Recurrent Generation of Verhulst Chaos Maps at Any Order and Their Stabilization Diagram by Anticipative Control

Daniel M. Dubois

asbl CHAOS, Centre for Hyperincursion and Anticipation in Ordered Systems,
Institute of Mathematics, B37, UNIVERSITY OF LIEGE,
12, Grande Traverse, B-4000 LIEGE 1, Belgium
Daniel.Dubois@ulg.ac.be
ulg.ac.be/mathgen/CHAOS

1 Introduction

In 1838, P.-F. Verhulst [1] proposed an extension of the Malthus equation of the continuous growth of a population. In the Malthus equation, the time derivative of the population is directly proportional to the population, and gives a solution as a temporal exponential curve without limitation and tending to infinity. Verhulst introduced, to the Malthus equation, a negative term, proportional to the square of the population, with a view to obtaining a stable stationary finite state of the population. The Verhulst nonlinear equation is called the logistic equation, for which an analytical solution exists.

For a non-continuous growth of a population, the differential equation is traditionally transformed with the Euler algorithm to a discrete equation, called the Verhulst map in the framework of chaos theory. This chaos map gives solutions of several types depending on the value of the growth parameter of the population: stable fixed points, bifurcations and then chaos. Simple analytical solutions of this chaos map exist for particular values of the growth parameter. In the chaos zone, this solution depends directly on the initial population, so the future evolution of the population is not predictable for long times if the initial condition is not known with precision. This phenomenon is called the sensitivity to initial conditions.

The purpose of this chapter is first to demonstrate that the Verhulst map is not the correct discrete equivalent to the Verhulst logistic equation. Due to the discrete time interval, the square of the population must be transformed to the product of the population at time t and the same population at the following discrete time $t + \Delta t$, giving a non-recursive equation called an incursive equation (for inclusive or implicit recursive equation) [4]. The solution of this incursive discrete equation is similar to the stable stationary solution of the differential Verhulst logistic equation. The same result can be obtained from an anticipative control to the Verhulst map.

Secondly, it will be shown that the Verhulst chaos map belongs to a class of chaos maps at any order n, where n is the degree of the map. In this class, the Verhulst map depends on the power two, $n = 2$, of the population. The concept of *canonical chaos map* will be introduced. The set of canonical chaos maps is given by the Tchebychev polynomials that verify a very simple relation of recurrence. Closed form solutions exist for these canonical chaos maps. This set of canonical chaos maps will be transformed to a set of chaos maps depending on a growth parameter.

Thirdly, this chapter will deal with the control of these chaos maps at any order, with the method of the *incursive predictive control* that belongs to the class of model predictive controls for which the model is the equation of the map itself, and for which no setpoint is defined [5]. Such a control transforms all the unstable states of the chaos maps to stable fixed points. A new type of diagram will be introduced, which we shall call the *stabilization diagram*. The resulting stabilization diagrams show a number of stable fixed points directly related to the order of the maps. Numerical simulations will be performed on these chaos maps for various orders: the first return map, its bifurcation diagram and its stabilization diagram will be displayed as a function of the growth parameter.

2 Analytical Solution of Chaos Maps

This section deals with the solution of the Verhulst logistic equation and the deduction of the so-called Verhulst chaos map. The closed form solution of the chaos map is then demonstrated and its relation to the Shift map is established.

2.1 From the Verhulst Differential Logistic Equation to the Verhulst Chaos Map

The original Verhulst [1] differential growth equation is given by

$$\frac{\mathrm{d}N(t)}{\mathrm{d}t} = rN(t)\left(1 - \frac{N(t)}{K}\right) \tag{1}$$

where $N(t)$ is the value of the population at the current time t, r is the growth rate and K is a limiting growth factor. The well-known analytical solution of this equation is given by

$$N(t) = \frac{\mathrm{e}^{rt}N(0)}{1 + (\mathrm{e}^{rt} - 1)N(0)/K} \tag{2}$$

where $N(0)$ is the initial condition of the population at time $t = 0$. This solution tends to $N(t) = K$, for $t \gg 1/r$. This means that the system loses its initial value and that its future is completely defined by the value of the parameter K, the value of which being fixed. So, such an equation is an anticipative system in the sense that the final future value, K, of the population is already known and completely fixed at the present time.

Let us now deduce the classical discrete version of this equation. In defining $P(t) = N(t)/K$, (1) becomes

$$\frac{\mathrm{d}P(t)}{\mathrm{d}t} = rP(t)[1 - P(t)] \; . \tag{3}$$

With the forward derivative, the discrete equation is given by the well-known Euler algorithm

$$P(t + \Delta t) = P(t) + \Delta t[rP(t) - rP^2(t)] \tag{4}$$

where Δt is the time step. The time step can be taken equal to one (in rescaling the growth rate r) without lack of generality. So, with $\Delta t = 1$, this equation can be rewritten as

$$\frac{P(t+1)}{1+1/r} = \frac{(1+r)P(t) - rP^2(t)}{1+1/r} \; . \tag{5}$$

With the change of variables $x(t) = P(t)/[1+1/r]$ and $a = 1+r$, the Verhulst chaos map is readily obtained:

$$x(t+1) = ax(t)[1 - x(t)] \tag{6}$$

with $a \in [0, 4]$ and $x(t) \in [0, 1]$.

Let us summarize some properties of this map [8].

The stationary states are given by $x(t + 1) = x(t) = x_0$, with $x_0 = 0$ or $x_0 = 1 - 1/a$. The stability criterion of the stationary solutions is given by:

$$\left| \frac{\mathrm{d}x(t+1)}{\mathrm{d}x(t)} \right| = |a - 2ax_0| < 1 \; . \tag{7}$$

For $x_0 = 0$, $|\ \mathrm{d}x(t+1)/\mathrm{d}x(t)\ |=|\ a\ |< 1$, so this stationary solution is stable for $a < 1$. This means that the population disappears due to a negative growth rate $r = a-1 < 0$. The case $a = 1$ corresponds to a null growth rate $r = 0$. For $x_0 = 1 - 1/a$,$|\ \mathrm{d}x(t + 1)/\mathrm{d}x(t)\ |=|\ -a + 2\ |< 1$, so this stationary solution is stable for $1 < a < 3$, and unstable for $a \geq 3$. For $3 \leq a \leq 4$, bifurcations occur (period doubling: $2, 4, 8, 16, \ldots$), and then chaos (period $3, 5, 7, \ldots$ mixed with even periods $6, \ldots$). Chaos begins to occur for the value of a related to the universal constant of Feigenbaum [7, 6]: $a_c = 3.569945672\ldots$ So the chaos zone is defined for $3.569945672\ldots \leq a \leq 4$

2.2 Analytical Solution of the Verhulst Chaos Map

The analytical solution of the Verhulst chaos map (6), for $a = 4$,

$$x(t+1) = 4x(t)[1 - x(t)] \tag{8}$$

with $x(t) \in [0,1]$, is given by

$$x(t) = \frac{1 - \cos(2^t g)}{2} \tag{9}$$

with $g = \arccos[1 - 2x(0)]$ where $x(0)$ is the initial condition at $t = 0$.

This solution of the Verhulst chaos map is an exact closed form solution, because it is not necessary to compute all the successive iterates of the equation to obtain any iterate $x(t)$ for any t. From the fixed value of the initial condition $x(0)$ at time $t = 0$, the variable g is fixed to a constant, and the successive values of $x(t)$ as a function of time t, are given by the calculation of $\cos(2^t g)$ for the values of time $t = 1, 2, 3, \ldots$, that is $\cos(2^1 g), \cos(2^2 g), \cos(2^3 g), \ldots$, and, contrary to the solution of the original Verhulst differential equation, the chaos map never loses its initial condition $x(0)$, via the g function.

So a natural chaos system, if it is not perturbed by external effects, never loses its initial condition, and its future is completely written in its initial condition. But man, who tries to predict the future of this system, is limited by the exact knowledge of the initial condition. Indeed, one can only measure the initial condition with a certain number of decimals. Therefore, the future of this system is written in the successive values of the decimals, in theory, until infinity!

The next section explains the shift of the digits of the initial condition, written in binary.

2.3 The Verhulst Chaos Map Transformed to the Shift Map

In order to explain that the whole future of a chaotic system is written in the digits of its initial condition, let us consider the following change of variables

$$x(t) = \frac{1 - \cos[2\pi y(t)]}{2} = \sin^2[\pi y(t)] \tag{10}$$

in the Verhulst map (8) which becomes successively:

$$\begin{aligned}
&\frac{1 - \cos[2\pi y(t+1)]}{2} = 4\frac{1 - \cos[2\pi y(t)]}{2}\left\{1 - \frac{1 - \cos[2\pi y(t)]}{2}\right\}, \text{ or} \\
&1 - 2\cos[2\pi y(t+1)] = 2\{1 - \cos[2\pi y(t)]\}\{1 + \cos[2\pi y(t)]\}, \text{ or} \\
&1 - 2\cos[2\pi y(t+1)] = 2 - 2\cos^2[2\pi y(t)], \text{ or} \\
&\cos[2\pi y(t+1)] = -1 + \{1 + \cos[4\pi y(t)]\},
\end{aligned}$$

or

$$\cos[2\pi y(t+1)] = \cos[4\pi y(t)] \tag{11}$$

and the following Shift map is obtained

$$y(t+1) = [2y(t)] \bmod 1 \tag{12}$$

with $y(t) \in [0, 1]$, where mod 1 is the modulo 1, which means that only the fractional part of $2y(t)$ is taken.

The exact closed form solution of (12) is given by

$$y(t) = [2^t y(0)] \bmod 1 \; . \tag{13}$$

For example, with $y(0) = 0.3203125\ldots$,
for which $x(0) = \sin^2[\pi y(0)] = 0.7137775467151410471604834284444\ldots$,
the Table 1 gives the successive values of $y(t)$ and $x(t)$.

Table 1. Numerical example of the Shift map and the Verhulst map

t	$y(t)$	$y(t+1)$	binary $y(t)$	$x(t) = \sin^2(\pi y(t))$	$x(t)$	$x(t+1)$
0	0.3203125	0.640625	0.0101001	0.7137775	0.7137775	0.8171966
1	0.640625	0.28125	0.101001	0.8171966	0.8171966	0.5975451
2	0.28125	0.5625	0.01001	0.5975451	0.5975451	0.9619397
3	0.5625	0.125	0.1001	0.9619397	0.9619397	0.1464466
4	0.125	...	0.001	0.1464466	0.1464466	...

In Table 1, the first column gives the time steps $t = 0, 1, 2, 3, 4$, the second column shows the $y(t)$ which are calculated from the shift map in the third column $y(t+1) = 2y(t) \bmod 1$, the fourth column shows the binary value of the decimal $y(t)$, the fifth column shows the $x(t) = \sin^2(\pi y(t))$ calculated from the shift map $y(t)$, the sixth column gives the $x(t)$ calculated from the Verhulst map in column seven $x(t+1) = 4x(t)[1 - x(t)]$.

It is clearly shown in Table 1 that, on one hand, the iterations shift to the left the digits of the binary $y(t)$, and on the other hand that the $x(t)$ calculated from the shift map and from the Verhulst map are identical.

But it is possible to obtain any iterate $y(t)$ at any time t, without computing all the preceding iterates, from the exact closed form solution.

Indeed, from the initial condition: $y(0) = 0.3203125\ldots$, (13) gives
$y(4) = [24y(0)] \bmod 1 = (5.125\ldots) \bmod 1 = 0.125\ldots$
and from (10), one obtains
$x(4) = \sin^2[\pi y(4)] = 0.1464466094067262377995778189475 8\ldots$

Let us now demonstrate that the chaos map does not represent a correct discrete Verhulst equation.

3 The Verhulst Incursive Map is the Correct Discrete Verhulst Equation

I proposed [4], several years ago, to transform the Verhulst chaos map (6) to the following Verhulst incursive map:

$$x(t+1) = ax(t)[1 - x(t+1)] \tag{14}$$

where the limiting factor $[1 - x(t)]$ is defined in the next time step as $[1 - x(t+1)]$.

This incursive equation (for inclusive or implicit equation) can be transformed (see the demonstration in the following section) to the following recursive equation

$$x(t+1) = ax(t)\left[1 - \frac{ax(t)}{1 + ax(t)}\right] = \frac{ax(t)}{1 + ax(t)} \tag{15}$$

where the limiting factor is equal to the whole equation.

This Verhulst incursive map (15) gives a stable solution for any value of a.

Indeed, the two stationary states of this incursive equation are the same as for the chaos map, $x_0 = 0$ and $x_0 = 1 - 1/a$. In applying the criterion of stability

$$\left|\frac{\mathrm{d}x(t+1)}{\mathrm{d}x(t)}\right| = \left|\frac{a}{[1 + ax(t)]^2}\right| < 1\ , \tag{16}$$

it is proved that, firstly, the stationary state $x_0 = 0$,

$$\left|\frac{\mathrm{d}x(t+1)}{\mathrm{d}x(t)}\right| = |a| < 1\ , \tag{17}$$

is stable for $a < 1$, and, secondly, the stationary state $x_0 = (a-1)/a$,

$$\left|\frac{\mathrm{d}x(t+1)}{\mathrm{d}x(t)}\right| = 1/a < 1\ , \tag{18}$$

is always stable for $a > 1$.

The analytical solution of this nonlinear (15) is given by

$$x(t) = \frac{(1 - 1/a)Ca^t}{1 + Ca^t} \tag{19}$$

as a closed form solution, where C is a constant. The initial value $x(0)$ defines the value of C by, $x(0) = (1 - 1/a)C/(1 + C)$, so the solution

$$x(t) = \frac{a^t x(0)}{1 + (a^t - 1)x(0)/(1 - 1/a)} \tag{20}$$

is similar to the solution of the original Verhulst differential logistic (2).

Let us demonstrate that the incursive (15) represents the correct discrete equivalent of the logistic (1). For that, let us write the solution (2) of the Verhulst logistic differential equation as

$$N(t_2) = \frac{\mathrm{e}^{r(t_2 - t_1)}N(t_1)}{1 + (\mathrm{e}^{r(t_2 - t_1)} - 1)N(t_1)/K} \tag{21}$$

which gives the growth of the population from the time t_1 to the time t_2. Indeed, in taking $t_1 = 0$ and $t_2 = t$, the continuous solution (2) is obtained.

Defining a time interval Δt between t_2 and t_1 as $t_2 = t_1 + \Delta t$, (21) becomes

$$N(t_1 + \Delta t) = \frac{\mathrm{e}^{r\Delta t} N(t_1)}{1 + (\mathrm{e}^{r\Delta t} - 1)N(t_1)/K} \tag{22}$$

Choosing $\Delta t = 1$, and $t_1 = t$, one obtains

$$N(t+1) = \frac{\mathrm{e}^{r} N(t)}{1 + (\mathrm{e}^{r} - 1)N(t)/K} \tag{23}$$

and with the change in variables, $\mathrm{e}^r = a$ and $N(t)/K = x(t)/(1 - 1/a)$, (23) is written as

$$x(t+1) = \frac{ax(t)}{1 + ax(t)} \tag{24}$$

which is the discrete (15), and this is the correct discrete algorithm for the Verhulst logistic differential equation.

As result, we conclude that the chaos emerging from the so-called Verhulst chaos map is due to instabilities of the Euler algorithm and not from fundamental biological properties of the Verhulst logistic differential equation.

This (24) can also be obtained from an incursive control as shown in the next section.

4 Incursive Control for Stabilizing Chaos Maps

This section presents a survey of the role of incursive control with an example of control of the Verhulst chaos map, with simulations.

Let us begin by the definitions of recursive and incursive systems.

4.1 Recursive and Incursive Systems Applied to the Verhulst Chaos Map

A recursive system computes its vector current state $\boldsymbol{x}(t)$, at successive time steps $t = 0, 1, 2, \ldots$, from a vector function $\boldsymbol{R}$ of its past and present states as

$$\boldsymbol{x}(t+1) = \boldsymbol{R}(\ldots, \boldsymbol{x}(t-2), \boldsymbol{x}(t-1), \boldsymbol{x}(t); \boldsymbol{p}) \tag{25}$$

where the vector $\boldsymbol{p}$ is a set of parameters.

A *weak* incursive system [4] computes its current state at time t, as a function of its states at past times,..., $t-2, t-1$, present time, t, and even its *predicted* states at future times $t+1, t+2, \ldots$.

$$\boldsymbol{x}(t+1) = \boldsymbol{A}(\ldots, \boldsymbol{x}(t-2), \boldsymbol{x}(t-1), \boldsymbol{x}(t), \boldsymbol{x}^*(t+1), \boldsymbol{x}^*(t+2), \ldots; \boldsymbol{p}) \tag{26}$$

where the future states $\boldsymbol{x}^*(t+1), \boldsymbol{x}^*(t+2), \ldots$, are computed with a predictive model of the system.

A strong incursive system [4] computes its current state at time t, as a function of its states at past times,..., $t-3, t-2, t-1$, present time, t, and even its states at future times $t+1, t+2, t+3, \ldots$.

$$\boldsymbol{x}(t+1) = \boldsymbol{A}(\ldots, \boldsymbol{x}(t-2), \boldsymbol{x}(t-1), \boldsymbol{x}(t), \boldsymbol{x}(t+1), \boldsymbol{x}(t+2), \ldots; \boldsymbol{p}) \tag{27}$$

where the future states $\boldsymbol{x}(t+1), \boldsymbol{x}(t+2), \ldots$, are computed by the system itself.

The Verhulst chaos map (6):

$$x(t+1) = ax(t)(1 - x(t)) \tag{28}$$

is a **recursive system**, and the Verhulst incursive map (14):

$$x(t+1) = ax(t)[1 - x(t+1)] \tag{29}$$

is a **strong incursive system**, because the future value in the saturation factor, $[1 - x(t+1)]$, is computed by the system itself. Indeed, in replacing successively $x(t+1)$ by $ax(t)[1 - x(t+1)]$ in (29), one obtains the following equation

$$x(t+1) = ax(t)(1 - ax(t)(1 - ax(t)(1 - ax(t)(1 - ax(t)(1 - ...))))) \tag{30}$$

which is an infinite recursive equation that converges to

$$x(t+1) = \frac{ax(t)}{1 + ax(t)} \tag{31}$$

which is the (15), given at the preceding section.

This incursive map (29) can be obtained from the following recursive map (28) to be controlled

$$x(t+1) = ax(t)[1 - x(t)] + u(t) \tag{32}$$

by an incursive control $u(t)$

$$u(t) = ax(t)[x(t) - x(t+1)] \tag{33}$$

which can be transformed to a recursive control with (31) as follows

$$u(t) = ax(t)\left[x(t) - \frac{ax(t)}{1 + ax(t)}\right] . \tag{34}$$

Indeed, in including (34) to (32), (31) is obtained. As it was shown in the preceding section, this incursive map is always stable. This incursive controller, which is a powerful tool for stabilizing chaos maps, belongs to a special class of controller, because no setpoint is defined. Such a controller stabilizes by itself the unstable states to stable fixed points.

In the next section, this incursive controller will be compared to model predictive controllers through numerical simulations in order to indicate how to stabilize the chaos of the Verhulst map.

4.2 Incursive and Model Predictive Controls of the Verhulst Chaos Map

Predictive models were developed over the past two or three decades in the field of control, what is referred as the Model-based Predictive Control (MPC). These include Model Predictive Heuristic Control (MPHC) [9], Dynamic Matrix Control (DMC) [3], Internal Model Control (IMC) [10] and Generalized Predictive Control (GPC) [2].

The key difference between a conventional feedback control and a predictive control is that the control error $e = x - r$, which is the difference between the process output x and the setpoint r (the desired output), used by the predictive controller is based on future and/or predicted values of the setpoint $x(t + \tau)$, and also on future and/or predicted values of the process output $x(t + \tau)$, rather than their current values. See [1] for an overview of MPC.

The basic principle of the model predictive control of a discrete system, with a time step Δt,

$$x(t + \Delta t) = F[x(t)] + u(t) \tag{35}$$

where $u(t)$ is a control action, consists in minimizing a cost function J, given for example by a weighted least squares criterion:

$$J = E\left\{\sum_{i=1}^{N}[x(t + i\Delta t) - r(t + i\Delta t)]^2 + \sum_{i=1}^{Nu} w_i u(t - \Delta t + i\Delta t)^2\right\} \tag{36}$$

where $\{r(t + i\Delta t)\}$ is the setpoint sequence (target tracking), $\{w_i\}$ is the weight sequence, and N and Nu are fixed integers representing the time horizons of the predicted outputs, $x(t + i\Delta t)$, and control sequence, $u(t - \Delta t + i\Delta t)$. The anticipative outputs $x(t + i\Delta t)$, $i = 1$ to N, are computed from the model of the process to be controlled. The Verhulst map (28), to be controlled by a control action $u(t)$, in the chaos regime with $a = 4$, is written as

$$x(t + 1) = 4x(t)[1 - x(t)] + u(t) \tag{37}$$

with the cost function (36) given by

$$J = [x(t + 1) - x_0]^2 + w_1 u^2(t) \tag{38}$$

with $\Delta t = 1$, $N = Nu = 1$, with a constant setpoint $r(t + 1) = x_0 = 3/4$, that is the unstable equilibrium of the map [5]. The objective of the control is to stabilize the map at its unstable equilibrium given by $x_0 = 3/4$, obtained from $x_0 = 4x_0(1 - x_0)$, corresponding to the equilibrium condition $x(t+1) = x(t) = x_0$ in (37) [without the control $u(t)$]. Indeed, in applying the classical criterion of stability given by

$$\left|\frac{\mathrm{d}x(t + 1)}{\mathrm{d}x(t)}\right| < 1 \tag{39}$$

to the chaos map at the equilibrium $x_0 = 3/4$, one obtains

$$\left|\frac{\mathrm{d}x(t+1)}{\mathrm{d}x(t)}\right| =\mid 4 - 8x_0 \mid = 2 \tag{40}$$

which is greater then 1, so this equilibrium state x_0 is unstable.

Putting (37) into (38), one obtains the following cost function at the current time

$$J = [4x(t)(1 - x(t)) + u(t) - x_0]^2 + w_1 u^2(t) \ . \tag{41}$$

The minimum of J is obtained with the condition $\mathrm{d}J/\mathrm{d}u(t) = 0$, so

$$u(t) = \frac{x_0 - 4x(t)[1 - x(t)]}{1 + w_1} \ . \tag{42}$$

Now, let us put (42) into (37), to obtain

$$x(t+1) = 4x(t)[1 - x(t)] + \frac{x_0 - 4x(t)[1 - x(t)]}{1 + w_1} \ . \tag{43}$$

After elementary mathematical transformations, (43) becomes

$$x(t+1) = \frac{x_0 + 4w_1 x(t)[1 - x(t)]}{1 + w_1} \ . \tag{44}$$

The equilibrium conditions $x(t+1) = x(t) = x_0$ of (44) are $x_0 = 0$ and $x_0 = 3/4$, as desired.

Applying the stability criterion (39) to (44), for $x_0 = 3/4$, one obtains

$$\left|\frac{\mathrm{d}x(t+1)}{\mathrm{d}x(t)}\right| = \left|\frac{4w_1 - 8w_1 x_0}{1 + w_1}\right| = \left|-\frac{2w_1}{1 + w_1}\right| < 1 \tag{45}$$

and the chaotic map is stabilized to $x = x_0 = 3/4$ for a weight in the range $0 < w_1 < 1$.

Let us show the differences between the incursive control and the model predictive control by considering the control of the chaos map (37), with the three following control functions [5]:

a) The incursive control [Equation (34), in the chaos regime with $a = 4$]:

$$u(t) = 4x(t)\frac{x(t) - 4x(t)[1 - x(t)]}{1 + 4x(t)} \tag{46}$$

b) The model predictive control [Equation (42)]:

$$u(t) = \frac{x_0 - 4x(t)[1 - x(t)]}{1 + w_1} \tag{47}$$

c) The incursive predictive control:

$$u(t) = \frac{x(t) - 4x(t)[1 - x(t)]}{1 + w_1} \tag{48}$$

This third control is based on the incursive control applied to the model predictive control by replacing the setpoint x_0 by $x(t)$ in the cost function (38) as follows

$$J = [x(t+1) - x(t)]^2 + w_1 u^2(t) \tag{49}$$

This new controller (48) is obtained by minimizing this cost function (49).

Let us point out that the control (46) is similar to the control (48) by taking a variable weight $w_1 = 1/4x(t)$.

Figures 1, 2 and 3 show the simulations of the chaos map (37), with the three controllers (46), (47) and (48).

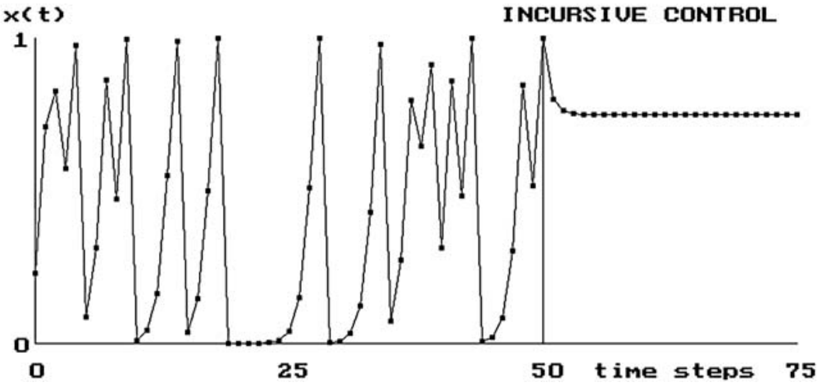

Fig. 1. Incursive Control (46), starting at step 50, of the chaos map

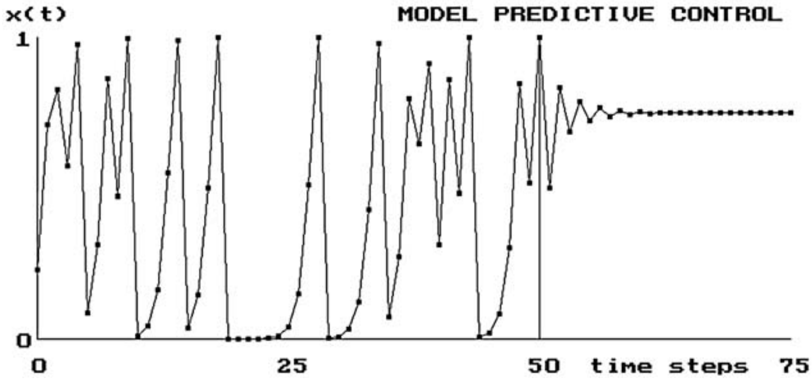

Fig. 2. Model Predictive Control (47), starting at step 50, of the chaos map

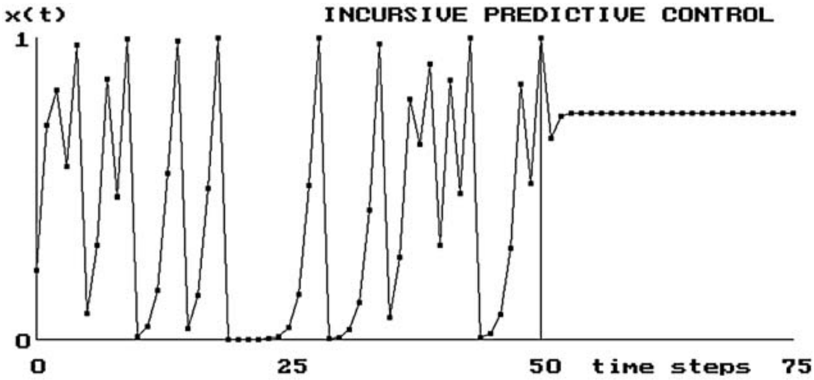

Fig. 3. Incursive Predictive Control (48), starting at step 50, of the chaos map

From the simulations, the best controller is the incursive control (46), in Fig. 1, where it is well seen that the control is optimal.

In Fig. 2, the model predictive control (47), with $w_1 = 1/2$, shows damped oscillations around the setpoint $x_0 = 3/4$. Let us notice that the damping effect depends on the value of the weight w_1.

In Fig. 3, the incursive predictive control (48) shows a better control than the model predictive control, with the same weight w_1. The incursive control and the incursive predictive control, find by themselves the setpoint which is the unstable equilibrium of the chaos map. Indeed, these incursive controllers do not use an explicit setpoint, but an implicit setpoint given by the unstable equilibrium state which is stabilized. The incursive control minimizes the distance between $x(t+1)$ and $x(t)$, instead of the distance between $x(t+1)$ and a setpoint x_0.

Let us now show how to generate Verhulst chaos maps at any order and how to control them.

5 Recurrent Generation of Chaos Maps at any Order

With a view to obtaining a simple rule to generate chaos maps at any order, let us make the following change of variables

$$X(t) = 1 - 2x(t) \tag{50}$$

in the Verhulst chaos map (8) which becomes

$$X(t+1) = 2X^2(t) - 1 \tag{51}$$

with $X(t) \in [1, -1]$, and the exact closed form solution (9) becomes

$$X(t) = \cos(2^t g) \tag{52}$$

with $g = \arccos[X(0)]$ or $X(0) = \cos g$.

The purpose of the next section is to define a set of canonical chaos maps with a general exact closed form solution given by

$$X(t) = \cos(n^t g) \tag{53}$$

for $n = 0, 1, 2, 3, 4, 5, \ldots$ and $g = \arccos[X(0)]$ or $X(0) = \cos g$.

5.1 Canonical Chaos Maps at any Order

The set of canonical chaos maps is defined by the following general equation

$$x(t+1) = \cos[n \arccos x(t)] \tag{54}$$

with $x(t) \in [1, -1]$ and $n \in N$. With the relation

$$\cos[n \arccos(x(t)] = x^n(t) + C_n^2 x^{n-2}(t)[x^2(t) - 1] + C_n^4 x^{n-4}[x^2(t) - 1]^2 + \cdots \tag{55}$$

where $C_n^m = n!/m!(n-m)!$, (54) becomes

$$x(t+1) = T_n[x(t)] = x^n(t) + C_n^2 x^{n-2}(t)[x^2(t) - 1] + C_n^4 x^{n-4}[x^2(t) - 1]^2 + \cdots \tag{56}$$

with $x(t) \in [1, -1]$ and $n \in N$. This equation (56) is the set of canonical chaos maps, for which the exact closed form solution is given by

$$x(t) = \cos(n^t g) \tag{57}$$

for $n = 0, 1, 2, 3, 4, 5, \ldots$ and $g = \arccos[x(0)]$ or $x(0) = \cos g$.

Let us remark that (55) corresponds to what is called the Tchebychev polynomials T_n given by

$$T_n(X) = X^n + C_n^2 X^{n-2}(X^2 - 1) + C_n^4 X^{n-4}(X^2 - 1)^2 + \cdots \tag{58}$$

which gives:

$$\begin{aligned}
T_0 &= 1 \ , \\
T_1 &= X \ , \\
T_2 &= 2X^2 - 1 \ , \\
T_3 &= 4X^3 - 3X \ , \\
T_4 &= 8X^4 - 8x^2 + 1 \ , \\
T_5 &= 16X^5 - 20X^3 + 5X \ , \\
T_6 &= 32X^6 - 48X^4 + 18X^2 - 1 \ , \\
&\ldots
\end{aligned} \tag{59}$$

where T_n is of degree n, and, for $n > 0$, the coefficient of the term of the highest degree of T_n is 2^n.

These Tchebychev polynomials verify a very simple relation of recurrence given by

$$T_{n+2} = 2XT_{n+1} - T_n \tag{60}$$

with the initial conditions $T_0 = 1$ and $T_1 = X$.

With the relation of recurrence (60), it is possible to generate the set of canonical chaos maps as follows:

$$\begin{aligned}
n &= 0 & x(t+1) &= 1 \\
n &= 1 & x(t+1) &= x(t) \\
n &= 2 & x(t+1) &= 2x^2(t) - 1 \\
n &= 3 & x(t+1) &= 4x^3(t) - 3x(t) \\
n &= 4 & x(t+1) &= 8x^4(t) - 8x^2(t) + 1 \\
n &= 5 & x(t+1) &= 16x^5(t) - 20x^3(t) + 5x(t) \\
n &= 6 & x(t+1) &= 32x^6(t) - 48x^4(t) + 18x^2(t) - 1 \\
n &= 7 & x(t+1) &= 64x^7(t) - 112x^5(t) + 56x^3(t) - 7x(t) \\
& & \ldots & \\
n &= 10 & x(t+1) &= 512x^{10}(t) - 128x^8(t) + 112x^6(t) - 400x^4(t) + 50x^2(t) - 1 \\
& & \ldots &
\end{aligned} \tag{61}$$

Introducing a growth parameter a in (56), the set of chaos maps at any order is given by the following map

$$\begin{aligned}
x(t+1) &= aT_n[x(t)] \\
&= a\{x^n(t) + C_n^2 x^{n-2}(t)[x^2(t) - 1] + C_n^4 x^{n-4}[x^2(t) - 1]^2 + \cdots\}
\end{aligned} \tag{62}$$

with $a \in [-1, +1]$ and $x(t) \in [-1, +1]$. This set of chaos maps (62) can be controlled by an action control $u(t)$ as follows

$$x(t+1) = aT_n[x(t)] + u(t) \tag{63}$$

with the following general incursive predictive controller

$$u(t) = \frac{x(t) - aT_n[x(t)]}{1 + w_1} \tag{64}$$

where

$$T_n[x(t)] = x^n(t) + C_n^2 x^{n-2}(t)[x^2(t) - 1] + C_n^4 x^{n-4}[x^2(t) - 1]^2 + \cdots\ , \tag{65}$$

with $a \in [-1, +1]$ and $x(t) \in [-1, +1]$.

In Subsect. 5.2, some numerical simulations of these chaos maps and their stabilization will be given.

5.2 Simulation and Incursive Control of Chaos Maps

In this last subsection, we show the numerical simulation of the chaos maps (62) for $n = 2, 3, 5, 10, 25$, and their stabilization diagram with the incursive predictive control (63), (64) for $n = 2, 3, 5, 10, 25$.

The chaos maps, which are numerically computed, are given by

$$x(t+1) = a[2x^2(t) - 1] \quad \text{for } n = 2 \,, \tag{66}$$
$$x(t+1) = a[4x^3(t) - 3x(t)] \quad \text{for } n = 3 \,, \tag{67}$$
$$x(t+1) = a[16x^5(t) - 20x^3(t) + 5x(t)] \quad \text{for } n = 5 \,, \tag{68}$$

and by

$$x(t+1) = a[512x^{10}(t) - 128x^8(t) + 112x^6(t) - 400x^4(t) + 50x^2(t) - 1] \tag{69}$$

for $n = 10$.

Let us remark that it is also possible to obtain this chaos map $n = 10$ by inserting the canonical map $n = 5$ in the chaos map $n = 2$ as follows:

$$x(t+1) = a\{2T_5^2[x(t)] - 1\} \,. \tag{70}$$

In the same way, the chaos map $n = 25$ is obtained in inserting the canonical chaos map $n = 5$ in the chaos map $n = 5$ as follows
$n = 25$

$$x(t+1) = a\{16T_5^5[x(t)] - 20T_5^3[x(t)] + 5T_5[x(t)]\} \tag{71}$$

with $T_5[x(t)] = 16x^5(t) - 20x^3(t) + 5x(t)$.

Figures 4 and 5 give the first return diagram of the chaos maps $n = 2$ and $n = 10$ for $a = 1$.

Figures 6, 8, 10, 12 and 14 give the bifurcation diagrams of the chaos maps $n = 2, 3, 5, 10, 25$, with $a \in [-1, +1]$.

Figures 7, 9, 11, 13 and 15 give the stabilization diagrams of the chaos maps $n = 2, 3, 5, 10, 25$, with $a \in [-1, +1]$, by using the incursive predictive control.

For example, the control for $n = 3$ is given by

$$x(t+1) = a[4x^3(t) - 3x(t)] + u(t) \,, \tag{72}$$
$$u(t) = \frac{x(t) - a[4x^3(t) - 3x(t)]}{1 + w_1} \,. \tag{73}$$

These stabilization diagrams of the chaos maps show that the number of fixed points is equal to $n = 2, 3, 5, 10, 25$, respectively. The proposition to create such stabilization diagrams could be of great interest in practice. Indeed, it is often impossible to detect, in real time, the unstable points of real chaotic systems. Instead of defining setpoints arbitrarily, it would be more accurate to let the controller self-stabilizes an unstable point.

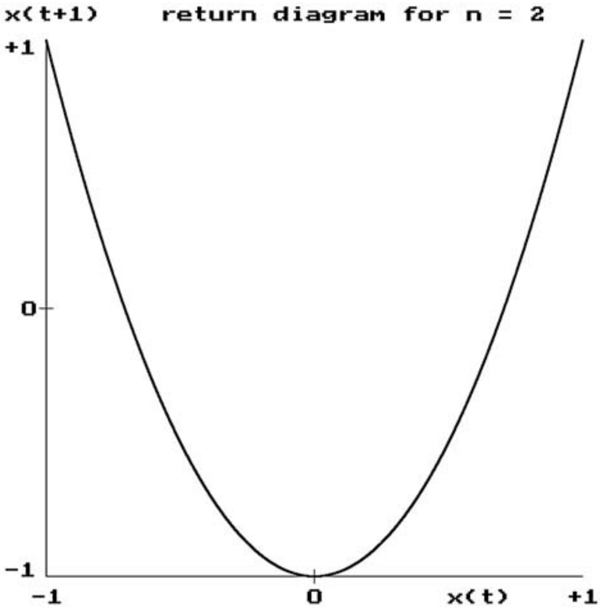

Fig. 4. First return diagram for $n = 2$

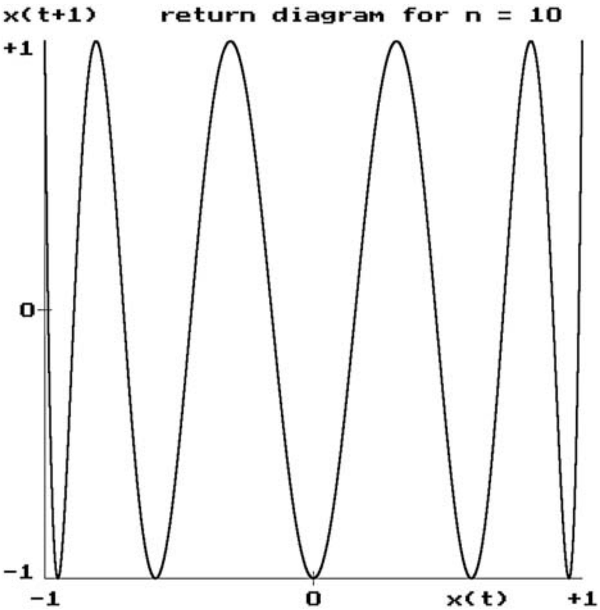

Fig. 5. First return diagram for $n = 10$

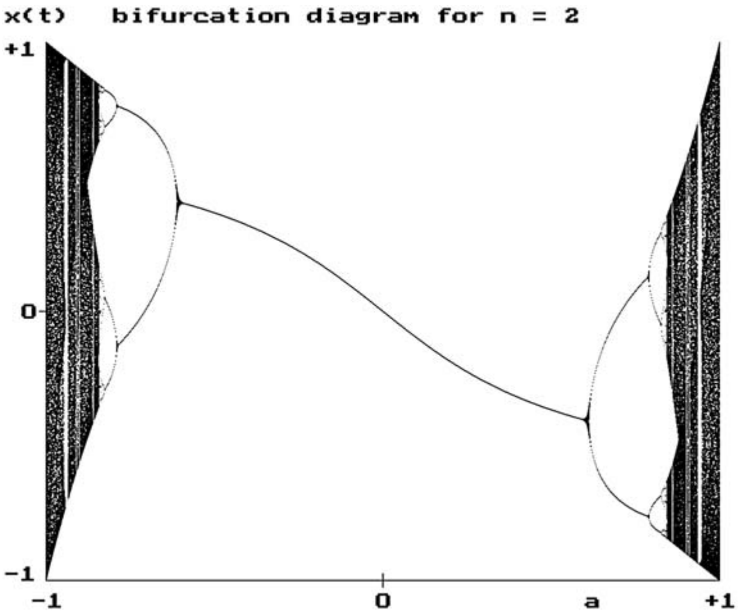

Fig. 6. Bifurcation diagram for $n = 2$

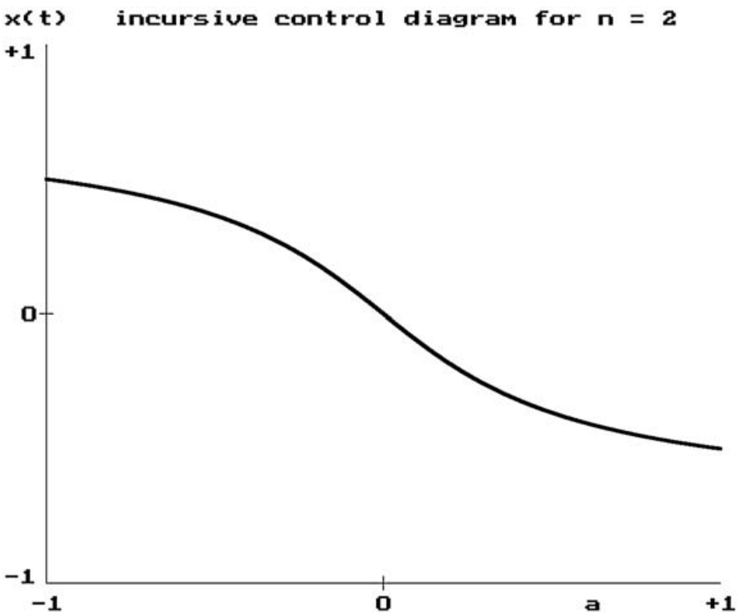

Fig. 7. Stabilization diagram for $n = 2$

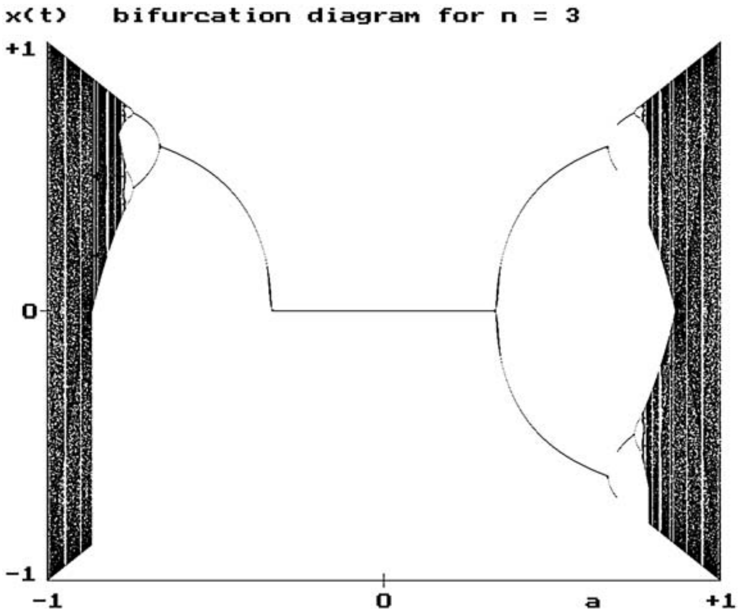

Fig. 8. Bifurcation diagram for $n = 3$

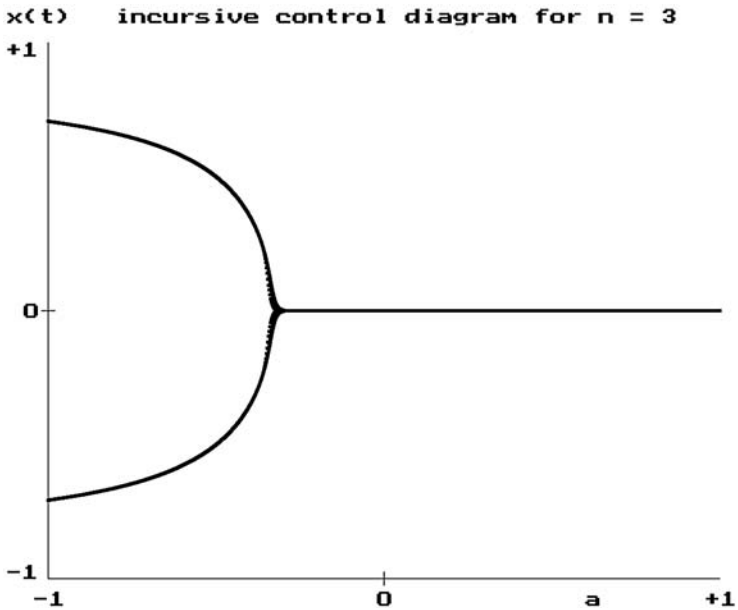

Fig. 9. Stabilization diagram for $n = 3$

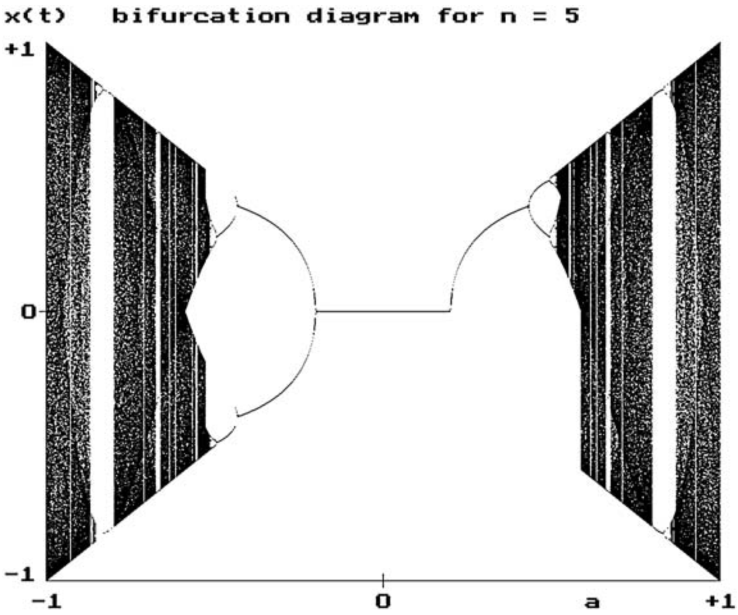

Fig. 10. Bifurcation diagram for $n = 5$

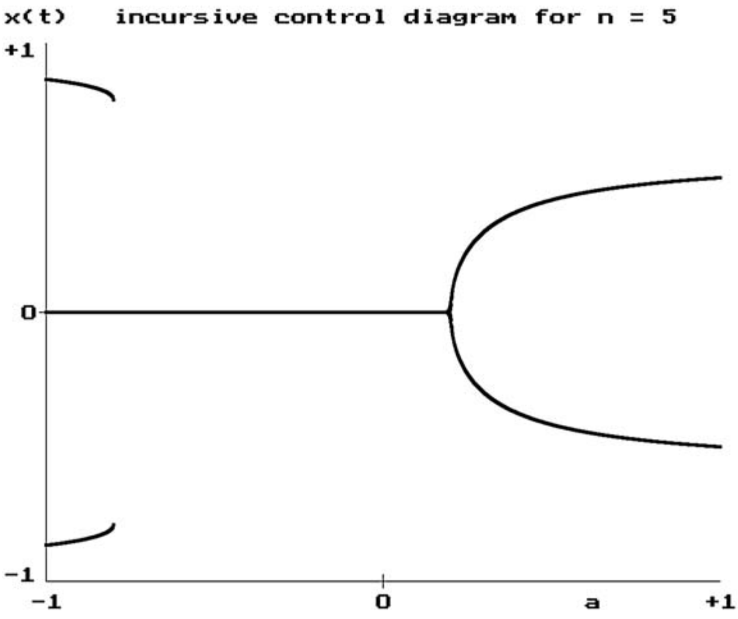

Fig. 11. Stabilization diagram for $n = 5$

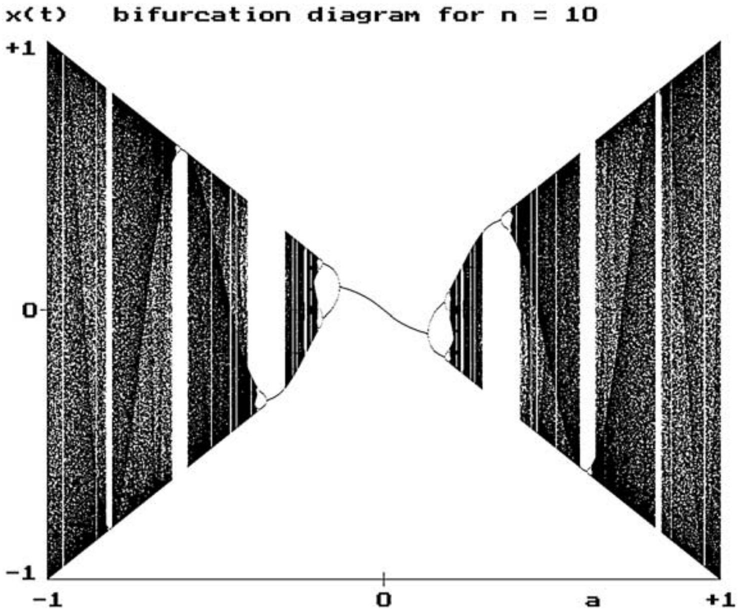

Fig. 12. Bifurcation diagram for $n = 10$

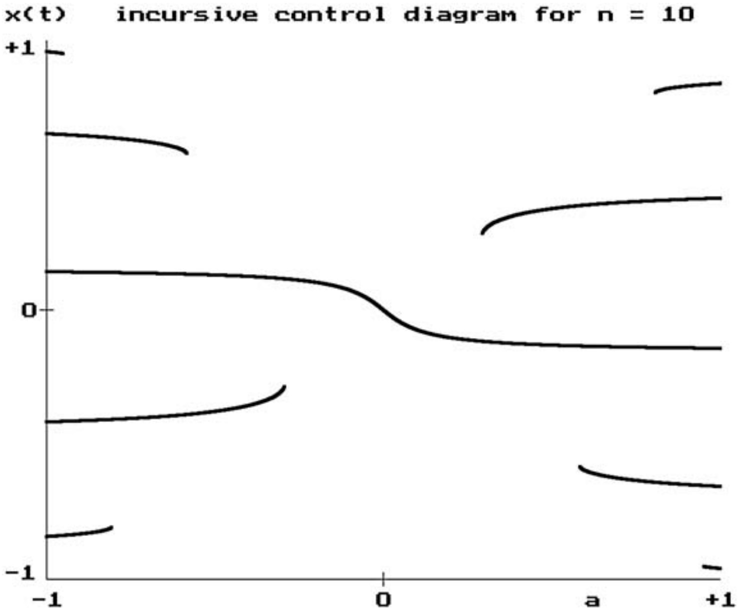

Fig. 13. Stabilization diagram for $n = 10$

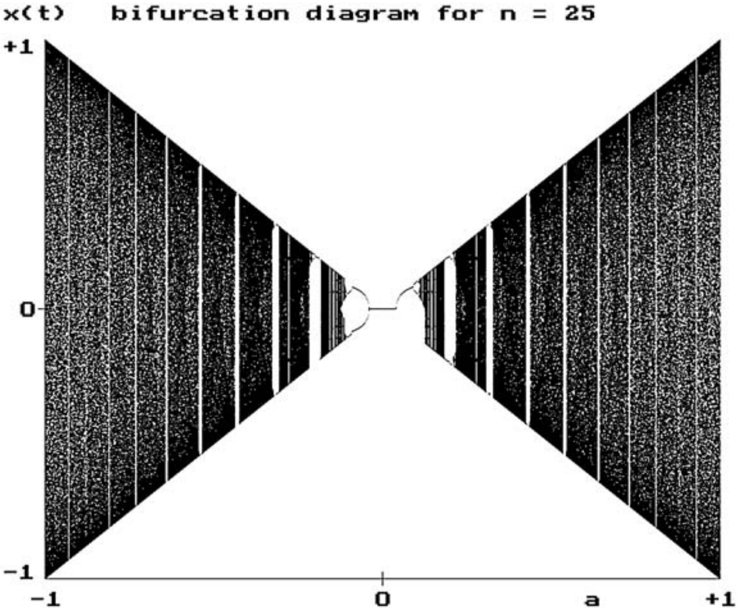

Fig. 14. Bifurcation diagram for $n = 25$

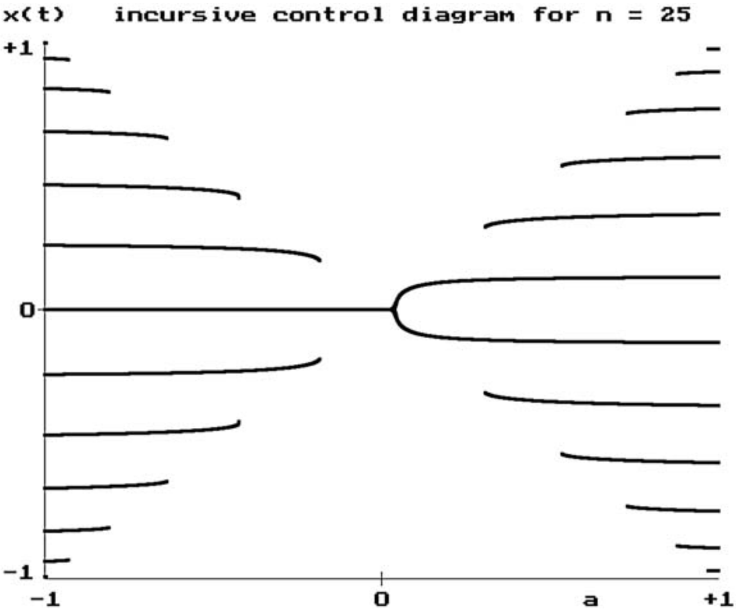

Fig. 15. Stabilization diagram for $n = 25$

6 Conclusions

This chapter firstly presented the Verhulst logistic differential equation and its analytical solution for various values of the growth parameter.

The classical discretization of this equation, with the Euler algorithm, gives the so-called Verhulst map, exhibiting fixed points, bifurcations and, then chaos for successive values of the growth parameter.

In the chaos zone, the Verhulst map has a closed form solution depending on the initial condition. It was recalled that the Verhulst map can be transformed to the Shift map for which the successive iterates are given by the fractional part of the shift, to the left, of the binary digits of the initial condition.

The first part of this chapter showed that the bifurcations and chaos in the Verhulst map are due to the Euler algorithm and not to the fundamental properties of the original Verhulst equation. It was demonstrated that the correct algorithm is the incursive algorithm (for inclusive or implicit algorithm) which gives fixed points similar to those of the original Verhulst equation, what we called the Verhulst incursive map.

Such a Verhulst incursive map can be obtained from the Verhulst chaos map with an incursive control, which is an anticipative controller.

The second part of this chapter demonstrated that the Verhulst chaos map belongs to a family of canonical chaos maps, given by the Tchebychev polynomials at any order (degree of the polynomials), for which closed form solutions exist.

These canonical chaos maps can be generated by a simple relation of recurrence.

With introduction of a growth parameter into these canonical chaos maps, a set of Verhulst chaos maps at any order is proposed.

For some of these Verhulst chaos maps, their first return diagrams and bifurcation diagrams are numerically computed.

Finally, the simulations of the anticipative control of these maps, with an incursive controller, give stable fixed points which are exhibited in stabilization diagrams.

In conclusion, the incursive tool is a powerful anticipative method for stabilizing chaos, and open new avenues of research and development in the framework of the theory of chaos.

Acknowledgements

I would like to thank Professor Mitchell J. Feigenbaum for very fascinating discussions and for having pointed out that the canonical chaos maps are related to the Tchebychev polynomials.

References

1. E.F. Camacho, C. Bordons: *Model Predictive Control* (Springer, Berlin 1999)
2. D.W. Clarke, C. Mohtadi: Automatica **23**, 137 (1987)
3. C.R. Cutler, B.L. Ramaker: Dynamic Matrix Control – A Computer Control Algorithm. In: *Proceedings of the 1980 Joint Automatic Control Conference* (San Francisco 1980) pp WP5-B,
4. D.M. Dubois: Review of Incursive, Hyperincursive and Anticipatory Systems – Foundation of Anticipation in Electromagnetism. In: *Computing Anticipatory Systems: CASYS'99 – Third International Conference*, ed by D.M. Dubois (The American Institute of Physics, AIP Conference Proceedings 517, 2000) pp 3–30
5. D.M. Dubois: Mathematical foundations of discrete and functional systems with strong and weak anticipations. In: *Anticipatory Behavior in Adaptive Learning Systems, State-of-the-Art Survey*, ed by M. Butz et al., Lecture Notes in Artificial Intelligence, Springer, LNAI 2684, pp 110–132, 2003
6. M.J. Feigenbaum: J. Stat. Phys **19**, 25 (1978)
7. M.J. Feigenbaum: Physica D **7**, 16 (1983)
8. H.-O. Peitgen , J. Hartmut, D. Saupe: *Chaos and Fractals, New Frontiers of Science* (Springer, Berlin 1992)
9. J. Richalet, A. Rault, J.L. Testud, J. Papon: Automatica, **14**, 413 (1978)
10. D.E. Rivera, M. Morari: Ind. Eng. Chem. Process Des. Dev. **25**, 252 (1986)
11. P.-F. Verhulst: Correspondance Mathématique et Physique **10**, 113 (1838)