LISBON SCHOOL OF ECONOMICS & MANAGEMENT
UNIVERSIDADE DE LISBOA

# MASTER

MATHEMATICAL FINANCE

# MASTER´S FINAL WORK

DISSERTATION

PERIODIC PATTERNS IN POLLING SYSTEMS

ALINA RAQUEL BASTOS TELES

OCTOBER - 2018

LISBON SCHOOL OF ECONOMICS & MANAGEMENT
UNIVERSIDADE DE LISBOA

# MASTER

MATHEMATICAL FINANCE

# MASTER´S FINAL WORK

DISSERTATION

PERIODIC PATTERNS IN POLLING SYSTEMS

ALINA RAQUEL BASTOS TELES

**SUPERVISOR:**
JOSÉ PEDRO GAIVÃO

OCTOBER - 2018

# Resumo


Uma estrutura matemática conveniente para modelar as flutuações estocásticas nos preços de mercado é baseada na teoria das redes de filas de espera. Os *polling systems* são uma classe especial de modelos de filas de espera. Uma definição clássica de um *polling system* consiste num sistema com múltiplas filas e um único servidor que muda de fila de acordo com uma determinada política de serviço.

Sabe-se que um *polling system* é recorrente positivo (admitindo uma única distribuição estacionária) se, e só se, a carga total do sistema for menor do que um. No regime transiente, i.e., carga total maior do que um, foi provado que um *polling system* com três filas de espera é assimptoticamente periódico para praticamente todas as opções de políticas de serviço. Quando os *polling systems* têm mais do que três filas de espera, a caracterização do seu comportamento num regime transiente é um assunto em aberto.

Tendo como hipótese que o processo é transiente, o objetivo deste trabalho é estudar os padrões periódicos resultantes das trocas entre filas efetuadas pelo servidor. Em particular, esta dissertação tem como finalidade desenvolver técnicas que permitam responder a algumas questões levantadas num recente artigo publicado por I. MacPhee e os seus coautores em 2006.



**Palavras-chave:** Redes de Filas, Polling Systems, Transitoriedade, Sistema Dinâmico, Periodicidade Assimptótica, q.c. Convergência.

# Abstract

A convenient mathematical framework for modelling the stochastic fluctuations in market prices is based on the theory of *queueing networks*. A special class of queueing models are the *polling systems*. A classical polling system consists of multiple queues and a single server that visits the queues following a service policy.

It is well known that a polling system is positive recurrent (admitting a unique stationary distribution) if and only if the total loading of the system is less than one. In the transient regime, i.e., total loading greater than one, it has been proved that a polling system with three queues is asymptotically periodic for almost every choice of service policy. When the polling system has more than three queues, characterizing its behaviour in the transient regime is widely open.

Under the assumption that the process is transient, the goal of this project is to study the periodic patterns arising from the switching of the server. In particular, this dissertation aims at developing tools to possibly answer some questions raised in a recent article published by I. MacPhee and his coauthors in 2006.

**Keywords:** Queueing Networks, Polling Systems, Transience, Dynamical System, Asymptotic Periodicity, a.s. Convergence.

# Acknowledgements

First of all, I would like to specially thank my supervisor, Prof. José Pedro Gaivão, for always believing in my capabilities to deliver this dissertation and for his advices along its writing. I will definitely take them with me.

I would like to thank my grandparents Maria Antónia and Joaquim for all the support, financial and emotional, that they provided me throughout my school years. I hope that achieving this final stage of my academic path makes them proud.

I would also like to thank my parents for the support and the understanding, even in the moments of greatest stress, and for giving their best in trying to understand how a mathematician's mind works.

To my sister, Nadine, I am grateful for the love and all the advices. She was and still is there for me whenever I need the most.

To my boyfriend, André, for his patience, and for always trying to make me see that life is much easier to deal with if you relax a little bit and smile.

A special thanks also to all my friends, in particular to Bárbara Santos, Beatriz Leal, Breno Gonçalves, Carla Tavares, Catarina Félix, Flávia Moreira and Leonor Lopes, for all the friendship and companionship they provided during my academic path.

To all, my deepest thank you. I am grateful for all the love and understanding that enabled me to be the person that I am today and to start my professional journey as a mathematician. You added happiness, took away the stress and multiplied my eagerness to do my best. You made me the mathematician that I am today – always hoping that every problem has its solution.

# Contents

# List of Figures

# Chapter 1

# Introduction

The need to avoid congestion (e.g., traffic situations, at supermarket or elevators waiting lines) on a daily basis increased the interest on *queueing models* that describe situations in which customers request service from a server. A *polling system* is an example of a queueing model in which customers arrive to one of the multiple waiting queues according to some switching rule, service policy and switch-over time, and where customers issues are solved by a single-server.

There are many applications (e.g., see [7]) involving polling systems:

- communication networks, with the aim of improving ways to share information between two entities;
- the increase of flexibility in manufacturing and production systems due to the appearance of multi-task machines;
- traffic signal control.

Recently, queueing systems have gained a renewed interest in financial markets, mainly to understand if investors' behaviour influences the stochastic fluctuations in market prices. *Agent-Based Models*, for example, consider that there is a relation between behavioural qualities of investors and quantitative features of the stock price process, i.e., asset prices must be modelled as stochastic processes in a random environment. Taking into account that, in real markets, buying and selling orders occur in different points in time, almost all automated financial trading systems are based on electronic *order books*, in which all unexecuted limit orders are stored and displayed while awaiting execution [2]. Thus, understanding queueing behaviour plays an important role in short-term market dynamics [3].

In this thesis, it will only be considered exhaustive polling systems in a transient regime, where the service rate in each queue is greater than the arrival rate and the switching of the server is made instantaneously.

MacPhee et al. in [6] proved the existence of asymptotically periodic patterns in the transient regime of exhaustive polling systems considering only three queues and a single server. Approximating the stochastic process by a deterministic one, called the *triangle process*, they showed that the sequence of queues visited by the server is eventually periodic, meaning that for almost every realization of the process, the sequence of queues visited by the server converges to a periodic pattern. Moreover, they have shown that there are at most four periodic patterns. Numerical evidence shows that, in fact, there are at most three.

Following the same reasoning, the aim of this dissertation is twofold:

**1.** study the existence of asymptotically periodic patterns in symmetric transient polling systems (no preferred queues and equal rates) with three queues. Notice that such specific service policies have not been considered by MacPhee et al. in [6];

**2.** develop an algorithm, using *Mathematica*, for a polling system with four or more queues, to obtain numerical evidence that the server visits the queues in a periodic way.

This dissertation is structured as follows: throughout Chapter 2, it is possible to get in touch with some classical results about these transient exhaustive polling systems (based on [6]), mainly to understand how the system in general works and to see what happens more specifically when it is considered a polling system with three queues only. Taking into account the study displayed in Section 2.2, Chapter 3 makes use of such results in order to show if there is any periodicity in the way that the server visits the three queues for a symmetric case. Concerning a transient exhaustive polling system with four or more queues, little or nothing is known. Chapter 4 presents a simulation related with this type of system whose results are shown in Chapter 5.

# Chapter 2

# Transient Exhaustive Polling Systems

## 2.1 System description

Consider an exhaustive polling system with $N$ nodes where customers queue and a single server that switches between nodes by a switching rule that considers not only the queue lengths but also its significance. At a current queue $i$, customers arrive according to a Poisson process at a rate $\lambda_i, i \in \{1, ..., N\}$ and each service is completed at a mean time $\mu_i^{-1}$ with i.i.d. service times having finite second moments. The server only switches instantaneously to the next queue $j, j \in \{1, ..., N\}, j \neq i$, when it is empty (all customers are served including any that arrived during the process) and chooses the next queue following a switching rule. Also, the arrival Poisson processes are independent between them.

### 2.1.1 Switching rule

An $N \times N$ matrix $B = (b_{i,j})$, with positive entries, is called a *switching rule*. Associated to a switching rule we define a *switching function*

$$s_i(x) = \arg\max_{j \neq i} b_{ij} x_j, \quad i, j = 1, 2, ..., N, \tag{2.1}$$

where $b_{ij}$ is the weight of queue $j$ having emptied queue $i$ and $x_j$ is the number of customers in queue $j$.

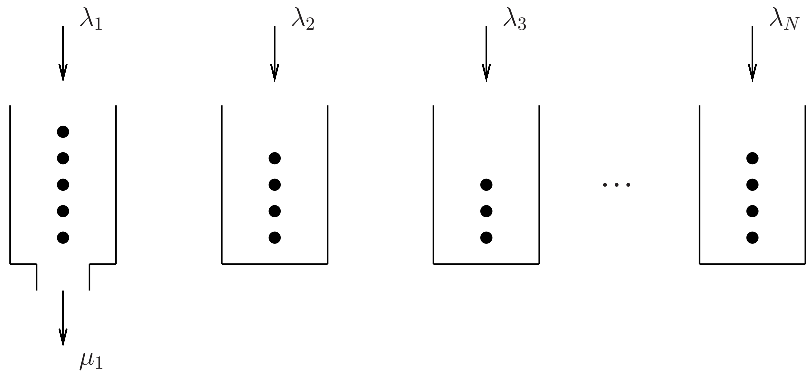



***Figure 2.1.*** *Illustration of a polling system with N queues*

In order to show if there is any periodicity in the queues visited by the server, these arrival and service processes can be modelled by a well defined, irreducible and aperiodic homogeneous *Markov chain* in discrete time with state space $\mathbb{N}_0^N \times \{1, ..., N\}$:

$$\Xi = \{(\xi_n, s_n)\}, \quad n = 0, 1, 2, ..., \tag{2.2}$$

where $\xi_n$ represents the number of customers in each queue and $s_n$ the server location immediately after the $n^{th}$ service. The average behaviour of each queue is:

$$\mathbb{E}\left[\xi_{n+1}^i - \xi_n^i \mid (\xi_n, s_n) = (x, j)\right] = \lambda_i \mu_j^{-1} - I_{\{i=j\}}, \quad i = 1, 2, ..., N, \tag{2.3}$$

for $x \in \mathbb{N}_0^N$ with $x_j \geq 1$ and the server at queue $j \in \{1, ..., N\}$ where $I_{\{i=j\}}$ takes value 1 when $i = j$ and 0 otherwise.

**Definition 2.1.1.** A Markov chain is *irreducible* if for any states

$$e = ((y_1, y_2, ..., y_N), i) \text{ and } \tilde{e} = ((\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_N), \tilde{i}),$$

where $(y_1, y_2, ..., y_N), (\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_N) \in \mathbb{R}_+^N$ and $i, \tilde{i} \in \{1, ..., N\}$, there is a positive probability of moving from $e$ to $\tilde{e}$.

The following quantity determines the *stability* of the process.

**Definition 2.1.2.** The *total loading* is defined by

$$\rho = \sum_{i=1}^{N} \rho_i, \tag{2.4}$$

where $\rho_i = \dfrac{\lambda_i}{\mu_i}$ at each queue $i$.

The proof of the next result can be found in [4, 6].

**Theorem 2.1.1.**

1. *The process* $\Xi$ *is positive recurrent if* $\rho < 1$ *and transient if* $\rho > 1$*;*

2. *The system is stable (converges to a unique stationary distribution) if and only if* $\rho < 1$*.*

With the purpose of studying the switching sequence, and in such a way that the server in the stochastic process does not remain serving at the same queue indefinitely, from now on the following conditions (which characterize a polling system in a transient regime) will be assumed:

$$\rho_i < 1, \ \ i = 1, 2, ..., N \ \text{ and } \ \rho = \sum_{i=1}^{N} \rho_i > 1. \tag{2.5}$$

### *2.1.2 Deterministic model*

Consider a particle moving in $\mathbb{R}_+^N \times \{1, ..., N\}$ with linear dynamics whose velocity is given by expression (2.3). If the server is at queue $j \in \{1, ..., N\}$, the velocity of the particle at any point $y = (y_1, y_2, ..., y_N)$ with $y_j > 0$ is

$$\mu_j^{-1} \sum_{i=1}^{N} \lambda_i e_i - e_j = \mu_j^{-1} \left( \sum_{i \neq j} \lambda_i e_i + (\lambda_j - \mu_j) e_j \right), \tag{2.6}$$

where $e_i$ denote the axial unit vectors in $\mathbb{R}^N$. Note that, when the hyperplane $\{y_j = 0\}$ that passes through the origin is reached, the velocity of the particle changes instantaneously due to the service switching rule.

Starting from an initial condition $y^{(0)} \in \partial\mathbb{R}_+^N$, it is denoted by $y^{(n)} \in \partial\mathbb{R}_+^N$, the position of the particle at the exact moment when the velocity changes. The sequence $y^{(n)}, n = 0, 1, 2, ...,$ can be determined using the linear transformations $T_j : \partial\mathbb{R}_+^N \to \partial\mathbb{R}_+^N$ defined by

$$T_j(y) = \sum_{i \neq j} \left( y_i + \frac{\lambda_i y_j}{\mu_j - \lambda_j} \right) e_i. \tag{2.7}$$

This means that from the starting point $y^{(0)}$ and an initial position $j_0 \in \{1, ..., N\}$ of the server,

$$\begin{cases} y^{(n)} = T_{j_n}(y^{(n-1)}) \\ j_n = s_{j_{n-1}}(y^{(n-1)}) \end{cases} , \ \ n = 1, 2, ..., \tag{2.8}$$

where $j_n$ is the sequence visited by the server. This is called the *deterministic system.*

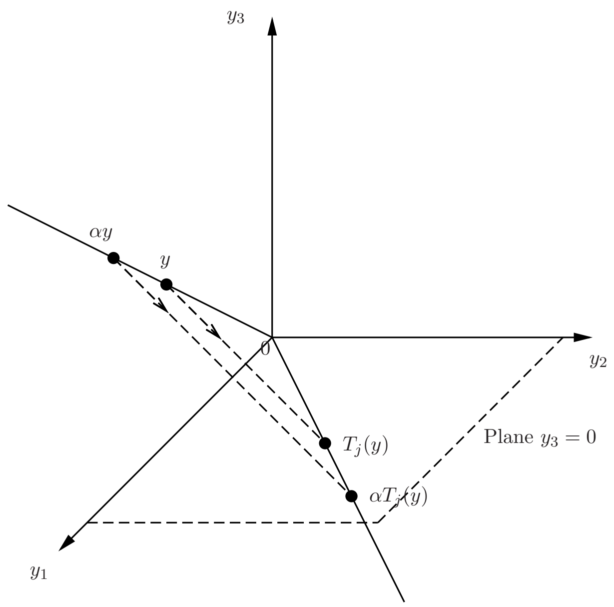



***Figure 2.2.*** *Example of a deterministic model's trajectory in* $\mathbb{R}^3_+$

In matrix form, the linear transformation $T_j$ can be represented by the $N \times N$ matrix

$$T_j = I + C_j, \tag{2.9}$$

where $C_j = \sum_{i=1}^N \alpha_{ij} e_i e_j^T$ with $\alpha_{ij} = \begin{cases} \dfrac{\lambda_i}{\mu_j - \lambda_j}, & i \neq j \\ -1, & i = j \end{cases}$.

Denote by $(\partial\mathbb{R}^N_+)_j$ the hyperplane $\{y_j = 0\}$.

**Lemma 2.1.2.** *$T_j$ is a projection of $\mathbb{R}^N_+$ onto $(\partial\mathbb{R}^N_+)_j$, i.e.,*

- *range $(T_j) = (\partial\mathbb{R}^N_+)_j$;*
- *$T_j^2 = T_j$.*

*Proof.* The first claim is obvious. To prove the second claim, we have

$$T_j T_j = (I + C_j)(I + C_j) = I + C_j + C_j + C_j C_j,$$

where

$$\begin{aligned} C_jC_j &= \left(\sum_{i=1}^{N} \alpha_{ij}e_ie_j^T\right)\left(\sum_{l=1}^{N} \alpha_{lj}e_le_j^T\right) \\ &= \sum_{i,l=1}^{N} \alpha_{ij}\alpha_{lj}(e_ie_j^T)(e_le_j^T) \\ &= \sum_{i,l=1}^{N} \alpha_{ij}\alpha_{lj}e_i(e_j^Te_l)e_j^T \\ &= \sum_{i=1}^{N} \alpha_{ij}\alpha_{jj}e_ie_j^T \\ &= -\sum_{i=1}^{N} \alpha_{ij}e_ie_j^T \\ &= -C_j. \end{aligned}$$

Then,

$$\begin{aligned} T_j^2 &= I + 2C_j + C_jC_j \\ &= I + 2C_j - C_j \\ &= I + C_j \\ &= T_j. \end{aligned}$$

■

### 2.1.3 Compactification

In this subsection, it will be introduced a compactification that will allow the study of the projections $T_j$.

For any linear transformation, it is possible to define its version (or projective) by making a suitable change of coordinates. In this case, a point in $\mathbb{R}_+^N$ can be projected onto the *simplex* $\Delta^{N-1} = \left\{y \in \mathbb{R}_+^N : \sum_{i=1}^N y_i = 1\right\}$ according to the map

$$\Lambda : \mathbb{R}_+^N \setminus \{0\} \to \Delta^{N-1} \text{ where } \Lambda(y) = \frac{y}{y_1 + ... + y_N}. \tag{2.10}$$

Figure 2.3, for example, illustrates a compactification in $\mathbb{R}_+^3$ and the linear transformation $T_j$.

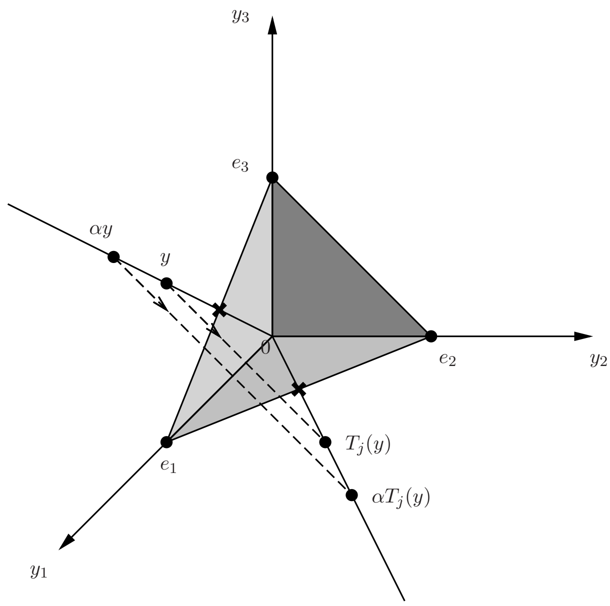



***Figure 2.3.*** *Example of a compactification in* $\mathbb{R}^3_+$ *under* $T_j$

It is obvious that being $T_j$ a linear transformation,

$$T_j(\alpha y) = \alpha T_j(y), \;\; \alpha > 0, \;\; y \in \partial\mathbb{R}^N_+,$$

as shown in Figure 2.2 and 2.3. This means that $T_j(y)$ and $T_j(\alpha y)$ represent the same points in $\Delta^{N-1}$. So, $T_j$ can be restricted to points $\bar{y} \in \partial\Delta^{N-1}$, the boundary of $\Delta^{N-1}$.

**Example 2.1.1.** (Compactification in $\mathbb{R}^2_+$)

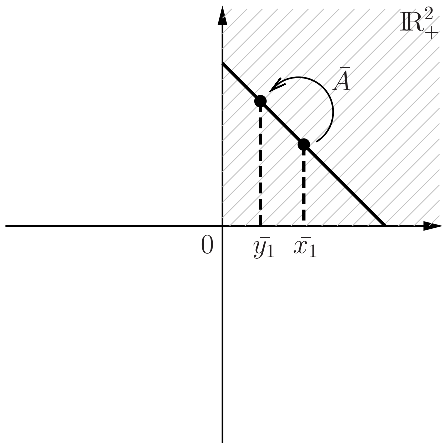
Let $A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$ and consider the unit simplex $\Delta^1$.

Defining $y = Ax = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2x_1 + x_2 \\ x_1 + x_2 \end{pmatrix}$ and making a change of coordinates $x \mapsto \bar{x} = \dfrac{x}{x_1 + x_2}$ we obtain $\begin{cases} \bar{x}_1 = \dfrac{x_1}{x_1 + x_2} \\ \bar{x}_2 = \dfrac{x_2}{x_1 + x_2} \end{cases}$. Then, the projective version $\bar{A}$ of $A$ is given by

$$\bar{y}_1 = \frac{y_1}{y_1 + y_2} = \frac{2x_1 + x_2}{3x_1 + 2x_2} = \frac{\dfrac{2x_1 + x_2}{x_1 + x_2}}{\dfrac{3x_1 + 2x_2}{x_1 + x_2}} = \frac{2\bar{x}_1 + \bar{x}_2}{3\bar{x}_1 + 2\bar{x}_2}$$

and

$$\bar{y}_2 = \frac{y_2}{y_1 + y_2} = \frac{x_1 + x_2}{3x_1 + 2x_2} = \frac{\frac{x_1 + x_2}{x_1 + x_2}}{\frac{3x_1 + 2x_2}{x_1 + x_2}} = \frac{\bar{x}_1 + \bar{x}_2}{3\bar{x}_1 + 2\bar{x}_2}.$$




Note that, $\bar{y}_1, \bar{y}_2 \geq 0$ and $\bar{y}_1 + \bar{y}_2 = 1$. So, in fact, $\bar{y} = (\bar{y}_1, \bar{y}_2) \in \Delta^1$. Since $\bar{x}_2 = 1 - \bar{x}_1$ and $\bar{y}_2 = 1 - \bar{y}_1$, $\bar{A}$ can be written as a function with one variable only:

$$\bar{y}_1 = \frac{2\bar{x}_1 + 1 - \bar{x}_1}{3\bar{x}_1 + 2(1 - \bar{x}_1)} = \frac{\bar{x}_1 + 1}{\bar{x}_1 + 2}.$$

■

### 2.1.4 The queue process

Under the condition $\rho_j < 1, j \in \{1, ..., N\}$ and knowing equation (2.7), it is clear that the trajectory leaving $\bar{y} \in \partial\Delta^{N-1}$ under $T_j$ next reaches $\partial\mathbb{R}^N_+$ in finite time at the point

$$\sum_{i \neq j} \left( \bar{y}_i + \frac{\lambda_i \bar{y}_j}{\mu_j - \lambda_j} \right) e_i$$

with projection

$$\bar{T}_j(\bar{y}) = \sum_{i \neq j} \frac{(\mu_j - \lambda_j)\bar{y}_i + \lambda_i \bar{y}_j}{(\mu_j - \lambda_j) + \mu_j \theta_j \bar{y}_j} e_i \in A^0_j, \tag{2.11}$$

where

$$A^0_j = \left\{ y \in \mathbb{R}^N : \sum_{i=1}^{N} y_i = 1, y_j = 0, y_i \geq 0 \ \text{ for } \ i \neq j \right\}$$

are regions of the switching boundary $\partial\Delta^{N-1}$ and $\theta_j := \mu_j^{-1} \sum_{i=1}^N \lambda_i - 1$.

Taking into account the previous results, it is possible to define a dynamical system $Z = \{z(n)\}, n = 0, 1, ...,$ called the *queue process*, living on $A^0 \equiv \bigcup_{i=1}^N A^0_i = \partial\Delta^{N-1}$. For given $z(0) \in \partial\Delta^{N-1}$,

$$z(n + 1) = \varphi(z(n)), \ \ n = 0, 1, ..., \tag{2.12}$$

where $\varphi(z) = \sum_{\substack{i,j=1 \\ i \neq j}}^{N} I_{\{s_i(z)=j\}} \bar{T}_j(z), z \in A^0$ and $s$ is the switching function defined in equation (2.1). This is an important process in queueing theory since it contains information about the projection of the dynamical system $y(n)$ at the switching moments.

## 2.2 The three queue case

In this section, some interesting results will be discussed about a restricted case of a transient exhaustive polling system with three queues only, where the switching decision is made under the switching rule characterized in equation (2.1).

In a situation where the server is at a certain queue $i$ that just got empty, the switching rule selects the next queue $k$ when $b_{ij}x_j < b_{ik}x_k$ for $i, j, k$ any permutation of $1, 2, 3$. The projection made by $\Lambda$ onto $\Delta^2$ (see rule (2.10)), reduces the switching boundaries $\{x : b_{ij}x_j = b_{ik}x_k \wedge x_i = 0\}$ to *decision points* represented by $d_i$, one on each side of the triangle $A^0$. For the stochastic process $\Xi$, when $b_{ij}x_j = b_{ik}x_k$, the decision for the next server location must be random so, for the queue process $Z$, both possible options at such points will be considered. In this situation, all trajectories of the queue process will be taken into consideration.

From this point on, $Z$ will be called the *triangle process.*

The linear transformation that determines the trajectories is not continuous at the $d_i$. However, for all the combinations of rates and decision points, the trajectories converge toward periodic patterns.

The proof of the following theorems can be found in [6].

**Theorem 2.2.1.** *For almost all switching rules, the corresponding polling system with three queues is asymptotically periodic, i.e., a.s. each trajectory $z(n)$ of the associated triangle process converges to a periodic pattern as $n \to \infty$.*

**Theorem 2.2.2.** *There are no more than four periodic patterns for any triangle process parameters.*

**Theorem 2.2.3.** *There are (infinitely many) choices of the switching rules that lead to the existence of aperiodic trajectories of the triangle process.*

## 2.3 The stochastic triangle process

The last theorems announced in Section 2.2 have important consequences for the behaviour of the underlying stochastic process defined in (2.2). Approximating the polling system by the triangle process, MacPhee et al. in [6] proved that the polling system is periodic whenever the triangle process is.

The proof of the next result can be found in [6] and corresponds to the main result of this article:

**Theorem 2.3.1.** *Suppose the service times have variances $\sigma_i^2 < \infty$. If the queue process $Z$ is asymptotically stable then the corresponding stochastic process of the polling system is also stable, i.e., a.s. each trajectory of the stochastic process converges onto one of the periodic patterns of $Z$.*

# Chapter 3

# The symmetric case of a three queue polling system

MacPhee et al. in [6] obtained very interesting results concerning the theory of queueing models, particularly with regard to the class of exhaustive polling systems with three queues, in the transient regime (see Section 2.2). However, they did not considered the simplest case: a polling system in the same conditions but with equal arrival and service rates in all queues, called the *symmetric* case.

Against this background, this chapter aims to effectively show the veracity of Theorem 2.2.1 but, in this case, a concrete number of patterns should be found.

## *3.1 Detailed analysis*

Consider, for the symmetric case,

$$\lambda_i = \lambda, \;\; \mu_i = \mu \;\text{ and }\; b_{ij} = 1, \;\; i, j = 1, 2, ..., N, \;\; i \neq j. \tag{3.1}$$

Consequently, and according to equation (2.11), the projection of the trajectory leaving $z \in A^0 \setminus A_j^0$ under $\bar{T}_j$ is given by

$$\bar{T}_j(z) = \sum_{i \neq j} \frac{(\mu - \lambda)z_i + \lambda z_j}{(\mu - \lambda) + \mu\theta z_j} e_i \in A_j^0, \;\; j = 1, 2, ..., N, \tag{3.2}$$

with $\theta = \mu^{-1}(N\lambda) - 1 = N\rho - 1$. Simplifying,

$$\bar{T}_j(z) = \sum_{i \neq j} \frac{(1 - \rho)z_i + \rho z_j}{(1 - \rho) + (N\rho - 1)z_j} e_i. \tag{3.3}$$

When $N = 3$, trajectories of the triangle process will be mapped onto the unit simplex $\Delta^2 = \left\{z \in \mathbb{R}^3_+ : \sum_{i=1}^3 z_i = 1, z_i \geq 0\right\}$. A point $z$ with coordinates $(z_1, 0, z_3)$ means that the second queue is empty. This way, the server can relocate to the first or third queue, depending which one has the maximum number of costumers. For example, if $z_1 > z_3$, the server will choose to empty the first queue, which corresponds to apply the transformation

$$\begin{aligned}
\bar{T}_1(z_1, 0, z_3) &= \sum_{\substack{i=1 \\ i \neq 1}}^{3} \frac{(1-\rho)z_i + \rho z_1}{(1-\rho) + (3\rho - 1)z_1} e_i \\
&= \frac{(1-\rho)z_2 + \rho z_1}{(1-\rho) + (3\rho - 1)z_1} e_2 + \frac{(1-\rho)z_3 + \rho z_1}{(1-\rho) + (3\rho - 1)z_1} e_3 \\
&= \left(0, \frac{(1-\rho)z_2 + \rho z_1}{(1-\rho) + (3\rho - 1)z_1}, \frac{(1-\rho)z_3 + \rho z_1}{(1-\rho) + (3\rho - 1)z_1}\right). \qquad (3.4)
\end{aligned}$$

Figure 3.1 shows an example of a trajectory of the triangle process for the two possible cases.

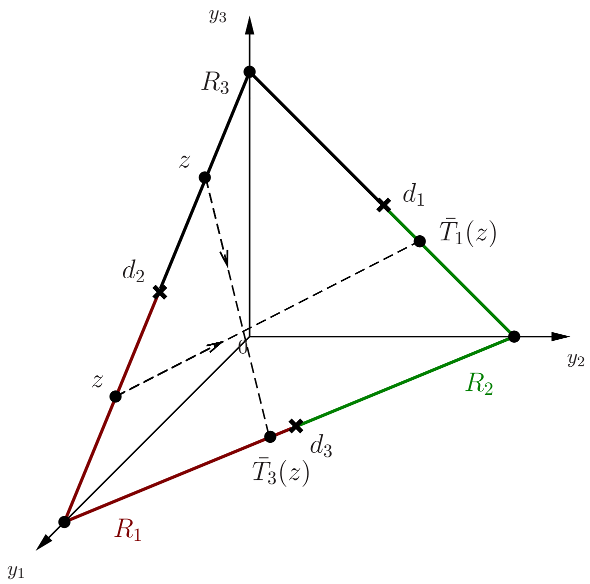



***Figure 3.1.*** *The symmetric triangle process trajectory*

The triangle system is then composed by $\bar{T}_1, \bar{T}_2, \bar{T}_3$ and by the switching rule $s_i(z) = \arg\max_{j \neq i} b_{ij} z_j, \;\; i, j = 1, 2, 3$, where the corresponding process $\varphi : \partial\Delta^2 \mapsto \partial\Delta^2$

consists of

$$\varphi(z) = \begin{cases} \bar{T}_2(z), & z_1 = 0 \wedge s_1(z) = 2 \\ \bar{T}_3(z), & z_1 = 0 \wedge s_1(z) = 3 \\ \bar{T}_1(z), & z_2 = 0 \wedge s_2(z) = 1 \\ \bar{T}_3(z), & z_2 = 0 \wedge s_2(z) = 3 \\ \bar{T}_1(z), & z_3 = 0 \wedge s_3(z) = 1 \\ \bar{T}_2(z), & z_3 = 0 \wedge s_3(z) = 2 \end{cases} . \tag{3.5}$$

In the general case (for any value of $N$ and $b_{ij} > 0$), $\varphi(z) = \bar{T}_j(z)$ if $z_i = 0$ and $s_i(z) = j$.

Given $z \in \partial\Delta^2$, define the boundary regions

$$R_i = \left\{z \in \partial\Delta^2 : \exists j : z_j = 0 \wedge s_j(z) = i\right\}, \tag{3.6}$$

for $i, j = 1, 2, 3, i \neq j$. Therefore, the triangle process can be rewritten as

$$\varphi(z) = \bar{T}_i(z), \;\; z \in R_i, \tag{3.7}$$

where $\bigcup_{i=1}^{3} R_i = \partial\Delta^2 \setminus \bigcup_{i=1}^{3} d_i$ with $d_i = \{z \in \partial\Delta^2 : \exists k \neq j : b_{ij}z_j = b_{ik}z_k \wedge z_i = 0\}$. $D = \bigcup_{i=1}^{3} d_i$ is the set of decision points.

The next step in this analysis will be to transform the triangle process $\varphi$ in a transformation of the interval

$$f : [0, 3] \mapsto [0, 3] \tag{3.8}$$

and study the dynamics of this transformation.

In equation (3.4), may be considered just the coordinate $z_1$ since $z_3 = 1 - z_1$, getting

$$\begin{aligned} \bar{T}_1(z_1, 0, z_3) &= \bar{T}_1(z_1, 0, 1 - z_1) \\ &= \left(0, \frac{(1-\rho)z_2 + \rho z_1}{(1-\rho) + (3\rho - 1)z_1}, \frac{(1-\rho)(1-z_1) + \rho z_1}{(1-\rho) + (3\rho - 1)z_1}\right) \end{aligned} \tag{3.9}$$

which corresponds to apply the transformation

$$h_{21}(x) = \frac{(1-\rho)(1-x) + \rho x}{(1-\rho) + (3\rho - 1)x} = \frac{(2\rho - 1)x + 1 - \rho}{(3\rho - 1)x + 1 - \rho}, \tag{3.10}$$

that is obviously equal to the transformation generated by going from plane $z_3 = 0$ to $z_1 = 0$ ($h_{31}(x)$).

Suppose that, at the moment when queue one goes empty, $z_3 > z_2$. In this case, the projection of the symmetric triangle process trajectories will be equal to

$$\begin{aligned}\bar{T}_3(0, z_2, z_3) &= \sum_{\substack{i=1\\ i\neq 3}}^{3} \frac{(1-\rho)z_i + \rho z_3}{(1-\rho)+(3\rho-1)z_3} e_i \\ &= \frac{(1-\rho)z_1 + \rho z_3}{(1-\rho)+(3\rho-1)z_3} e_1 + \frac{(1-\rho)z_2 + \rho z_3}{(1-\rho)+(3\rho-1)z_3} e_2 \\ &= \left( \frac{(1-\rho)z_1 + \rho z_3}{(1-\rho)+(3\rho-1)z_3}, \frac{(1-\rho)z_2 + \rho z_3}{(1-\rho)+(3\rho-1)z_3}, 0 \right) \\ &= \left( \frac{(1-\rho)z_1 + \rho z_3}{(1-\rho)+(3\rho-1)z_3}, \frac{(1-\rho)(1-z_3) + \rho z_3}{(1-\rho)+(3\rho-1)z_3}, 0 \right) \end{aligned} \tag{3.11}$$

because $z_2 = 1 - z_3$. This corresponds to apply the transformation

$$h_{13}(x) = h_{23}(x) = \frac{(1-\rho)(1-x) + \rho x}{(1-\rho)+(3\rho-1)x} + 1 = \frac{(2\rho-1)x + 1 - \rho}{(3\rho-1)x + 1 - \rho} + 1. \tag{3.12}$$

In a similar way, a transformation resulting from the server switching to the second queue, as a consequence of $z_2 > z_1$, can be found:

$$\begin{aligned}\bar{T}_2(z_1, z_2, 0) &= \sum_{\substack{i=1\\ i\neq 2}}^{3} \frac{(1-\rho)z_i + \rho z_2}{(1-\rho)+(3\rho-1)z_2} e_i \\ &= \left( \frac{(1-\rho)z_1 + \rho z_2}{(1-\rho)+(3\rho-1)z_2}, 0, \frac{(1-\rho)z_3 + \rho z_2}{(1-\rho)+(3\rho-1)z_2} \right) \\ &= \left( \frac{(1-\rho)(1-z_2) + \rho z_2}{(1-\rho)+(3\rho-1)z_2}, 0, \frac{(1-\rho)z_3 + \rho z_2}{(1-\rho)+(3\rho-1)z_2} \right) \end{aligned} \tag{3.13}$$

which, again, corresponds to apply

$$h_{32}(x) = h_{12}(x) = \frac{(1-\rho)(1-x) + \rho x}{(1-\rho)+(3\rho-1)x} + 2 = \frac{(2\rho-1)x + 1 - \rho}{(3\rho-1)x + 1 - \rho} + 2. \tag{3.14}$$

Taking into account the above results, it is possible to define a single function that will be an important basis in the search for periodic patterns. Defining

$$g(x) = \frac{(2\rho-1)x + 1 - \rho}{(3\rho-1)x + 1 - \rho}, \tag{3.15}$$

the transformation $f$ for this particular case will be

$$f(x) = \begin{cases} g\left(x+\frac{1}{2}\right)+\frac{3}{2}-g\left(\frac{1}{2}\right), & x \in \left[0, \frac{1}{2}\right[ \\ g\left(x-\frac{1}{2}\right)+\frac{5}{2}-g\left(\frac{1}{2}\right), & x \in \left[\frac{1}{2}, \frac{3}{2}\right[ \\ g\left(x-\frac{3}{2}\right)+\frac{1}{2}-g\left(\frac{1}{2}\right), & x \in \left[\frac{3}{2}, \frac{5}{2}\right[ \\ g\left(x-\frac{5}{2}\right)+\frac{3}{2}-g\left(\frac{1}{2}\right), & x \in \left[\frac{5}{2}, 3\right] \end{cases}. \tag{3.16}$$

The graph of this function is depicted in Figure 3.2.

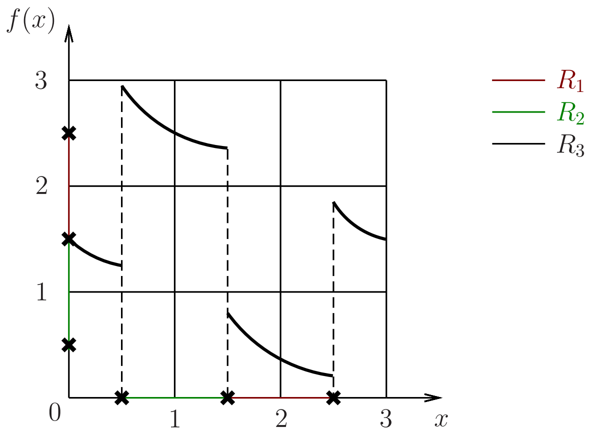



***Figure 3.2.*** *Graphical representation of the transformation f*

## 3.2 Web diagram

A graphical procedure, known as *web diagram*, can be used to visualize successive iterations of a function $f(x)$, representing the orbit of $x$ [8, 9]. In particular, the segments of the diagram connect the points

$$\{(x, f(x)), (f(x), f^2(x)), (f^2(x), f^3(x)), (f^3(x), f^4(x)), ...\}. \tag{3.17}$$

**Definition 3.2.1.** $\mathscr{O} = \{x, f(x), f^2(x), f^3(x), f^4(x), ...\}$, called the *orbit* of $x$, is the set of points formed by the sequence

$$\begin{cases} x_0 = x \\ x_1 = f(x_0) \\ x_2 = f^2(x_0) = f(x_1) \\ \vdots \\ x_{n+1} = f(x_n) \end{cases} \quad \text{where } n \geq 0, \;\; n \in \mathbb{N}_0. \tag{3.18}$$

**Definition 3.2.2.** $x$ is a *periodic point* of period $p$ if $p$ is the smallest positive integer such that $f^p(x) = x$, where $f^p = f \circ f \circ \ldots \circ f$, $p$ times. This way, its orbit $\mathscr{O} = \{x, f(x), f^2(x), ..., f^{p-1}(x)\}$ is a finite set known as *periodic orbit*. When $p = 1$, $f(x) = x$ and $x$ is said to be a *fixed point* of $f$ [1].

The following theorem is a particular case of the *Banach fixed-point theorem*:

**Theorem 3.2.1.** *Let $f : I \to I = [a, b]$ where $f \in C^1$ and $|f'(x)| < 1, \forall x \in [a, b]$. Then, $f$ has a unique fixed point $x^* = f(x^*)$ and, for any $x \in [a, b]$,*

$$\lim_{n \to \infty} f^n(x) = x^*, \tag{3.19}$$

*i.e., the orbit of any point converges to a fixed point.*

For the symmetric three queue polling system case, was verified, using *Mathematica*, the existence of two periodic orbits with period equal to three. This corresponds, in the polling system, to exactly two periodic patterns for the symmetric case characterized in expression (3.1), $\forall \rho \in \left(\frac{1}{3}, 1\right)$.

Note that

$$\sum_{i=1}^{3} \rho_i > 1 \Leftrightarrow 3\rho > 1 \Leftrightarrow \rho > \frac{1}{3} \tag{3.20}$$

and

$$\rho = \frac{\lambda}{\mu} < 1. \tag{3.21}$$

The figures presented in the next page represent two examples of this numerical and analytical evidence:

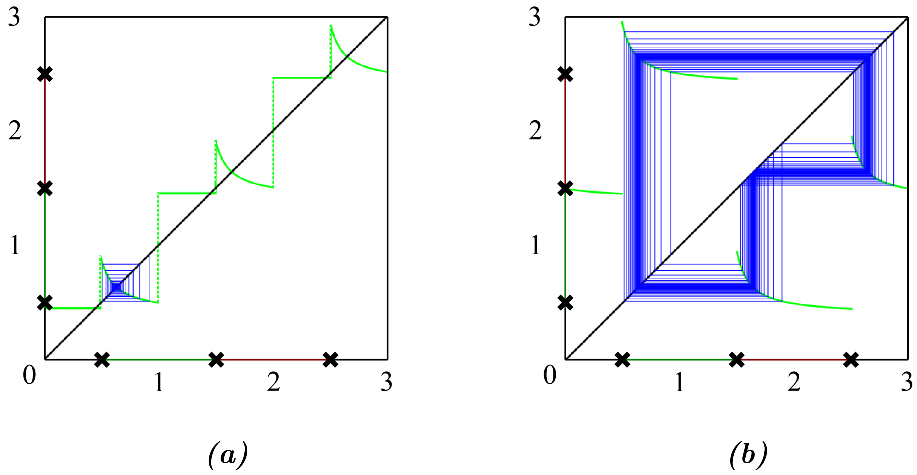



*Figure 3.3.* *(a) Graphical representation of* $f^3$ *and (b) web diagram for an initial point* $x_0 = 0.925$ *and* $\rho = 0.835333$ *with* 205 *iterations*

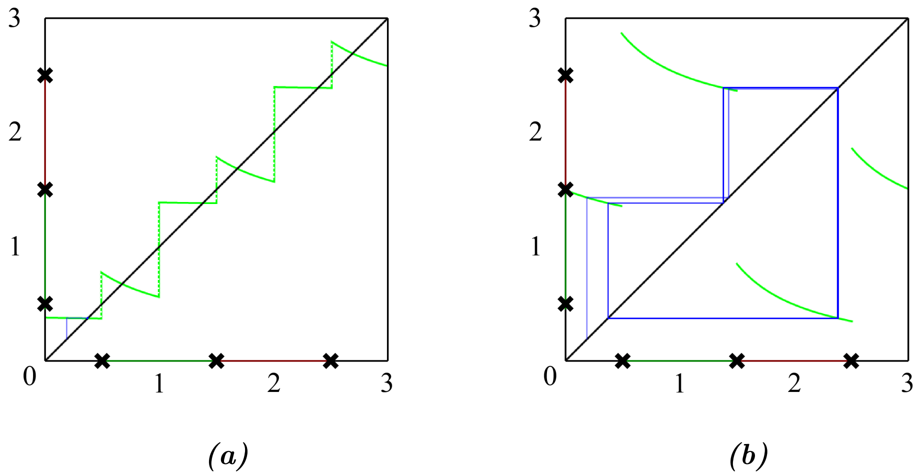



*Figure 3.4.* *(a) Graphical representation of* $f^3$ *and (b) web diagram for an initial point* $x_0 = 0.2$ *and* $\rho = 0.553333$ *with* 205 *iterations*

It is notorious that, for each orbit, the system will converge to one of its three fixed points. Consequently, the following theorem can be announced:

**Theorem 3.2.2.** *In the symmetric case, the triangle process converges to a periodic orbit. There are, in total, two periodic orbits.*

It follows from the results stated in Section 2.3, that the corresponding polling system (transient) converges to a periodic pattern. There are, in total, two periodic patterns.

# Chapter 4

# Simulation

At this stage, one might ask two questions: Is there any periodic pattern in a transient exhaustive polling system with four or more queues? And, the system will always converge (with probability 1) to a periodic pattern?

With the aim of trying to find an answer to these questions, it was created a program in *Mathematica* that simulates an exhaustive polling system with $N$ queues, in a transient regime, with the aid of a function that simulates a Birth-Death process.

## *4.1 Birth-death process*

First, it will be made a brief introduction about Markov chains in continuous time (see [5] for more detail) in order to create an algorithm for the Birth-Death process.

**Definition 4.1.1.** Let $\{X(t) : t \geq 0\}$ be a stochastic process in continuous time with state space $E = \{0, 1, 2, 3, ...\}$. $X$ is a Markov chain if it satisfies the *Markov property*

$$\begin{aligned} P\left(X(t+s) = j \mid X(s) = i, X(s_n) = i_n, ..., X(s_0) = i_0\right) &= P\left(X(t+s) = j \mid X(s) = i\right) \\ &= p_{ij}(t), \end{aligned} \tag{4.1}$$

for all $t > 0, s > s_n > \ldots > s_0 > 0$ and $j, i, i_n, ..., i_0 \in E$.

For an homogeneous Markov chain, the *transition probabilities* are defined by

$$p_{ij}(t) = P\left(X(t) = j \mid X(0) = i\right). \tag{4.2}$$

The *Birth-Death process* is a Markov chain in continuous time that generalises the well known *Poisson process*, in the sense that decreases in the process are allowed. Their transition probabilities $p_{ij}(n)$ are well defined, i.e., the limits

$$q_{ij} = \begin{cases} \lim\limits_{n \to 0} \dfrac{p_{ij}(n)}{n}, & i \neq j \\ \lim\limits_{n \to 0} \dfrac{p_{ij}(n) - 1}{n}, & i = j \end{cases} \tag{4.3}$$

exist and represent the *intensities* of the jumps from state $i$ to $j$, knowing that $p_{ii}(0) = 1$. The intensities of the jumps combined result in the *matrix of intensities* of the process:

$$Q = (q_{ij})_{i,j \in E}. \tag{4.4}$$

It is possible to simulate a Markov chain using the matrix of intensities $Q$ of a stochastic process. In the case of the Birth-Death process, the matrix of intensities will be equal to

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & \dots \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & 0 & \dots \\ 0 & \mu_2 & -(\lambda_2 + \mu_2) & \lambda_2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \tag{4.5}$$

where $\mu_0 = 0, \lambda_0 > 0$ and $\lambda_i, \mu_i > 0, i = 1, 2, ...$ . The Poisson process has a $Q$ matrix similar to matrix (4.5), although without the service rates.

Let

$$\lambda_i = \sum_{i \neq j} q_{ij} \tag{4.6}$$

be the *total intensity* in which $X(t)$ leaves state $i \in E$. The "probability" to occur a jump from state $i$ to $j$ is defined by

$$r_{ij} = \begin{cases} \dfrac{q_{ij}}{\lambda_j}, & i \neq j \\ 0, & i = j \end{cases} \tag{4.7}$$

and $R = (r_{ij})_{ij}$ represents a Markov chain at the jumps' moment. Considering the

Birth-Death process, its easy to verify that

$$R = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots \\ \frac{\mu_1}{\lambda_1+\mu_1} & 0 & \frac{\lambda_1}{\lambda_1+\mu_1} & 0 & \cdots \\ 0 & \frac{\mu_2}{\lambda_2+\mu_2} & 0 & \frac{\lambda_2}{\lambda_2+\mu_2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \tag{4.8}$$

## *4.2 Generating a sample of a single queue*

The simulation of a single queue, in a polling system, may be generated using the Birth-Death process with parameters $\lambda$ (birth or arrival rate) and $\mu$ (death or service rate).

Let $(u_n)_{n\geq 1}$ be a sequence of i.i.d. random real numbers uniformly distributed in $[0,1]$. Let $T_0 = 0$ and $\gamma_0$ be the initial state of the process, i.e., the initial number of elements in the queue. For $n \geq 1$, define the following sequences:

$$t_n = -\frac{1}{\lambda+\mu}\log(1-u_n) \text{ and} \tag{4.9}$$

$$T_n = T_{n-1} + t_n. \tag{4.10}$$

Notice that $t_n \sim Exp(\lambda+\mu)$ and $T_n$ represents the time of occurrence of the $n^{th}$ service or arrival in the queue.

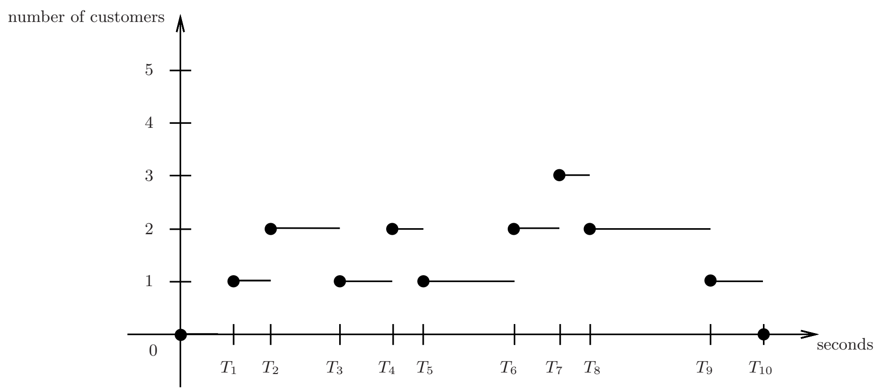



***Figure 4.1.*** *Example of a Birth-Death process*

At time $T_n$, the number of customers in the queue is given by

$$\gamma_n = \gamma_{n-1} + (-1)^{c_n}, \;\; c_n \sim Bernoulli\left(\frac{\mu}{\lambda+\mu}\right) \tag{4.11}$$

with $c_n$ i.i.d.. Thus, the queue can be generated by

$$\xi(t) = \gamma_n, \;\; T_n \leq t < T_{n+1}. \tag{4.12}$$

### 4.2.1 Algorithm

The following algorithm represents the key to perform the simulation of a single queue (for more detail, see Section A.1.1):

**1.** generate a sequence of random reals $u_n$ in the interval $[0, 1]$;

**2.** define $t_n = -\dfrac{1}{\lambda+\mu}\log(1-u_n), \;\; n \geq 1$;

**3.** compute recursively: $T_n = T_{n-1} + t_n, \;\; T_0 = 0$;

**4.** generate a sequence $c_n$ of 0 and 1, where the probability of having $c_n = 1$ is $\dfrac{\mu}{\lambda+\mu}$;

**5.** compute recursively: $\gamma_n = \gamma_{n-1} + (-1)^{c_n}$;

**6.** define the function $\xi(t) = \gamma_n$ if $t \in [T_n, T_{n+1}[$.

## 4.3 Generating a sample of a polling system

In order to perform the simulation of the transient exhaustive polling system with $N$ queues, the parameters can be chosen in the following way:

**1.** generate randomly $\epsilon_i \sim 0$ ($\epsilon_i \in \,]0, 0.1[$, for example) and $\gamma_0 \in \{1, 2, ..., 10\}$ for the first queue. It will be assumed that there are no customers in the remaining queues in the beginning of the simulation;

**2.** define $\rho_i = \dfrac{1}{N} + \epsilon_i, \;\; i = 1, 2, ..., N$. Consequently, $\begin{cases} \lambda_i = 1 + \epsilon_i N \\ \mu_i = N \end{cases}$.

### *4.3.1 Algorithm*

Let $p_i = \{T^i, \xi^i\}$ be the process of a single queue $i \in \{1, ..., N\}$. If $s \in \{1, ..., N\}$ is the queue with the maximum number of customers (here it is considered that all queues have weight equal to one), then $p_s$ can be described by the Birth-Death process algorithm with service rate different than zero and the remaining $p_i, i \neq s$, follow the same process but with zero service rates (which corresponds to a Poisson process with parameter $\lambda_i$).

After queue $s$ goes empty, the server will be relocated to the queue with more customers and the above scheme will be repeated in a predefined number of times $n - 1 \geq 2$.

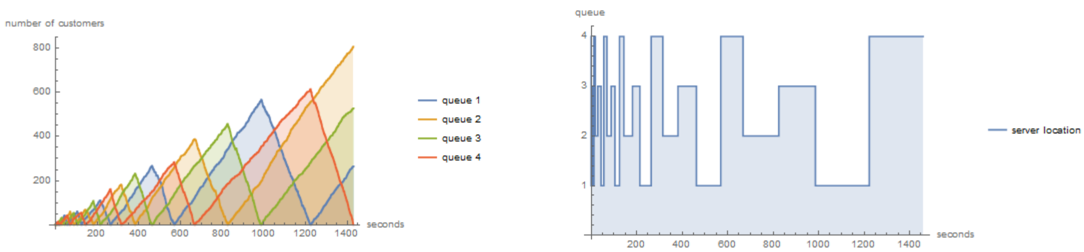
The detailed code based in this algorithm can be found in Sections A.1.2 and A.1.3.

# Chapter 5

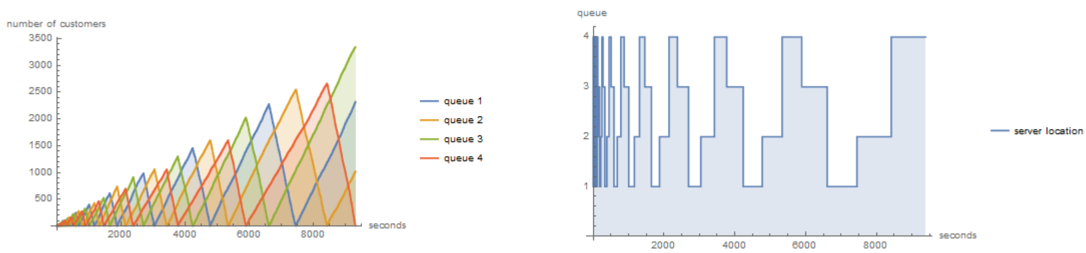
# Numerical results

In this chapter, some results will be presented in relation to the simulation performed.

When simulating a polling system with four waiting queues, it is evident that from a certain moment there is a periodic pattern that remains. Knowing that $n$ refers to the number of times that the server changes queue, the following results were generated:



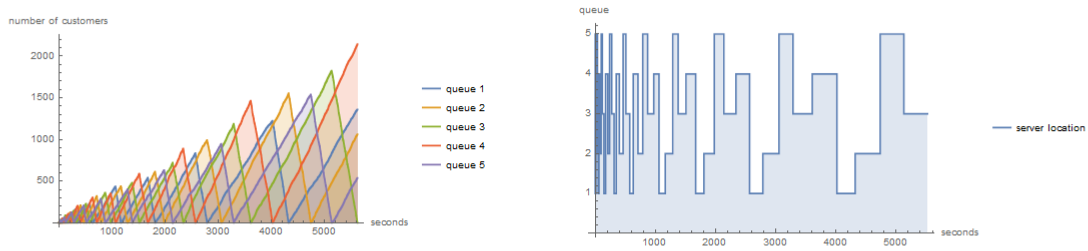

***(a)*** $n = 25,\ Pattern = \{1, 4, 2, 3\}$



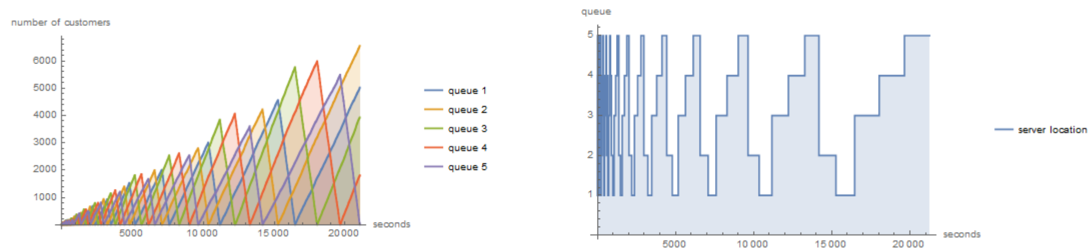

***(b)*** $n = 50,\ Pattern = \{3, 1, 2, 4\}$

***Figure 5.1.*** *Simulation of a polling system with four queues*

The same happens in a polling system with five waiting queues. Although, in this case, a change occurred on the extension of the pattern sequence:




***(a)*** $n = 50,\ Pattern = \{2,5,3,4,1\}$




***(b)*** $n = 75,\ Pattern = \{2,1,3,4,5\}$

***Figure 5.2.*** *Simulation of a polling system with five queues*

As a consequence of the previous results, one can ask the following question:

**Question 5.1.** *For almost all switching rules, the corresponding polling system with four or more queues is asymptotically periodic, i.e., a.s. each trajectory of the associated stochastic process converges to a periodic pattern.*

# Chapter 6

# Conclusions

The analysis performed in this project concerning the dynamics of transient exhaustive polling systems led to very interesting results. I. MacPhee and his coauthors, in 2006, proved the existence of periodic patterns for polling systems with three queues only, under generic conditions. In this thesis, it has been proved that the triangle process, in the symmetric case, exhibits the same behaviour as a generic polling system. However, in this case, the way that the server changes between the three queues suggests that the triangle process converges to one of two periodic orbits.

Additionally, according to the results of the simulation performed in this project, there is evidence of having periodic patterns in a polling system with four or more queues. However, this last statement still needs to be proved analytically.

In the end, lot of open questions in this area still remain open, mostly regarding polling systems with more than four queues, due to difficulties in analysing such cases.

# Bibliography


[1] Baraviera, A. T. and Branco, F. M. (2012). Sistemas dinâmicos: uma primeira visão. *Notas para minicurso do II Colóquio de Matemática da regiao Sul, Universidade Federal do Rio Grande do Sul, Rio Grande do Sul.*

[2] Bayraktar, E., Horst, U., and Sircar, R. (2007). Queueing Theoretic Approaches to Financial Price Fluctuations. *Handbooks in Operations Research and Management Science*, 15:637–677.

[3] Cont, R. and De Larrard, A. (2013). Price dynamics in a markovian limit order market. *SIAM Journal on Financial Mathematics*, 4(1):1–25.

[4] Foss, S., Last, G., et al. (1996). Stability of Polling Systems with Exhaustive Service Policies and State-Dependent Routing. *The Annals of Applied Probability*, 6(1):116–137.

[5] Gaivão, J. P. (2018). Lecture notes in "Processos Estocásticos e Aplicações".

[6] MacPhee, I., Menshikov, M., Popov, S., Volkov, S., et al. (2006). Periodicity in the transient regime of exhaustive polling systems. *The Annals of Applied Probability*, 16(4):1816–1850.

[7] Roubos, A. (2007). Polling Systems and Their Applications. Master's thesis, vrije Universiteit amsterdam.

[8] Weisstein, E. W. "Logistic Map." From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/LogisticMap.html`. Accessed: 2018-08-22.

[9] Weisstein, E. W. "Web Diagram." From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/WebDiagram.html`. Accessed: 2018-08-22.

# Appendix A

# Mathematica code

## A.1 Simulation

### A.1.1 Birth-death process

```
(*returns a truncated birth-death process*)
SampleBirthDeathProcess[\[Lambda]_, \[Mu]_, \[Xi]0_, tmin_, tmax_,
    truncate_] := Module[{aux, cust, T = {tmin}, X = {\[Xi]0}, t},
    While[And[Last[T] <= tmax, Or[Last[X] > 0, Not[truncate]]],
     t = 1/(\[Lambda] + \[Mu]) (-Log[1 - RandomReal[]]);
     T = Append[T, Last[T] + t];
     X = Append[X,
        Last[X] + (-1)^
          RandomVariate[
           BernoulliDistribution[\[Mu]/(\[Lambda] + \[Mu])]]];
     ];
    If[Not[truncate], T[[-1]] = tmax];
    Transpose@{T, X}
    ];
```

### A.1.2 Polling System with N queues

```
(*polling system*)
SamplePollingSystem[\[Lambda]_, \[Mu]_, \[Xi]0_, n_, tmin_,
    tmax_] :=
  Module[{Ts = {0}, S, s,
    p = Table[Table[Subscript[m, i, j], {i, 1}, {j, 2}],
      Length[\[Lambda]]], aux, tinitial, snext, \[Xi],
    pnew = Table[Table[Subscript[m, i, j], {i, 1}, {j, 2}],
      Length[\[Lambda]]]},
   s = Ordering[\[Xi]0, -1][[1]];
   S = {s};
   p[[s]] =
    SampleBirthDeathProcess[\[Lambda][[s]], \[Mu][[s]], \[Xi]0[[
   s]],
     tmin, tmax, True];
   aux = p[[s]][[-1, 1]];

   Do[p[[i]] =
     SampleBirthDeathProcess[\[Lambda][[i]], 0, \[Xi]0[[i]],
   tmin,
      aux, False], {i, Cases[Range[Length[\[Lambda]]], Except[s
   ]]}];
   Do[tinitial = aux;
    \[Xi] = Table[p[[i]][[-1, 2]], {i, Length[\[Lambda]]}];
    snext = Ordering[\[Xi], -1][[1]];
    Ts = Append[Ts, tinitial];
    S = Append[S, snext];
    pnew[[snext]] =
     SampleBirthDeathProcess[\[Lambda][[snext]], \[Mu][[
       snext]], \[Xi][[snext]], tinitial, tmax, True];
    p[[snext]] = Join[p[[snext]], pnew[[snext]]];
    aux = pnew[[snext]][[-1, 1]];

    Do[p[[i]] =
```

```
        Join[p[[i]],
         SampleBirthDeathProcess[\[Lambda][[i]], 0, \[Xi][[i]],
          tinitial, aux, False]], {i,
        Cases[Range[Length[\[Lambda]]], Except[snext]]}];
      , {n}];
    {p, Transpose@{Ts, S}}
    ];

(*generates the graph of the polling system*)
PlotSample[ListProcess_] :=
  ListStepPlot[
   Table[Legended[ListProcess[[k]], Row[{"queue ", k}]], {k,
     Length[ListProcess]}], PlotRange -> Full, Filling -> Axis,
   AxesLabel -> {"seconds", "number of customers"}, ImageSize ->
    400];
```

### A.1.3 Polling System with N queues with variable parameters

```
(*simulation with variable parameters*)
Manipulate[
 Module[{\[Epsilon], sys, \[Lambda], \[Mu], aux\[Xi]0, \[Xi]0},
  \[Epsilon] = RandomReal[{0, 0.1}, q];
  \[Lambda] = 1 + \[Epsilon] q; (*arrival rates*)
  \[Mu] = ConstantArray[q, Length[\[Lambda]]]; (*service rates*)
  aux\[Xi]0 = ConstantArray[0, Length[\[Lambda]] - 1];
  \[Xi]0 =
   PrependTo[aux\[Xi]0,
    RandomInteger[{1, 10}]]; (*initial number of customers*)
  sys = SamplePollingSystem[\[Lambda], \[Mu], \[Xi]0, n, 0,
   100000];
  {StringForm["Pattern=``",
    FindTransientRepeat[sys[[2]][[;; , 2]], 2][[2]]],
```

```
  GraphicsRow[{PlotSample@sys[[1]],
    ListStepPlot[
     Table[Legended[{sys[[2]]}[[k]], "server location"], {k,
       Length[{sys[[2]]}]}], PlotRange -> Full, Filling -> Axis
  ,
     AxesLabel -> {"seconds", "queue"}, ImageSize -> 400]},
   ImageSize -> {1100, 600}, Spacings -> 50]}
 ], {{n, 2, "number of times that the server changes queue"},
  1, 100,
  1}, {{q, 4, "number of queues"}, 3, 20, 1}]
```