
   For[tmp = 1, tmp <= ord, tmp++,
    xnew = xnew +
      Transpose[bfohapproxbetaval][[tmp]].Der[u, pl - 1 - (tmp - 2),
        T, 0]];
   xlast = xnew;
   bfohapproxsequence = Append[bfohapproxsequence, xlast];
   ]; If[matrices, myprint[StringJoin[name, " matrices:"]];
   Print["\!\(\*OverscriptBox[\"A\", \"~\"]\)=",
    NumberForm[bfohapproxalfa // MatrixForm, printprec]];
   Print["\!\(\*OverscriptBox[\"B\", \"~\"]\)=",
    NumberForm[bfohapproxbeta // MatrixForm, printprec]];
   Print["\!\(\*SubscriptBox[OverscriptBox[\"B\", \"~\"], \"i\"]\)=",
    NumberForm[bfohapproxbetaval // MatrixForm, printprec]];
   Print["\!\(\*SubscriptBox[\"x\", \"0\"]\)=",
    NumberForm[bfohapproxx0 // MatrixForm, printprec]];
   myprint[StringJoin[name, " results:"]];
   Print[NumberForm[bfohapproxsequence // MatrixForm, printprec]];]];
```

```
(*CONtinuous*)
consol = Expand[InverseLaplaceTransform[N[Simplify[Inverse[s*pe - pa].(pe.x0minus +
    pb*LaplaceTransform[input[t], t, s])]], s, t]];
consol = consol /.DiracDelta[t] -> 0; myprint["Continuous Solution:"];
Print[Expand[consol]];


(* Legends for Plots *)
totalplotlegends = {};
If[whichmodels[[1]] == 1, AppendTo[totalplotlegends, "ZOH"]];
If[whichmodels[[2]] == 1, AppendTo[totalplotlegends, "ZOH-approx"]];
If[whichmodels[[3]] == 1, AppendTo[totalplotlegends, "BZOH"]];
If[whichmodels[[4]] == 1, AppendTo[totalplotlegends, "BZOH-approx"]];
If[whichmodels[[5]] == 1, AppendTo[totalplotlegends, "TFOH"]];
If[whichmodels[[6]] == 1, AppendTo[totalplotlegends, "TFOH-approx"]];
If[whichmodels[[7]] == 1, AppendTo[totalplotlegends, "BFOH"]];
If[whichmodels[[8]] == 1, AppendTo[totalplotlegends, "BFOH-approx"]];


(* ZOH-ZOHapprox-BZOH-BZOHapprox-TFOH-TFOHapprox-BFOH-BFOHapprox plots *)
myprint["All methods plots:"];
For[plotx = 1, plotx <= 3, plotx++,
  totallistplot = {};
  If[whichmodels[[1]] == 1,
   plotzoh =Transpose[Prepend[{Transpose[zohsequence][[plotx]]},
   Table[k*T, {k, 0, PlithosVimaton}]]]; AppendTo[totallistplot, plotzoh]];
  If[whichmodels[[2]] == 1,
   plotzohapprox =Transpose[Prepend[{Transpose[zohapproxsequence][[plotx]]},
   Table[k*T, {k, 0, PlithosVimaton}]]]; AppendTo[totallistplot, plotzohapprox]];
  If[whichmodels[[3]] == 1,
   plotbzoh =Transpose[Prepend[{Transpose[bzohsequence][[plotx]]},
   Table[k*T, {k, 0, PlithosVimaton}]]]; AppendTo[totallistplot, plotbzoh]];
  If[whichmodels[[4]] == 1,
   plotbzohapprox =Transpose[Prepend[{Transpose[bzohapproxsequence][[plotx]]},
   Table[k*T, {k, 0, PlithosVimaton}]]]; AppendTo[totallistplot, plotbzohapprox]];
  If[whichmodels[[5]] == 1,
   plottfoh =Transpose[Prepend[{Transpose[tfohsequence][[plotx]]},
   Table[k*T, {k, 0, PlithosVimaton}]]]; AppendTo[totallistplot, plottfoh]];
  If[whichmodels[[6]] == 1,
   plottfohapprox =Transpose[Prepend[{Transpose[tfohapproxsequence][[plotx]]},
   Table[k*T, {k, 0, PlithosVimaton}]]]; AppendTo[totallistplot, plottfohapprox]];
  If[whichmodels[[7]] == 1,
   plotbfoh =Transpose[Prepend[{Transpose[bfohsequence][[plotx]]},
   Table[k*T, {k, 0, PlithosVimaton}]]]; AppendTo[totallistplot, plotbfoh]];
  If[whichmodels[[8]] == 1,
   plotbfohapprox =Transpose[Prepend[{Transpose[bfohapproxsequence][[plotx]]},
   Table[k*T, {k, 0, PlithosVimaton}]]]; AppendTo[totallistplot, plotbfohapprox]];
  Print[ListPlot[totallistplot, AspectRatio -> 1,
    AxesLabel -> {t, Subscript[x, plotx]}, AxesOrigin -> {0, 0},
    PlotMarkers -> Automatic, PlotLegend -> totalplotlegends,
    LegendPosition -> {1.1, -0.4}, LegendTextSpace -> 10, Joined -> True]];];
```

```
(*Abs.Differences from Continuous model *)
 myprint["Abs.Differences from Continuous model:"];
 For[plotx = 1, plotx <= 3, plotx++,
   totallistplot = {}; Subscript[consolx, plotx] =
    Flatten[consol[[plotx]] /.t -> Table[k*T, {k, 0, PlithosVimaton}]];
   If[whichmodels[[1]] == 1,
    absdiffconzoh = Abs[Transpose[zohsequence][[plotx]] -
    Subscript[consolx, plotx]]; AppendTo[totallistplot, absdiffconzoh]];
   If[whichmodels[[2]] == 1,
    absdiffconzohapprox = Abs[Transpose[zohapproxsequence][[plotx]] -
    Subscript[consolx, plotx]]; AppendTo[totallistplot, absdiffconzohapprox]];
   If[whichmodels[[3]] == 1,
    absdiffconbzoh = Abs[Transpose[bzohsequence][[plotx]] -
    Subscript[consolx, plotx]]; AppendTo[totallistplot, absdiffconbzoh]];
   If[whichmodels[[4]] == 1,
    absdiffconbzohapprox = Abs[Transpose[bzohapproxsequence][[plotx]] -
    Subscript[consolx, plotx]]; AppendTo[totallistplot, absdiffconbzohapprox]];
   If[whichmodels[[5]] == 1,
    absdiffcontfoh = Abs[Transpose[tfohsequence][[plotx]] -
    Subscript[consolx, plotx]]; AppendTo[totallistplot, absdiffcontfoh]];
   If[whichmodels[[6]] == 1,
    absdiffcontfohapprox = Abs[Transpose[tfohapproxsequence][[plotx]] -
    Subscript[consolx, plotx]]; AppendTo[totallistplot, absdiffcontfohapprox]];
   If[whichmodels[[7]] == 1,
    absdiffconbfoh = Abs[Transpose[bfohsequence][[plotx]] -
    Subscript[consolx, plotx]]; AppendTo[totallistplot, absdiffconbfoh]];
   If[whichmodels[[8]] == 1,
    absdiffconbfohapprox = Abs[Transpose[bfohapproxsequence][[plotx]] -
    Subscript[consolx, plotx]]; AppendTo[totallistplot, absdiffconbfohapprox]];
   Print[ListPlot[totallistplot,
     AxesLabel -> {t, Subscript[x, plotx]}, AxesOrigin -> {0, 0},
     PlotMarkers -> Automatic, PlotLegend -> totalplotlegends,
     LegendPosition -> {1.1, -0.4}, LegendTextSpace -> 10, Joined -> True]];];


 (* Logarithmic Abs.Differences *)
 myprint["Logarithmic Abs.Differences:"];
 For[plotx = 1, plotx <= 3, plotx++,
   totallistplot = {}; Subscript[consolx, plotx] =
    Flatten[consol[[plotx]] /.t -> Table[k*T, {k, 0, PlithosVimaton}]];
   If[whichmodels[[1]] == 1,
    logabsdiffconzoh = Log[10, Abs[ Transpose[zohsequence][[plotx]] -
    Subscript[consolx, plotx]]]; AppendTo[totallistplot, logabsdiffconzoh]];
   If[whichmodels[[2]] == 1,
    logabsdiffconzohapprox = Log[10, Abs[ Transpose[zohapproxsequence][[plotx]] -
    Subscript[consolx, plotx]]]; AppendTo[totallistplot, logabsdiffconzohapprox]];
   If[whichmodels[[3]] == 1,
    logabsdiffconbzoh = Log[10, Abs[ Transpose[bzohsequence][[plotx]] -
    Subscript[consolx, plotx]]]; AppendTo[totallistplot, logabsdiffconbzoh]];
   If[whichmodels[[4]] == 1,
```

```
    logabsdiffconbzohapprox = Log[10, Abs[ Transpose[bzohapproxsequence][[plotx]] -
    Subscript[consolx, plotx]]]; AppendTo[totallistplot, logabsdiffconbzohapprox]];
   If[whichmodels[[5]] == 1,
    logabsdiffcontfoh = Log[10, Abs[ Transpose[tfohsequence][[plotx]] -
    Subscript[consolx, plotx]]]; AppendTo[totallistplot, logabsdiffcontfoh]];
   If[whichmodels[[6]] == 1,
    logabsdiffcontfohapprox = Log[10, Abs[ Transpose[tfohapproxsequence][[plotx]] -
    Subscript[consolx, plotx]]]; AppendTo[totallistplot, logabsdiffcontfohapprox]];
   If[whichmodels[[7]] == 1,
    logabsdiffconbfoh = Log[10, Abs[ Transpose[bfohsequence][[plotx]] -
    Subscript[consolx, plotx]]]; AppendTo[totallistplot, logabsdiffconbfoh]];
   If[whichmodels[[8]] == 1,
    logabsdiffconbfohapprox = Log[10, Abs[ Transpose[bfohapproxsequence][[plotx]] -
    Subscript[consolx, plotx]]]; AppendTo[totallistplot, logabsdiffconbfohapprox]];
   Print[ListPlot[totallistplot,
     AxesLabel -> {t, Subscript[logx, plotx]}, AxesOrigin -> {0, 0},
     PlotMarkers -> Automatic, PlotLegend -> totalplotlegends,
     LegendPosition -> {1.1, -0.4}, LegendTextSpace -> 10, Joined -> True]];];

)
```

Suppose now we want to run our code in the following example. We simply set the parameters involved and we call our function.

```
pe = {{-1, 12, 37}, {2, 6, 13}, {-1, 2, 8}};
pa = {{-38, -54, -47}, {3, -11, -32}, {-3, -9, -13}};
pb = {{0}, {0}, {1}};
x0minus = {{1}, {0}, {0}};
T = 0.1;
input[t_] := t;
steps = 10;
whichmodels = {1, 1, 1, 1, 1, 1, 1, 1};
matrices = True;
myanalysis[pe, pa, pb, x0minus, T, steps, whichmodels, matrices]
```

and we get

ZOH matrices:

$$\tilde{A}=\begin{pmatrix} 0.924703543586239 & -0.1003952752183485 & -0.1254940940229356 \\ 0.02509881880458713 & 1.03346509173945 & 0.04183136467431187 \\ -0.0836627293486237 & -0.111550305798165 & 0.860562117752294 \end{pmatrix}$$

$$\tilde{B}=\begin{pmatrix} 2.259870966558588 - 3.555769230769231\, s + 1.269230769230769\, s^2 \\ -2.919956988852862 + 4.601923076923077\, s - 1.673076923076923\, s^2 \\ 0.941523296176209 - 1.548076923076923\, s + 0.576923076923077\, s^2 \end{pmatrix}$$

$$\tilde{B}_i=\begin{pmatrix} (2.259870966558588) & (-3.555769230769231) & (1.269230769230769) \\ (-2.919956988852862) & (4.601923076923077) & (-1.673076923076923) \\ (0.941523296176209) & (-1.548076923076923) & (0.576923076923077) \end{pmatrix}$$

$$x_0=\begin{pmatrix} \frac{141}{260} \\ -\frac{159}{520} \\ \frac{27}{52} \end{pmatrix}$$

ZOH results:

$$\begin{pmatrix} \frac{141}{260} & -\frac{159}{520} & \frac{27}{52} \\ 0.3652804666631617 & -0.1550934888877206 & 0.3961449629590685 \\ 0.1992354234706629 & -0.00807847449022092 & 0.2852615816340698 \\ 0.04218183078755566 & 0.1359393897374818 & 0.1843687008750618 \\ -0.1075101848487338 & 0.2775033949495784 & 0.0916553501680735 \\ -0.2511750509656404 & 0.4170583503218808 & 0.005638832260399241 \\ -0.3899053044176296 & 0.5549684348058779 & -0.07489478268625549 \\ -0.5245954387263225 & 0.691531812908776 & -0.1509393763625811 \\ -0.6559778032462997 & 0.826992601082102 & -0.223308670273667 \\ -0.7846519949161532 & 0.961550664972054 & -0.2926688832401708 \\ -0.911108922188959 & 1.095369640729656 & -0.3595654690988445 \end{pmatrix}$$

ZOH−approx matrices:

$$\tilde{A}=\begin{pmatrix} 0.924703543586239 & -0.1003952752183485 & -0.1254940940229356 \\ 0.02509881880458713 & 1.03346509173945 & 0.04183136467431187 \\ -0.0836627293486237 & -0.111550305798165 & 0.860562117752294 \end{pmatrix}$$

$$\tilde{B}=\begin{pmatrix} 2.259870966558588 - 3.555769230769231\, s + 1.269230769230769\, s^2 \\ -2.919956988852862 + 4.601923076923077\, s - 1.673076923076923\, s^2 \\ 0.941523296176209 - 1.548076923076923\, s + 0.576923076923077\, s^2 \end{pmatrix}$$

$$\tilde{B}_i=\begin{pmatrix} (2.259870966558588) & (-3.555769230769231) & (1.269230769230769) \\ (-2.919956988852862) & (4.601923076923077) & (-1.673076923076923) \\ (0.941523296176209) & (-1.548076923076923) & (0.576923076923077) \end{pmatrix}$$

$$x_0=\begin{pmatrix} 0.5423076923076923 \\ -0.3057692307692308 \\ 0.5192307692307693 \end{pmatrix}$$

ZOH−approx results:

$$\begin{pmatrix} 0.5423076923076923 & -0.3057692307692308 & 0.5192307692307693 \\ 0.3652804666631617 & -0.1550934888877206 & 0.3961449629590685 \\ 0.1992354234706629 & -0.00807847449022092 & 0.2852615816340698 \\ 0.04218183078755566 & 0.1359393897374818 & 0.1843687008750618 \\ -0.1075101848487338 & 0.2775033949495784 & 0.0916553501680735 \\ -0.2511750509656404 & 0.4170583503218808 & 0.005638832260399241 \\ -0.3899053044176296 & 0.5549684348058779 & -0.07489478268625549 \\ -0.5245954387263225 & 0.691531812908776 & -0.1509393763625811 \\ -0.6559778032462997 & 0.826992601082102 & -0.223308670273667 \\ -0.7846519949161532 & 0.961550664972054 & -0.2926688832401708 \\ -0.911108922188959 & 1.095369640729656 & -0.3595654690988445 \end{pmatrix}$$

BZOH matrices:

$$\tilde{A}=\begin{pmatrix} 0.924703543586239 & -0.1003952752183485 & -0.1254940940229356 \\ 0.02509881880458713 & 1.03346509173945 & 0.04183136467431187 \\ -0.0836627293486237 & -0.111550305798165 & 0.860562117752294 \end{pmatrix}$$

$$\tilde{B}=\begin{pmatrix} -1.54782134113372+\frac{0.2519230769230769}{s}+1.269230769230769\,s \\ 2.099273780377906-\frac{0.4173076923076922}{s}-1.673076923076923\,s \\ -0.7892459345930224+\frac{0.1826923076923078}{s}+0.576923076923077\,s \end{pmatrix}$$

$$\tilde{B}_i=\begin{pmatrix} (0.2519230769230769) & (-1.54782134113372) & (1.269230769230769) \\ (-0.4173076923076922) & (2.099273780377906) & (-1.673076923076923) \\ (0.1826923076923078) & (-0.7892459345930224) & (0.576923076923077) \end{pmatrix}$$

$$x_0=\begin{pmatrix} 0.5423076923076922 \\ -0.3057692307692308 \\ 0.5192307692307693 \end{pmatrix}$$

BZOH results:

$$\begin{pmatrix} 0.5423076923076922 & -0.3057692307692308 & 0.5192307692307693 \\ 0.3652804666631617 & -0.1550934888877207 & 0.3961449629590685 \\ 0.1992354234706629 & -0.00807847449022109 & 0.2852615816340696 \\ 0.04218183078755572 & 0.1359393897374813 & 0.1843687008750617 \\ -0.1075101848487338 & 0.2775033949495779 & 0.0916553501680733 \\ -0.2511750509656407 & 0.4170583503218803 & 0.005638832260398963 \\ -0.3899053044176297 & 0.5549684348058769 & -0.07489478268625563 \\ -0.5245954387263227 & 0.6915318129087749 & -0.1509393763625812 \\ -0.6559778032462998 & 0.826992601082101 & -0.2233086702736671 \\ -0.7846519949161533 & 0.961550664972052 & -0.292668883240171 \\ -0.911108922188959 & 1.095369640729655 & -0.3595654690988446 \end{pmatrix}$$

BZOH−approx matrices:

$$\tilde{A}=\begin{pmatrix} 0.924703543586239 & -0.1003952752183485 & -0.1254940940229356 \\ 0.02509881880458713 & 1.03346509173945 & 0.04183136467431187 \\ -0.0836627293486237 & -0.111550305798165 & 0.860562117752294 \end{pmatrix}$$

$$\tilde{B}=\begin{pmatrix} -1.54782134113372+\frac{0.2519230769230769}{s}+1.269230769230769\,s \\ 2.099273780377906-\frac{0.4173076923076922}{s}-1.673076923076923\,s \\ -0.7892459345930224+\frac{0.1826923076923078}{s}+0.576923076923077\,s \end{pmatrix}$$

$$\tilde{B}_i=\begin{pmatrix} (0.2519230769230769) & (-1.54782134113372) & (1.269230769230769) \\ (-0.4173076923076922) & (2.099273780377906) & (-1.673076923076923) \\ (0.1826923076923078) & (-0.7892459345930224) & (0.576923076923077) \end{pmatrix}$$

$$x_0=\begin{pmatrix} 0.5423076923076923 \\ -0.3057692307692308 \\ 0.5192307692307693 \end{pmatrix}$$

BZOH−approx results:

$$\begin{pmatrix} 0.5423076923076923 & -0.3057692307692308 & 0.5192307692307693 \\ 0.3652804666631617 & -0.1550934888877207 & 0.3961449629590685 \\ 0.1992354234706629 & -0.00807847449022109 & 0.2852615816340696 \\ 0.04218183078755572 & 0.1359393897374813 & 0.1843687008750617 \\ -0.1075101848487338 & 0.2775033949495779 & 0.0916553501680733 \\ -0.2511750509656407 & 0.4170583503218803 & 0.005638832260398963 \\ -0.3899053044176297 & 0.5549684348058769 & -0.07489478268625563 \\ -0.5245954387263227 & 0.6915318129087749 & -0.1509393763625812 \\ -0.6559778032462998 & 0.826992601082101 & -0.2233086702736671 \\ -0.7846519949161533 & 0.961550664972052 & -0.292668883240171 \\ -0.911108922188959 & 1.095369640729655 & -0.3595654690988446 \end{pmatrix}$$

TFOH matrices:

$$\tilde{A}=\begin{pmatrix} 0.924703543586239 & -0.1003952752183485 & -0.1254940940229356 \\ 0.02509881880458713 & 1.03346509173945 & 0.04183136467431187 \\ -0.0836627293486237 & -0.111550305798165 & 0.860562117752294 \end{pmatrix}$$

$$\tilde{B}=\begin{pmatrix} 2.273648876274603-3.569547140485246\,s+1.269230769230769\,s^2 \\ -2.924549625424867+4.606515713495082\,s-1.673076923076923\,s^2 \\ 0.956832084749559-1.563385711650274\,s+0.576923076923077\,s^2 \end{pmatrix}$$

$$\tilde{B}_i=\begin{pmatrix} (2.273648876274603) & (-3.569547140485246) & (1.269230769230769) \\ (-2.924549625424867) & (4.606515713495082) & (-1.673076923076923) \\ (0.956832084749559) & (-1.563385711650274) & (0.576923076923077) \end{pmatrix}$$

$$x_0=\begin{pmatrix} \frac{141}{260} \\ -\frac{159}{520} \\ \frac{27}{52} \end{pmatrix}$$

TFOH results:

$$\begin{pmatrix} \frac{141}{260} & -\frac{159}{520} & \frac{27}{52} \\ 0.3639026756915602 & -0.1546342252305201 & 0.3946140841017335 \\ 0.1967295926592978 & -0.007243197553099156 & 0.2824773251769976 \\ 0.03875243906867898 & 0.1370825203104407 & 0.1805582656318658 \\ -0.1116957242849308 & 0.2788985747616444 & 0.0870047507945217 \\ -0.2559796717918775 & 0.4186598905972936 & 0.0003003646756921663 \\ -0.3952167862165511 & 0.556738928738852 & -0.0807964291295002 \\ -0.5303219031911155 & 0.6934406343970407 & -0.1573021146567937 \\ -0.6620440267816363 & 0.829014675593882 & -0.2300489186462608 \\ -0.7909963896511809 & 0.963665463217064 & -0.2997182107235321 \\ -0.917681064239795 & 1.097560354746603 & -0.3668678491553251 \end{pmatrix}$$

TFOH–approx matrices:

$$\tilde{A} = \begin{pmatrix} 0.924703543586239 & -0.1003952752183485 & -0.1254940940229356 \\ 0.02509881880458713 & 1.03346509173945 & 0.04183136467431187 \\ -0.0836627293486237 & -0.111550305798165 & 0.860562117752294 \end{pmatrix}$$

$$\tilde{B} = \begin{pmatrix} 2.273648876274603 - 3.569547140485246\, s + 1.269230769230769\, s^2 \\ -2.924549625424867 + 4.606515713495082\, s - 1.673076923076923\, s^2 \\ 0.956832084749559 - 1.563385711650274\, s + 0.576923076923077\, s^2 \end{pmatrix}$$

$$\tilde{B}_i = \begin{pmatrix} (2.273648876274603) & (-3.569547140485246) & (1.269230769230769) \\ (-2.924549625424867) & (4.606515713495082) & (-1.673076923076923) \\ (0.956832084749559) & (-1.563385711650274) & (0.576923076923077) \end{pmatrix}$$

$$x_0 = \begin{pmatrix} 0.5423076923076923 \\ -0.3057692307692308 \\ 0.5192307692307693 \end{pmatrix}$$

TFOH–approx results:

$$\begin{pmatrix} 0.5423076923076923 & -0.3057692307692308 & 0.5192307692307693 \\ 0.3639026756915602 & -0.1546342252305201 & 0.3946140841017335 \\ 0.1967295926592978 & -0.007243197553099156 & 0.2824773251769976 \\ 0.03875243906867898 & 0.1370825203104407 & 0.1805582656318658 \\ -0.1116957242849308 & 0.2788985747616444 & 0.0870047507945217 \\ -0.2559796717918775 & 0.4186598905972936 & 0.0003003646756921663 \\ -0.3952167862165511 & 0.556738928738852 & -0.0807964291295002 \\ -0.5303219031911155 & 0.6934406343970407 & -0.1573021146567937 \\ -0.6620440267816363 & 0.829014675593882 & -0.2300489186462608 \\ -0.7909963896511809 & 0.963665463217064 & -0.2997182107235321 \\ -0.917681064239795 & 1.097560354746603 & -0.3668678491553251 \end{pmatrix}$$

BFOH matrices:

$$\tilde{A}=\begin{pmatrix} 0.924703543586239 & -0.1003952752183485 & -0.1254940940229356 \\ 0.02509881880458713 & 1.03346509173945 & 0.04183136467431187 \\ -0.0836627293486237 & -0.111550305798165 & 0.860562117752294 \end{pmatrix}$$

$$\tilde{B}=\begin{pmatrix} -1.561599250849735+\frac{0.2519230769230769}{s}+1.283008678946784\,s \\ 2.103866416949911-\frac{0.4173076923076922}{s}-1.677669559648928\,s \\ -0.804554723166373+\frac{0.1826923076923078}{s}+0.5922318654964275\,s \end{pmatrix}$$

$$\tilde{B}_i=\begin{pmatrix} (0.2519230769230769) & (-1.561599250849735) & (1.283008678946784) \\ (-0.4173076923076922) & (2.103866416949911) & (-1.677669559648928) \\ (0.1826923076923078) & (-0.804554723166373) & (0.5922318654964275) \end{pmatrix}$$

$$x_0=\begin{pmatrix} \frac{141}{260} \\ -\frac{159}{520} \\ \frac{27}{52} \end{pmatrix}$$

BFOH results:

$$\begin{pmatrix} \frac{141}{260} & -\frac{159}{520} & \frac{27}{52} \\ 0.3639026756915602 & -0.1546342252305202 & 0.3946140841017334 \\ 0.1967295926592979 & -0.00724319755309949 & 0.2824773251769975 \\ 0.03875243906867929 & 0.1370825203104401 & 0.1805582656318657 \\ -0.1116957242849305 & 0.2788985747616434 & 0.0870047507945216 \\ -0.2559796717918772 & 0.4186598905972923 & 0.0003003646756919442 \\ -0.3952167862165502 & 0.5567389287388501 & -0.0807964291295002 \\ -0.5303219031911145 & 0.6934406343970386 & -0.1573021146567938 \\ -0.6620440267816347 & 0.829014675593879 & -0.2300489186462609 \\ -0.7909963896511791 & 0.96366546321706 & -0.2997182107235323 \\ -0.917681064239793 & 1.097560354746598 & -0.3668678491553253 \end{pmatrix}$$

BFOH–approx matrices:

$$\tilde{A}=\begin{pmatrix} 0.924703543586239 & -0.1003952752183485 & -0.1254940940229356 \\ 0.02509881880458713 & 1.03346509173945 & 0.04183136467431187 \\ -0.0836627293486237 & -0.111550305798165 & 0.860562117752294 \end{pmatrix}$$

$$\tilde{B}=\begin{pmatrix} -1.561599250849735+\frac{0.2519230769230769}{s}+1.283008678946784\,s \\ 2.103866416949911-\frac{0.4173076923076922}{s}-1.677669559648928\,s \\ -0.804554723166373+\frac{0.1826923076923078}{s}+0.5922318654964275\,s \end{pmatrix}$$

$$\tilde{B}_i=\begin{pmatrix} (0.2519230769230769) & (-1.561599250849735) & (1.283008678946784) \\ (-0.4173076923076922) & (2.103866416949911) & (-1.677669559648928) \\ (0.1826923076923078) & (-0.804554723166373) & (0.5922318654964275) \end{pmatrix}$$

$$x_0=\begin{pmatrix} 0.5423076923076923 \\ -0.3057692307692308 \\ 0.5192307692307693 \end{pmatrix}$$

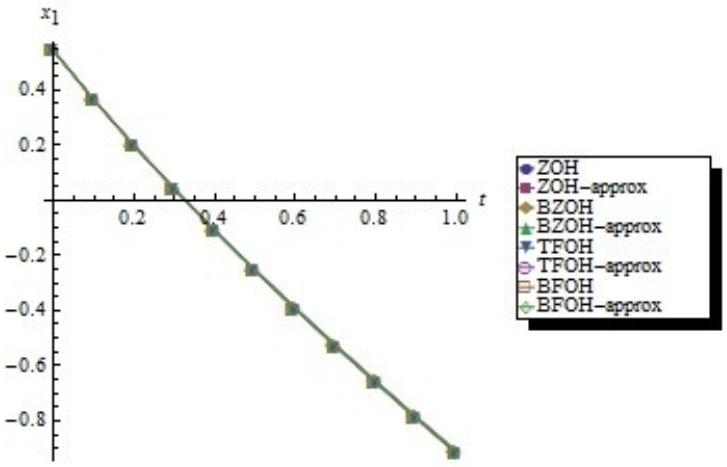
BFOH−approx results:

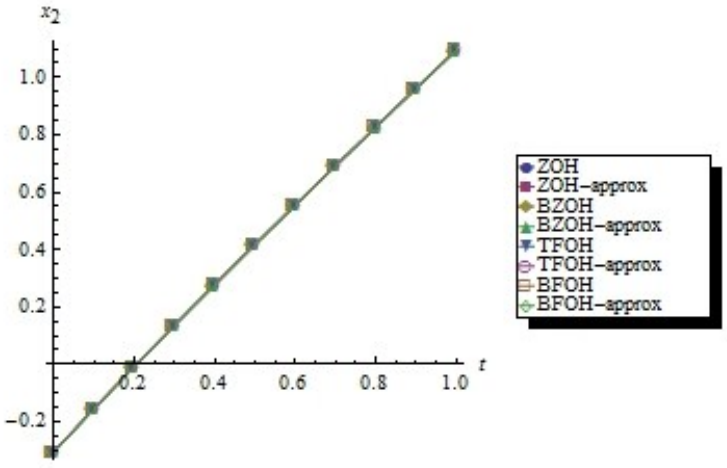
0.5423076923076923	-0.3057692307692308	0.5192307692307693
0.3639026756915602	-0.1546342252305202	0.3946140841017334
0.1967295926592979	-0.00724319755309949	0.2824773251769975
0.03875243906867929	0.1370825203104401	0.1805582656318657
-0.1116957242849305	0.2788985747616434	0.0870047507945216
-0.2559796717918772	0.4186598905972923	0.0003003646756919442
-0.3952167862165502	0.5567389287388501	-0.0807964291295002
-0.5303219031911145	0.6934406343970386	-0.1573021146567938
-0.6620440267816347	0.829014675593879	-0.2300489186462609
-0.7909963896511791	0.96366546321706	-0.2997182107235323
-0.917681064239793	1.097560354746598	-0.3668678491553253

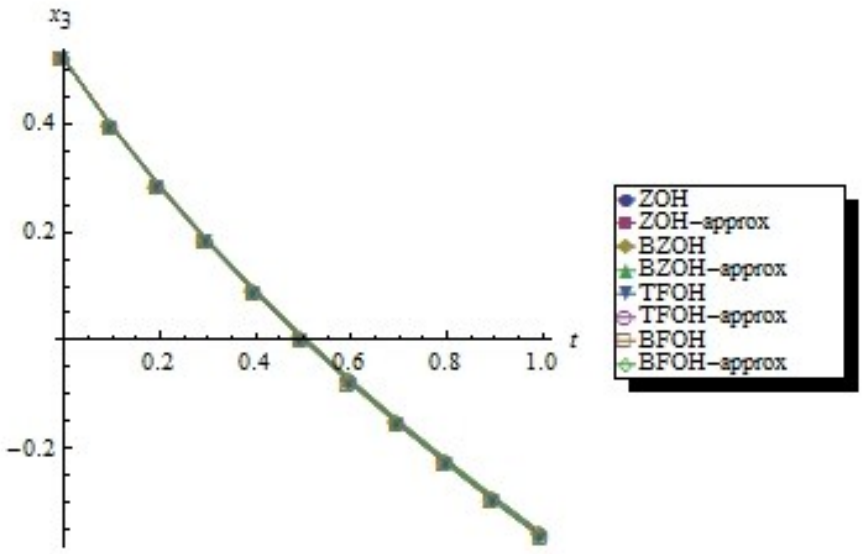
Continuous Solution:

$$\{\{0.200481 + 0.341827\, e^{-2.\, t} - 1.16442\, t\},$$
$$\{-0.191827 - 0.113942\, e^{-2.\, t} + 1.30481\, t\},$$
$$\{0.139423 + 0.379808\, e^{-2.\, t} - 0.557692\, t\}\}$$

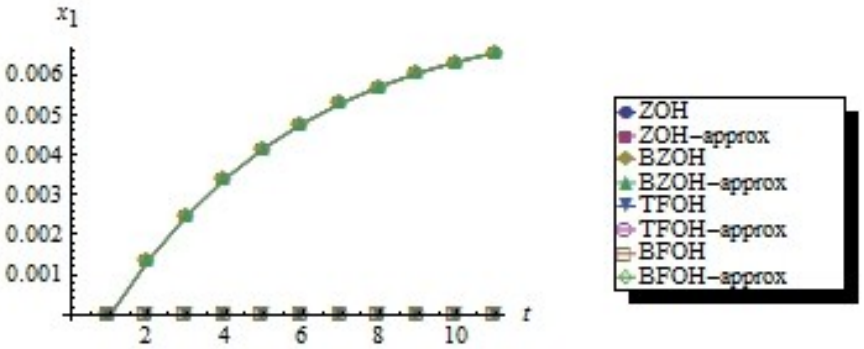
All methods plots:

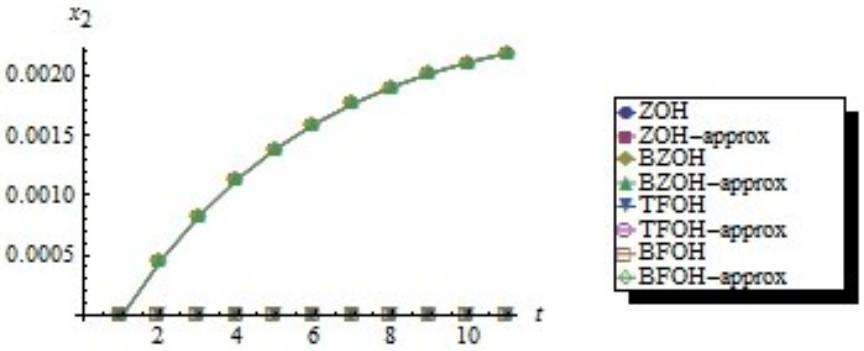


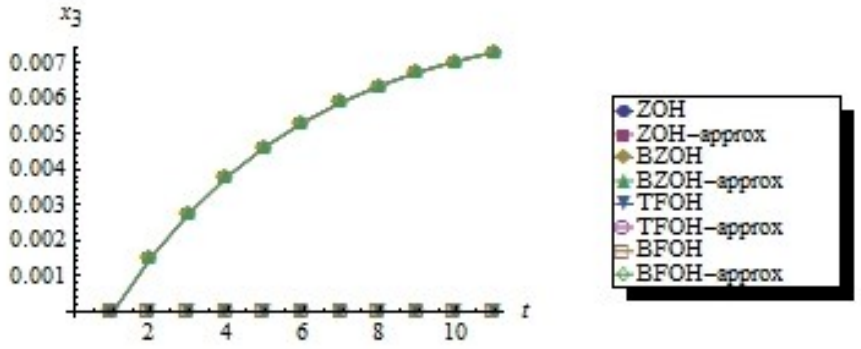




Abs.Differences from Continuous model:







Logarithmic Abs.Differences: